Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Constructing tubular networks that occupy arbitrary regions in $\mathbb{R}^3$

To cite this article: F Ghasempour *et al* 2018 *J. Phys.: Conf. Ser.* **1047** 012003

IOP Conf. Series: Journal of Physics: Conf. Series **1047** (2018) 012003 doi:10.1088/1742-6596/1047/1/012003

Constructing tubular networks that occupy arbitrary regions in $\mathbb{R}^3$

F Ghasempour[1], W Jiang[1], F Mendivil[2], S D Peterson[3] and E R Vrscay[1]

[1] Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

[2] Department of Mathematics and Statistics, Acadia University, Wolfville, Nova Scotia, Canada B4P 2R6

[3] Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

E-mail: fghasemp@uwaterloo.ca, wenzhe.jiang0801@gmail.com, franklin.mendivil@acadiau.ca, peterson@uwaterloo.ca, ervrscay@uwaterloo.ca

Abstract. Advances in additive manufacturing have enabled industry to relax many design constraints imposed by traditional construction methods. As such, it is now possible to design and build objects or devices with complex internal structures that conform to irregular external envelopes. In applications where fluid distribution or storage is an integral part of a larger system, additive manufacturing technologies allow "left over space" in an overall device volume to now be effectively utilized. Herein we describe a general algorithmic framework for the construction of branched networks of tubes which occupy a specified, and possibly quite complicated, region $D \subset \mathbb{R}^3$ as fully as possible. Such networks can be important in a variety of industrial applications ranging from heat exchangers to storage vessels to fluid distribution networks. Depending on the specific application of interest, such a design problem can be extremely ill-posed: For a given region D, our algorithm can produce an enormous number of networks, which generally requires that a much smaller number of "best" networks be isolated by ranking the networks based upon some problem-dependent properties (*e.g.*, heat transfer rate, minimal friction losses, volume fraction, etc.). Some simple examples are presented to illustrate the algorithm and its output with comments added to illustrate its potential utility in industrial applications.

1. Introduction

Additive manufacturing, which generally builds a 3D object by adding layers of material [5], has revolutionized the design process by enabling the construction of objects with complex structures with relative ease [17]. This now allows designers to develop engineering structures that minimize material usage, have complex internal configurations and/or conform to complex external envelopes [7, 14]. In engineering systems consisting of fluid distribution or storage elements, additive manufacturing opens the possibility of exploiting left over or wasted space within a given geometric envelope to house these flow elements.

In this paper we describe a general algorithmic framework for the construction of branched tubular networks that occupy a specified, and possibly quite complicated, region $D \subset \mathbb{R}^3$ as fully as possible. Such "conformable tubular networks" (CTNs) can be important in a variety of industrial applications ranging from heat exchangers to fluid distribution or storage networks in complex geometries. Our algorithm is comprised of three fundamental modules: (i) the *analysis and decomposition* of the three-dimensional region D into subregions D_i, (ii) the *packing* of each subregion D_i with an assembly of tubes and finally (iii), the *connection* of the tubes, both within each subregion D_i as well as between neighbouring subregions.

1.1. A simple, motivating example: The "barbeque pool heater"

In [9] and [11], we considered some simple, yet nontrivial, "saddle-bag" regions D (*i.e.*, two larger rectangular regions – the "bags" – connected by a narrower rectangular channel) and produced sets of networks for these regions. One of the motivations for these constructions was the so-called "barbeque (BBQ) pool heater" problem [1] wherein water from a pool is pumped into a tubular network situated inside a barbeque via an inlet pipe. As the water travels through the network, it is heated by the barbeque flames. The heated water then exits the network through an outlet pipe and re-enters the pool. While this is possible to design and construct using conventional design and manufacturing techniques, it presents a simple framework through which to explore a more general design paradigm for encapsulating a (possibly branched) tubular network in a prescribed geometric envelope.

Figure 1 shows a particular example of a saddle-bag region representing, for example, the available space inside the barbeque (left), along with one of the many one inlet-one outlet networks produced by our algorithm (right). The network shown is a "28-8-28 saddle-bag network," where two "side bags," each of which contains 28 tubes, are connected to a middle region containing 8 tubes. In this simple example we utilize a single diameter for the tubes, which is prescribed ahead of time. For the barbque pool heater, which relies on natural convection and radiation to heat the tubes passing over the flames, the tube diameters shown in the Figure would not represent the actual vessels, but rather the desired spacing between the tubes for optimal heat transfer. The actual physical tubes would concentric with the tubes shown, but with smaller diameter to enable space for the heated air to pass around the fluid-transporting tubes.

Whether or not a given network is "good" is problem-specific. In this particular example, with the constraints that were imposed during the construction phase, hundreds of thousands of unique networks were generated. For this example, selecting the best network(s) would require an additional heat transfer analysis step. The focus of this paper is the methodology for constructing the tubular networks, not the problem-specific step of imposing additional physics and/or constraints for paring down the solution space.

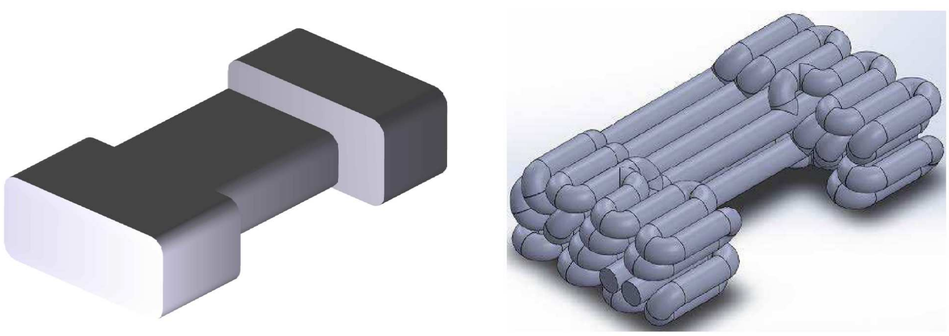


Figure 1. A "saddle-bag" region D (left) and a "28-8-28" tubular network produced by our algorithm which occupies the region.

2. The algorithm

This section describes the skeleton of the algorithm for constructing branched tubular networks in a given region of space. The algorithm is comprised of three phases, which will be detailed in the following subsections. It is presumed that the volume to be packed with tubes is prescribed.

Phase 1: Analysis/decomposition of region $D \in \mathbb{R}^3$ to be occupied by the network

Clearly, the simplest situation is when the 3D region D is a long block with constant cross section (*e.g.*, rectangular, triangular), in which case the tubes can be packed in a parallel manner and connected to each other at the ends. The next simplest case, a variation of the above, is when D exhibits some kind of longitudinal symmetry, *i.e.*, there exists a principal (but not necessarily straight) axis along which the cross section may vary. In this case, the construction problem is quasi-two-dimensional and the problem of packing tubes in three dimensions is reduced to circle-packing in two dimensions. A decomposition of D may then be performed by segmentation, *i.e.*, by approximating it as well as possible by a union of blocks B_i with constant cross sections. A sketch of such a generic decomposition is shown in Figure 2. Each block B_i is then packed with a parallel set of tubes as shown schematically in the figure. The "saddle-bag" region shown in Figure 1 possesses this longitudinal symmetry and is composed of three blocks.

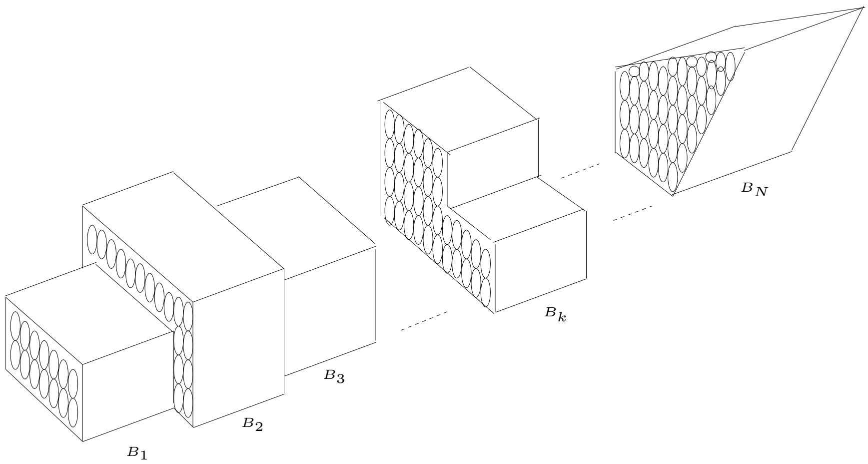


Figure 2. Generic structure of tubular network exhibiting variation along a principal direction.

In more complicated situations, it may be necessary to employ a method which can extract the fundamental morphological characteristics of region D, namely, its shape and structure. For example, if D possesses a dendritic structure in which several subprincipal axes emanate from a principal location or axis, it is desirable to determine these axes. Such information will be important for both the segmentation/discretization of the region as well as the determination of packing directions. Methods such as skeletonization [12, 13] can be employed to identify principal (of subprincipal) axes in a complex geometry, which can then segmented into a series of blocks for packing, as discussed above.

Phase 2: Packing the tubes

As mentioned in Phase 1, each block B_i comprising the envelope volume is assumed to contain a set of tubes arranged in a parallel fashion. The problem of packing each block B_i is therefore reduced to the two-dimensional problem of packing circles as efficiently as possible into its (constant) cross-sectional region. Circle packing is a well-studied problem with a long history because of its mathematical challenges as well as its important real-world applications (*e.g.*, cutting circular discs out of sheets of metal in a way that minimizes leftover material). There is a vast literature which is concerned with optimal and suboptimal methods for packing circles into regions. For a more recent review of the literature, see [8]. A significant portion of the literature has been devoted to the problem of packing as many identical circles of fixed radius into a given region, usually a regular region such as a circle, square or rectangle [2, 15, 16]. Another important set of problems is concerned with the efficient packing of square, rectangular and circular regions with circles of prescribed, but not necessarily identical, radii [4, 3].

In the the 28-8-28 "saddle-bag" network presented in Section 1, a very simple hexagonal packing of tubes of equal diameters was employed. In general, however, the cross-sectional regions are more complicated, *i.e.*, polygonal at the very best. The efficient packing of such regions will generally require tubes/circles of different sizes. Early investigations, however, showed that a more efficient packing of complicated regions could be achieved not with circles of prescribed sizes but with circles that were "tailor-made" for local regions, *i.e.*, circles designed to touch several already-packed circles as well as the boundary, if applicable. As such, the radii of these circles would have to be computed as the region was being packed. To the best of our knowledge, such algorithms did not exist in the literature, which motivated us to develop our own methods.

The starting point of our work on packing was the circle-packing algorithm by George, George and Lamar [4], hereby to be referred to as the GGL algorithm. One of the original motivations of the GGL paper was the efficient packing of pipes or bottles in a rectangular box for shipping (under the effect of gravity). Since the aims of our tubular network problem differ from those that led to the GGL algorithm, a number of differences between our algorithms and the GGL algorithm naturally arose. Here, we shall simply state that during the early stages of our research, a number of packing methods were developed and tested. A detailed discussion of these early methods is given in [11].

Our early modified-GGL methods were designed to pack regions with polygonal boundaries and worked very well for (convex and nonconvex) polygonal regions for which the lengths of the segments comprising the boundary are large compared to the circle sizes. If this condition is not satisfied – which includes the case of quite irregular boundary curves – the methods do not work very well. One could still consider employing one or more of these methods by using polygonal approximations of irregular boundary curves, but possibly at the expense of yet-unpacked regions. For this reason, we have recently developed another method of circle-packing which can be used in the case of irregular boundaries. This method, which will be described briefly below, works with a *discretization* of region D into rectangular regions of dimension Δx by Δy. The parameters Δx and Δy are chosen according to the refinement desired for the problem.

In this method, we begin once again with a Step 1 (hexagonal) packing of region D with circles of radius R. After Step 1 is completed, additional circles are packed in a stepwise fashion as follows:

(i) At the beginning of each step, let $D' \subset D$ denote the subregion of D which is not covered by packed circles and which has at least part of the boundary of D as its boundary. (This condition will exclude regions that lie between packed circles. It can be modified to accommodate such regions if they are suitably large.)

(ii) Let $\partial D'$ denote the boundary of region D'. Compute the distance of each mesh point in D' to $\partial D'$. The result is a distance function defined over D' which may be represented in terms of contours. Let $(x_{\max}, y_{\max})$ denote a point in D' at which the distance function achieves a local maximum value, to be denoted as $d_{\max}$. Pack, if possible, a circle of radius $d_{\max}$ centered at $(x_{\max}, y_{\max})$. Pack as many such circles in D' as possible. After all possible circles are packed, redefine $D' \subset D$ as the new subregion of D which remains unpacked and go to (i).

IOP Conf. Series: Journal of Physics: Conf. Series **1047** (2018) 012003 doi:10.1088/1742-6596/1047/1/012003

The above procedure may be stopped when circles of radii less than an acceptable radius $R_{\min}$ are packed (for example, if there is a minimum tube radius that can be manufactured, or there is a minimum radius imposed by some other exogenous constraint).

A few remarks on this method are in order.

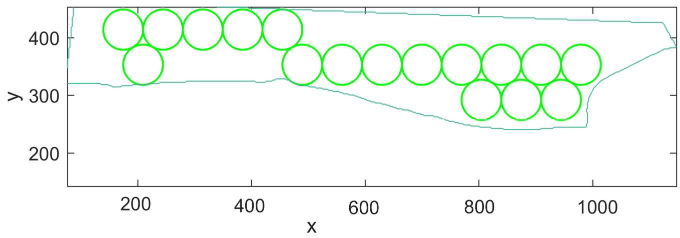
- From a practical perspective, the computation of distances between points of region D' and its boundary must be performed over a *mesh* of interior points. The MATLAB function `bwdist` can be used.
- It is not only quite possible, but quite probable, that points of maximum distance $d_{\max}$ are not isolated but lie on a continuous curve or region. At present, we simply use a center point of such a curve or region, acknowledging that some improvements could be made in future versions of the algorithm.

This method works very well for regions with polygonal boundaries. In Figure 3 are shown some results obtained in the case of a region D with a more complicated boundary. A mesh constructed from 1086 by 326 points was employed. A prescribed radius of $R = 35$ was used for the Step hexagonal packing (top left) with zero angle of orientation. The MATLAB distance function `bwdist` was used to compute the distances of all mesh points in the unpacked region to the boundary of the region. The resulting contours of the distance function are shown in the plot at the top right. Circles are then constructed at local maxima of the distance function as shown in the plot at the bottom left. After all possible circles are plotted in this way, the distances of all mesh points in the remaining unpacked region are computed and new circles are packed (bottom right). In this example, the prescribed minimum radius was $R_{\min} = 3$. The packing procedure was repeated until no circles of radius greater than or equal to $R_{\min}$ could be packed.

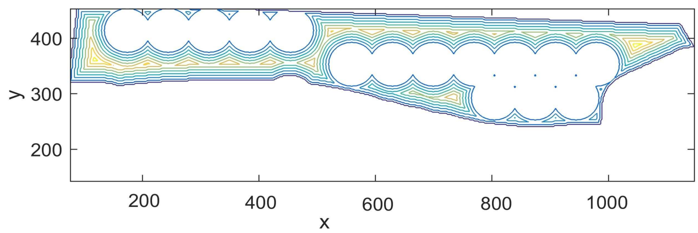
Packing common cross-sectional regions first In order to facilitate connections of tubes in adjacent cross-sections, it is advantageous to look for an interior (planar) region R that is common to several, if not all, adjacent block cross-sections. These regions are packed first. In this way, it is possible that some tubes will extend continuously from one block into another possibly through all blocks comprising the network. Such tubes are clearly visible in the 28-8-28 network shown in Figure 1 (there are 8 such tubes in total). Each non-common region is then packed. The packing of each such region begins with circles that touch already-packed circles from the common region. The packing then proceeds outward to the boundaries.

Phase 3: Connecting the tubes to produce a tubular network
With reference to the general network shown in Figure 2, after all of the blocks B_i have been packed with tubes (including tubes that extend continuously from one block to the next, possibly from one end of the network to the other), it remains to connect them to produce a feasible tubular network with a specified number of inlets and outlets. As will be shown below, there are, in general, a significant number of possible connections at each end and each block-to-neighbouring-block interface which can result in an enormous number of possible networks. Herein we will restrict ourselves to a small set of elements to enable connections, detailed in the following subsection. That will be followed by a example showing how these simple connecting elements can lead to a very large number of unique networks.

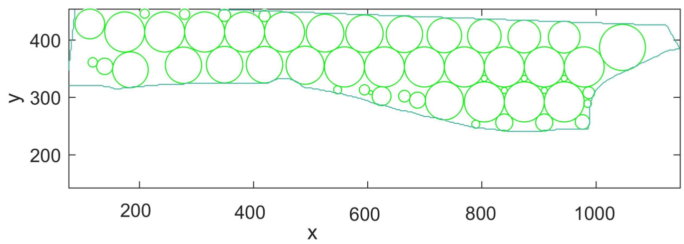
Basic connection elements Currently, we have employed the following three **fundamental elements** to construct our tubular networks: (i) straight tubes, (ii) 90-degree bends and (iii) T-joints (or “Tees”). These three elements are sketched in Figure 4.



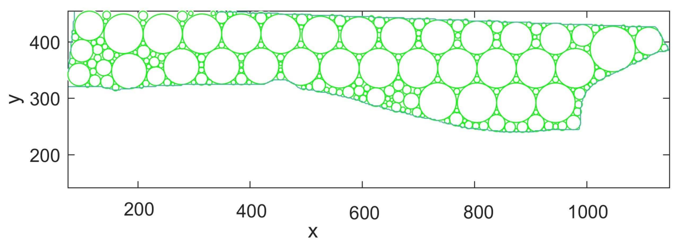
Step 1: 17 circles of radius $R = 35$.



Contour plot of distance function from points in the unpacked region to its boundary.



Packing the unpacked region of previous figure using distance function contours. New circles are centered at local maxima of the distance function.



Additional packing after distance function contours are computed for unpacked region in upper plot.

Figure 3. Packing method applied to a region D with a more complicated boundary.

Note that the sketches in Figure 4 suggest that the diameters of all tubes are the same. This is not a fundamental restriction of the algorithm, although accommodating multiple tube sizes will require an augmentation to these elements. Using these elements, we currently employ four basic operations to construct our our tubular networks, shown schematically in Figure 5.

- **Endcap connection:** A 180-degree bend, formed with two 90-degree bends, which connects two neighbouring tubes at their ends. It is used at the extreme ends of the

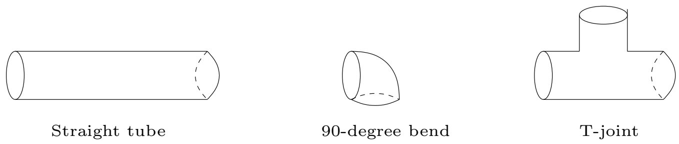


Figure 4. The three fundamental elements employed in the construction of tubular networks in this paper.

network as well as at the ends of each block B_i.

- **Simple merge:** Tube a flows into a neighbouring tube b. Formed with a 90-degree bend and a T-joint. This operation introduces branching in the tubular network. Used only at an at interface of two neighbouring blocks B_i, producing a 2-to-1 reduction in the number of tubes from one block to the next.
- **Consecutive merge:** Tube a flows into neighbouring tube b which then flows into neighbouring tube c. Used only at an interface of two neighbouring blocks B_i, producing a 3-to-1 reduction in the number of tubes.
- **Merge/shift/merge operation:** As shown below, tube a flows into neighbouring tube b. As well, tube c flows into tube d. Tube b is then shifted to the former position of c. The shift is formed with two 90-degree bends. Used only at at interface of two neighbouring blocks B_i, producing a 4-to-2 reduction in the number of tubes.

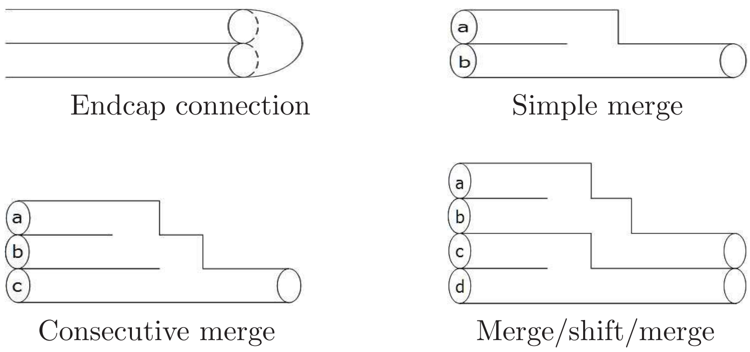


Figure 5. The four basic connection schemes used to construct the tubular networks.

The above list of connection schemes is by no means complete. Consecutive merges with higher than a 3-to-1 reduction, as well as merge/shift/merge schemes with higher reduction ratios can be considered, if necessary. Additional elements and combinations can also be introduced to accommodate specific geometries, *e.g.*, non-90-degree bends, contractions/expansions, curved tubes, manifolds connecting many tubes, etc.

The connection algorithm In general, two neighbouring blocks B_i and B_{i+1} will have a common area, to be denoted as region $\mathcal{R}_i$, as sketched in Figure 6. Circles in this region represent tubes that can run continuously from B_i to B_{i+1}. Tubes that are in block B_i but not completely in region $\mathcal{R}_i$ clearly cannot be continued into block B_{i+1}. As such, they must be either (i) turned back into block B_i via an endcap connection or (ii) merged into a neighbouring tube which lies in $\mathcal{R}_i$. There may also be the possibility of shifting a tube into a neighbouring position which extends into region $\mathcal{R}_i$ by means of a merge/shift/merge operation. The same comments apply to tubes in block B_{i+1} which cannot run into block B_i.

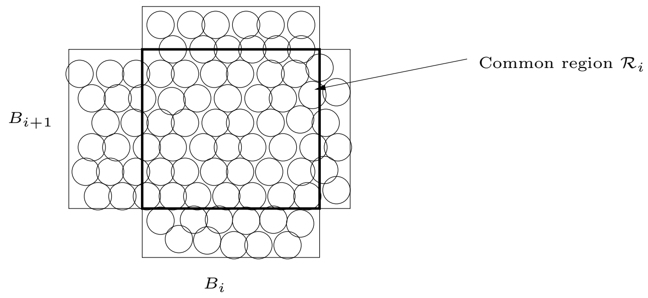


Figure 6. Two packed neighbouring blocks B_i and B_{i+1} with common region $\mathcal{R}_i$.

The 28-8-28 saddle-bag network shown in Figure 1 illustrates these connection schemes. The bags of the saddle-bag network, B_1 and B_3, contain 28 tubes whereas the interior connector region B_2 contains only 8 tubes. As such, 8 tubes run from one end, *i.e.*, B_1, through the connector region B_2, to the other end B_3. The other tubes 20 tubes in B_1 and B_3, however, must remain in their respective bags.

Before discussing the more general case, we illustrate the basic idea of our connection algorithm by means of a particular solution to a network which is composed of a much smaller number of tubes, namely, a 9-4-9 saddle-bag network. One such network is shown in Figure 7.

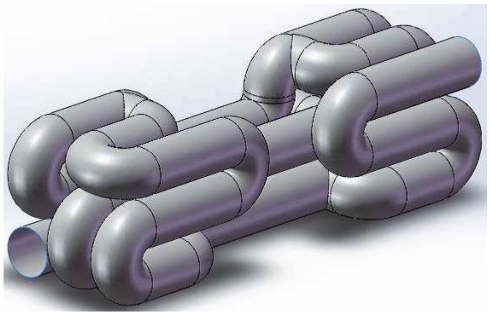


Figure 7. A 9-4-9 network produced by our algorithm.

The cross-sectional regions of the two bags as well as the connector region of a generic 9-4-9 network are shown in Figure 8.

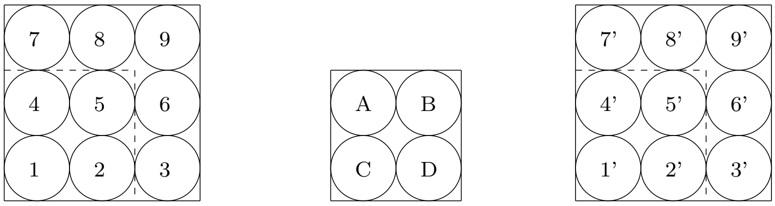


Figure 8. Cross sections of 9-4-9 saddle-bag network. **Left:** Saddle bag “L” with 9 tubes. **Middle:** Connector region with 4 tubes. **Right:** Saddle bag “R” with 9 tubes.

Using the ideas described at the beginning of this section, we present one solution for the connections at the B_1-B_2 and B_2-B_3 interfaces. We start at side L and determine how the nine tubes in the saddle bag will become Tubes A,B,C and D in the connector region.

IOP Conf. Series: Journal of Physics: Conf. Series **1047** (2018) 012003 doi:10.1088/1742-6596/1047/1/012003

- Tube A: Tube 7 merges into Tube 4, which then becomes Tube A. (This implies that Tube 4 is not available for any other operation.)
- Tube B: Tube 9 merges into Tube 8. Tube 8 then merges into Tube 5, which then becomes Tube B. (This implies that Tube 5 is not available for any other operation.)
  Tubes 6 and 3 must still be "removed." This means that Tube 2 will have to be moved:
- Tube C: Tube 2 merges into Tube 1, which then becomes Tube C.
- Tube D: Tube 6 merges into Tube 3, and then Tube 3 shifts to Tube 2 (after Tube 2 has merged to Tube 1).

The result of the above series of steps is a particular **merge-shift configuration** for the 9-4-9 saddle-bag network which is conveniently represented by the diagram shown in Figure 9.

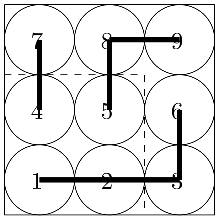


Figure 9. Diagrammatic representation of the resulting set of connections produced by the procedure outlined above.

Of course, similar shifting and merging operations must be performed on Tubes 3', 6', 7', 8' and 9' in order to move from saddle bag R into the connection region. For simplicity, we consider the same merging and shifting operations on saddle bag R as was done for saddle bag L in the steps above. Notationally, we simply replace all tube numbers with primes, *i.e.*, 3 with 3', etc. The resulting, almost-finished network may be represented by the diagram in Figure 10.

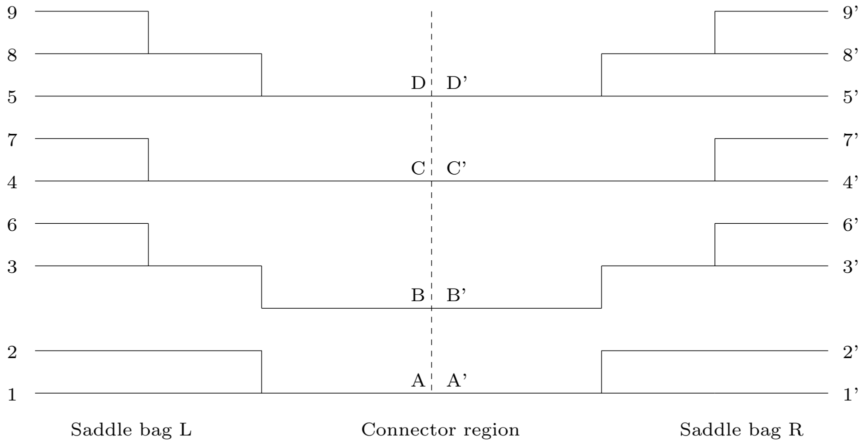


Figure 10. Graphical represntation of shift/merge connections between saddle bags and connector region.

The remaining problem is to make appropriate endcap connections between neighbouring tubes at both ends. One solution to the endcap problem for the above connection scheme is shown in Figure 11. Note that we have once again used symmetry, *i.e.*, the same set of connections on each side of the network. The net result is that Tubes 3 and 3' are, respectively input and output (or vice versa).

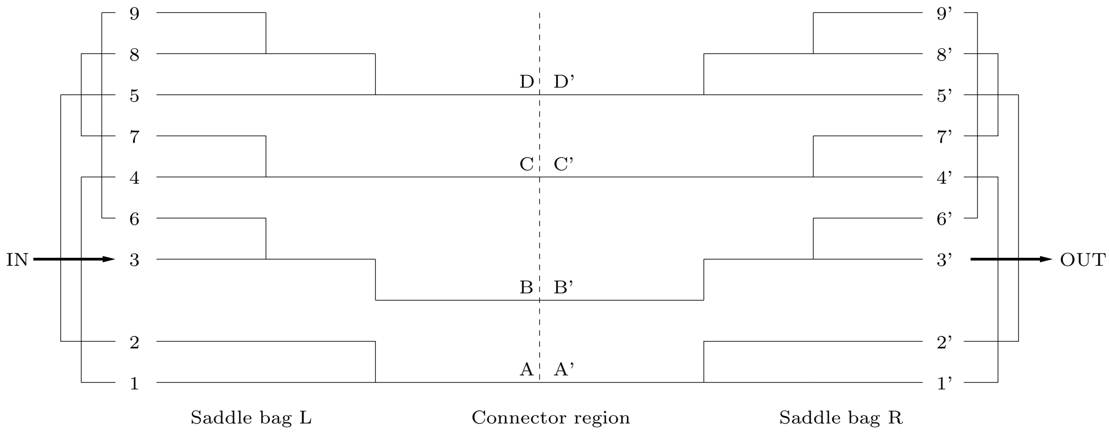


Figure 11. Graph of Figure 10 with a feasible set of endcap connections.

Here we once again emphasize that the connection scheme in Figure 11 is only one of several possible endcap schemes associated with this merge-shift configuration.

A brief description of the general connection algorithm The problem of connecting tubes at block interfaces is more complicated than that of connecting them at the extreme ends of the network. As such, we first consider the interface problem, with reference to Figure 6. For simplicity, our discussion below is restricted to the circles (which represent tubes) that are in Block B_i of the figure. Firstly, circles of B_i that lie entirely in the common region $\mathcal{R}_i$ are referred to as **common circles**. The remaining circles of B_i are referred to as **boundary circles** (BCs). It is the boundary circles which must be "processed," *i.e.*, endcapped, merged, or shifted so that they disappear when B_i is exited. The farther a circle lies from the common region, the higher the priority that it be considered for connection/shifting. This is accomplished by assigning a "layer-rank" index L_r to each circle, as follows:

- First-layer BCs, also called 1B circles, touch common circles. $L_r = 1$.
- All other BCs are called 2B circles, $L_r = 0$.
- First-layer common circles, $L_r = 2$, or 1C circles, touch at least one BC.
- Second-layer common circles, $L_r = 3$, or 2C circles, touch at least one 1C circle.
- All other common circles are not involved in this algorithm since it is not possible for them to be connected at an interface using any of the merge operations shown in Figure 5. $L_r = \infty$.

Circles are then processed in increasing order of layer rank index L_r.

There is another important factor. Since it is necessary that all BCs be "processed" and not ignored in any step, it is convenient to consider those BCs with a lower number of available connections first. As such, we define the **connectivity degree**, CD, of a BC as the number of (neighbouring) circles to which it may be connected.

Finally, it is convenient to define the **connectivity value** CV of a circle/tube, the maximum number of tubes to which it can be connected in one of the four operations shown in Figure 5:

- A 1B tube can be the middle of a consecutive merge (tube b, connected to both tubes a and c). A 1C tube can be at the end of a consecutive merge or a merge/shift/merge (tube c, connected to tubes a and b in both cases). $CV = 2$.
- 2B or 2C tubes can be connected to only one tube. $CV = 1$.

IOP Conf. Series: Journal of Physics: Conf. Series **1047** (2018) 012003 doi:10.1088/1742-6596/1047/1/012003

Tube	L_r	CV	Available tubes	CD
1	3 (2C)	1	2,4	2
2	2 (1C)	2	1,3,5	3
3	1 (1B)	2	2,6	2
4	2 (1C)	2	1,5,7	3
5	2 (1C)	2	2,4,6,7	4
6	1 (1B)	2	3,5,9	3
7	1 (1B)	2	4,8	2
8	1 (1B)	2	5,7,9	3
9	0 (2B)	1	6,8	2

Table 1. Availability list for the 9-4-9 network Saddle bag L/connector interface shown in Figure 8.

The indices associated with each tube, along with the set of tubes to which each tube can be connected, comprise an "availability list". For example, the availability list for interface between Saddle Bag L and the connector region of the 9-4-9 network shown in Figure 8 is presented in Table 1.

Starting with such an availability list, our algorithm examines all possible connections at an interface in a systematic manner. Given the different natures of the circles/tubes involved (*i.e.*, BCs vs. common circles) as well as the connection operations in Figure 5, there is a rather complicated set of rules for making connections. Furthermore, after a connection is made, the availability list is updated according to another set of rules. (For example, the CV values of the tubes involved in the connection are decreased and if the CV of any tube becomes zero, it must be removed from the availability list.) Because of space limitations, a discussion of these rules is not possible here. We therefore refer the interested reader to [11] (Section 3.4.2, p. 62) for details. Here we simply state that at each step of the algorithm, after one or more connections have been made and the availability list updated accordingly, all tubes available for connection according to the list are examined, starting with those of lowest layer-rank index L_r and proceeding with increasing L_r value. For a given tube, all tubes available for connection are examined. As such, the algorithm is naturally recursive in nature, examining all possible connections in a tree-like manner. If, at any step, all boundary circles of the interface are connected, the resulting set of connections are feasible, then the connection scheme is stored.

After all interface connection schemes have been computed, they must then be combined with each other, along with endcap connections at both ends of the network, to produce possible final tubular networks. Each of these networks must be tested for "feasibility" according to some requirements, *e.g.*, a prescribed number of inputs/outputs, no disconnected components, etc.

We now state some results for the simple 9-4-9 network problem – one inlet and one outlet – in order to give an idea of the complexity of this problem.

- With reference to the leftmost diagram in Figure 8, there are 22 feasible sets of connections at the interface of a bag and the connector region, one of which is shown in Figure 9.
- There are 5 possible sets of endcap connections at each end – four endcap connections between neighbouring tubes and one unconnected inlet/outlet.

The 9-4-9 networks are comprised of two ends and two interfaces. As such, there are $5 \times 22 \times 22 \times 5 = 133,100$ connections to examine. If we require that there must be at least one connected path between the inlet on one end and the outlet on the other end, *i.e.*, that the graph representing the tubular network be connected, then the number of such feasible networks is reduced significantly to 13,100. If, in addition, we demand that there are no closed loops or dead-ends in a network, *i.e.*, that each point in the network lies on a simple (*i.e.*, nonintersecting) path from an inlet at one end to an outlet at the other end, then the number of feasible networks is reduced to 10,066.

Note: In order to determine whether or not a network is connected/disconnected and whether or not there exist closed loops/dead ends, we must examine the incidence matrix $A(G)$ associated with the graph G that represents the network [6]. For details, see [11], Section 3.5, p. 66.

Comments on practical applications
In practice, the presented algorithm may need modification to suit problem-specific needs, particularly when selecting the "best" network from the family of generated solutions. For example, it may be desirable to minimize pressure drop across the tubular bed, or maximize volumetric capacity, or perhaps both. These outcome-based metrics can be used to develop a scheme for ranking the potential solutions. In simple cases, like the 9-4-9 example, there may not be large differences in performance between networks, but the variability in solutions is expected to increase substantially in more complex problems, as the number of solutions increases.

Concluding remarks

In this paper, we have outlined an algorithm for the construction of conformable tubular networks which occupy, as fully as possible, a given three-dimensional region D. The algorithm decomposes the region (with assumed longitudinal axis) into a series of blocks with fixed cross-sections. Circles are subsequently packed into each cross-section and connected via a set of rules to produce a tubular network.

Currently, we are focused on an extension of the work in Phase 1 to accommodate more complicated regions which may not possess a single longitudinal axis. This necessitates the use of morphological methods such as skeletonization to extract principal and subprincipal axes of more complicated regions and to then use this information to devise efficient packing strategies.

Our framework is flexible and can accommodate the specific requirements that may arise in industrial applications (*e.g.*, flow through the tubes, heat transfer, etc.) through the imposition of additional constraints either within the algorithm itself, or during the selection of viable tubular networks.

Acknowledgments

This research has been supported primarily by a Collaborative Research and Development Grant from the Natural Sciences and Engineering Research Council (Grant No. CRDPJ 453649-13). Additional support, in the form of teaching assistantships (JW), was provided by the Department of Applied Mathematics and the Faculty of Mathematics, University of Waterloo.

References

[1] http://www.redneckpoolheater.com
[2] Birgin EG and Gentil JM 2010 New and improved results for packing identical unitary radius circles within triangles, circles, rectangles and strips *Comp. Oper. Res.* **37** 1318-1327
[3] Castillo I, Kampas FJ and Pinter JD, Solving circle packing problems by global optimization: Numerical results and industrial applications 2008 *Eur. J. Oper. Res.* **191** 786-802
[4] George JA, George JM and Lamar BW 1995 Packing different-sized circles into a rectangular container *Eur. J. Oper. Res.* **84** (3) 693-712
[5] Gibson I, Rosen DW, Stuker B 2015 *Additive Manufacturing Technologies* Second Edition (NY: Springer)
[6] Gross JL and Yellen J 2005 *Graph Theory and its Applications* Second Edition (Boca Raton, FL: CRC Press)
[7] Gu DD, Meiners W, Wissenbach and Poprawe R 2012 Laser additive manufacturing of metallic components: Materials, processes and mechanisms *Int. Mat. Rev.* **57** 133-164
[8] Hifi M and M'Hallah R 2009 A literature review on circle and sphere packing problems: models and methodologies *Adv. Oper. Res.* Article ID 150624 http://dx.doi/org/10.1155/2009/150624

[9] Jiang W, Kettlewell B, Qiao T, Mendivil F, Peterson SD and Vrscay ER 2015 The barbeque pool heater: An algorithm to construct tubular networks the occupy arbitrary regions in $\mathbb{R}^3$ *Proc. AMMCS-CAIMS 2015 Congress* (Waterloo, ON, Canada)

[10] Jiang W, Qiao T, Mendivil F, Peterson SD and Vrscay ER 2015 Some novel circle-packing algorithms devised for the construction of tubular networks in $\mathbb{R}^3$ *Proc. AMMCS-CAIMS 2015 Congress* (Waterloo, ON, Canada)

[11] Jiang W 2015 *Construction of optimal tubular networks in arbitrary regions in* $\mathbb{R}^3$ M.Math. thesis, Dept. of Applied Mathematics, University of Waterloo

[12] Passalis G, Theoharis T, Kakadiaris IA 2007 A novel depth buffer-based shape descriptor for three-dimensional object retrieval, *The Visual Computer* **23** 5-14

[13] Saha PK, Borgefors G and Sanniti di Baja G 2016 A survey on skeletonization algorithms and their applications, *Patt. Rec. Lett.*, **76** 3-12

[14] Sames WJ, List FA, Pannala S, Dehoff RR and Babu SS 2016 The metallurgy and processing science of metal additive manufacturing, *Int. Mat. Rev.* **61** 1-46

[15] Specht E 2013 High density packings of equal circles in rectangles with variable aspect ratio *Comp. Oper. Res* **40** 58-69

[16] Szabo PG, Markot MCs, Csendes T, Specht E, Casado LG and Garcia I 2007 *New Approaches to Circle Packing in a Square with Program Codes* (NY: Springer)

[17] Vaezi M, Chianrabutra, Mellor B and Yang S 2013 Multiple material additive manufacturing Part 1: A review *Virtual and Physical Prototyping* **8** 19-50