Science maps for information retrieval

Bascur Cifuentes, J.P.

Chapter 2

An interactive visual tool for scientific literature search: Proposal and algorithmic specification

Abstract[1]

Literature search is a critical step in scientific research. Most of the current literature search tools present the search results as a list of documents. These tools fail to show the structure of the search results. To address this issue, we propose an interactive visual tool for searching scientific literature. This tool creates, labels and visualizes clusters of documents that may be of relevance to the user. In this way, it provides the user with an overview of the structure of the search results. This overview is intended to be understandable even to a user who has only a limited familiarity with the scientific domain of interest. We present the concept of our tool, show a case study of its use and describe the technical specifications of the tool. In particular, we provide a detailed specification of the algorithm that we use to visualize clusters of documents.

2.1 Introduction

Literature search is an essential part of any research project. Many of the current literature search tools (e.g. Google Scholar [66], Web of Science [41], Scopus [56] and Dimensions [51]) present the search results as a list of documents, without showing the structure of the results. Getting an understanding of the structure of the results, for instance by providing a breakdown of the search results into different research topics, can be useful for exploring the literature [1], especially for making serendipitous discoveries or for users that are new to a field of research.

There is some literature studying the idea of showing the structure of search results. An example is the recent work on a tool called PaperPoles [71], which uses citation links to create clusters of related papers. Various tools have also been made publicly available, some of them with a clear focus on literature search and others with a primary focus on bibliometric analysis. For instance, CiteSpace [39], CitNetExplorer [157] and Citation Gecko [163] can be used to visualize networks of

[1]This chapter is based on: Juan Pablo Bascur, Nees Jan van Eck and Ludo Waltman. 2019. An interactive visual tool for scientific literature search: Proposal and algorithmic specification. Proceedings of the 8th International Workshop on Bibliometric-Enhanced Information Retrieval (BIR) Co-Located with the 41st European Conference on Information Retrieval (ECIR 2019), 76–87. https://ceur-ws.org/Vol-2345/paper7.pdf [16]

citations between documents. Open Knowledge Maps [122] shows clusters of semantically-related papers. VOSviewer [156] presents visualizations of co-occurrence networks derived from papers (e.g. co-authorship links between authors, citation links between documents, or co-occurrence links between terms).

While these tools are helpful, some of them (e.g. CiteSpace, VOSviewer) were developed primarily for bibliometric analysis, not for literature search. Others (e.g. CitNetExplorer, Citation Gecko) have the limitation of showing search results only at the level of individual papers, not at aggregate levels. To overcome the limitations of currently available tools, we propose a new tool for literature search. This tool uses an interactive visual interface to show the structure of the search results. We make use of ideas and techniques that we also used in the development of other tools (i.e., VOSviewer and CitNetExplorer), but we now focus specifically on literature search rather than on bibliometric analysis. To some degree, the proposed tool resembles Open Knowledge Maps. However, by relying on the Scatter/Gather approach [48], the tool offers a higher level of interactivity, which facilitates the exploration of large document spaces.

This paper is divided into three parts: We first provide a description of the proposed tool (Section 2.2), we then present a case study demonstrating the use of the tool (Section 2.3) and finally we give a technical specification of the algorithms included in the tool (Section 2.4).

2.2 Description of the tool

Our proposed tool is based on the Scatter/Gather approach [48]. This approach consists of exploring a set of documents through multiple iterations of scattering and gathering. To scatter means creating clusters of documents and labeling them to understand their contents. To gather means selecting the clusters of interest, resulting in a new set of documents (Figure 2.1). The documents in our tool are scientific papers.

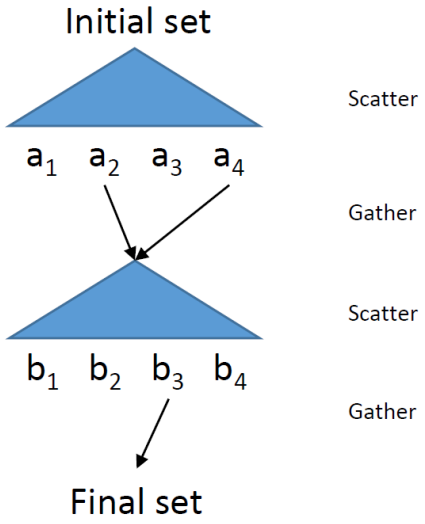

Figure 2.1: The Scatter/Gather approach. Figure inspired by Figure 1 of Cutting et al. [48]. The user scatters the initial set of documents into labeled clusters of documents (a1, a2, a3, and a4). Then she gathers the clusters she is interested in and creates a new set of documents. Then she scatters the new set into new clusters (b1, b2, b3, and b4). This process can continue a number of times.

Our tool scatters a set of papers into clusters. The clustering uses the citation links between papers. Each cluster is given a label. The label of a cluster consists of the ten noun phrases with the

highest weighted frequency in the titles and abstracts of the papers in the cluster. The weighting considers the frequency of occurrence of the noun phrases in the focal cluster relative to other clusters. This clustering and labeling method is based on Waltman and Van Eck [164].

Our tool also visualizes the clusters to complement the labels. It visualizes the clusters as bubbles in a packed bubble chart. The size of the bubbles reflects the number of papers in the clusters and the distance between the bubbles approximately reflects the number of citation links between the clusters.

Our tool supports multiple iterations of scattering and gathering. The user can load the initial set of papers, choose the clusters to gather, choose the number of clusters to scatter, retrieve the papers in the clusters, and so on.

2.3 Case study of the tool

2.3.1 Set up

First, let us consider a user working with a traditional literature search engine for scientific literature, like Google Scholar. She has to come up with several search queries. She does not have a background in the academic field that she is looking into, so probably she will not come up with good queries. Also, she has no way to know if she is missing important papers or even entire subfields!

Second, let us assume instead that she uses a literature search engine that offers some very basic features for exploring the structure of the search results, like Web of Science. She can now see to which academic fields her search results belong. Despite of this, she still has basically the same problems as with Google Scholar.

Third, now let us assume that she uses our proposed tool for her literature search. For this example, we will follow her through all the steps of the search process. We will assume that she is interested in getting to know the scientific literature about the review process of grant proposals. For the initial set of papers, we will use the set of the cluster of scientometrics papers obtained using the algorithmic methodology employed at CWTS [164]. We believe that she would have used the same set because it covers her topic.

2.3.2 Example of the search process

The researcher retrieves the set of papers and chooses a value of 10 for the number of clusters in the first scattering. Then she sees the visualization (Figure 2.2A) and the labels (Table 2.1) of the clusters. From the labels, she sees that her topic of interest is in cluster 6. She also checks the labels of the clusters close to cluster 6 (clusters 0, 3, 5, 8 and 9). Their labels indicate that they do not relate to her topic of interest, so she only gathers cluster 6.

She chooses to have 5 clusters for the second scattering and sees the visualization (Figure 2.2B) and the labels (Table 2.2) of the clusters. Now the labels are more ambiguous, so she will have to also read the titles of the papers inside clusters to understand what the clusters are about. She suspects that her topic of interest is in clusters 1 and 2. From the visualization and the labels, she also sees that her topic could be in cluster 4. She reads the titles of the top 5 most cited papers in these three clusters (Tables 2.3, 2.4 and 2.5). She finally decides that she should start reading paper 3 from cluster 1 and papers 2 and 4 from cluster 2.

In this example, we have illustrated how our tool could improve scientific literature search. The key advantage of the proposed tool is that the user is informed about the way in which the scientific literature is organized. For instance, the user is able to see how a field is divided into subfields or topics. As a result of this, the user is able to discard papers unrelated to the topic of interest without the need to skim the titles of large numbers of individual papers. Instead, the user examines the labels of clusters and then decides to discard entire clusters that appear to be of no relevance. Also, the user does not need to try to come up with a detailed keyword query that identifies exactly the right papers. It is sufficient to be able to identify a broad set of papers that could potentially be

of relevance. Within this broad set of papers, the papers of interest can then be found by drilling down into the right clusters.

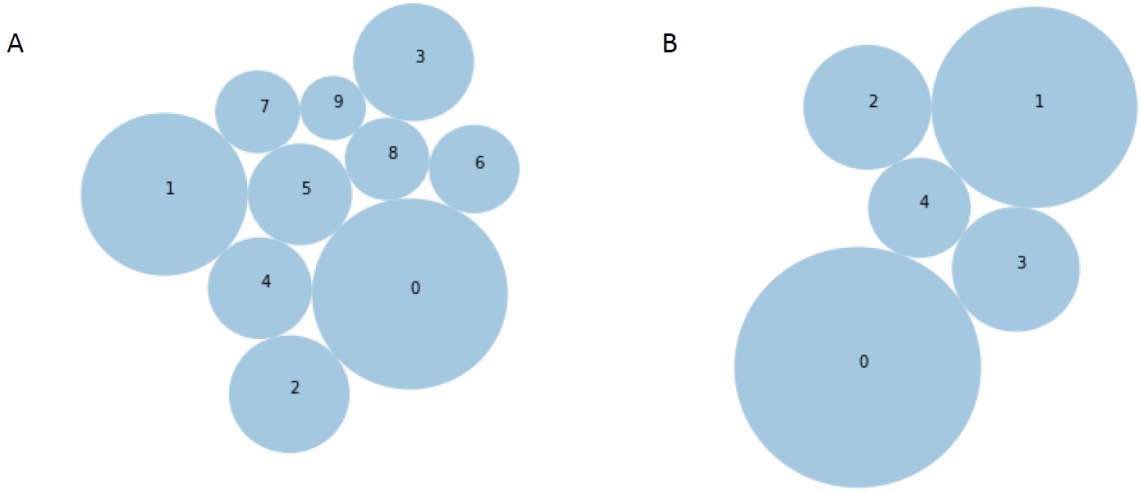

Figure 2.2: Visualization of clusters. The size of a cluster reflects the number of documents belonging to the cluster. Clusters that are strongly related (based on citation links) tend to be located close to each other. The numbers are the identifiers of the clusters. A: First scattering. B: Second scattering.

2.4 Technical specification

2.4.1 Clustering the documents

We cluster the papers by applying the Leiden algorithm to their citations links [153, 164]. The Leiden algorithm identifies clusters (or communities) of nodes within a network. We apply the Leiden algorithm to a directed network where the papers are the nodes and the edges are the citations between citing and cited papers. The Leiden algorithm has a resolution parameter that determines the number and size of clusters. To avoid requiring the user to set the resolution parameter manually, we developed a rule of the thumb that enables the user to specify the number of clusters C that she wishes. According to this rule, the resolution parameter is chosen in such a way that the largest cluster includes between $N/(C-2)$ and $N/(C)$ papers, where N is the total number of papers in the collection. To obtain the desired number of clusters after the clustering algorithm has been run, we keep the top C largest clusters and merge them with the other smaller clusters. We merge the pairs of clusters that have the highest relatedness, which we define as $e(c_1, c_2)/(n(c_1) * n(c_2))$, where c_1 and c_2 are the clusters, $e(c_1, c_2)$ is the number of edges between two clusters and $n(c)$ is the number of papers in a cluster.

2.4.2 Labeling the clusters

We label clusters using the approach developed by Waltman and Van Eck [164]. This approach extracts cluster labels from noun phrases in the titles and abstracts of the papers belonging to a cluster. It labels a cluster using noun phrases that are common in the cluster and relatively uncommon in other clusters. The only modification that we make to the approach introduced in [164] is that we report 10 noun phrases instead of 5.

Table 2.1: Labels of the first scattering. Scattered from the cluster of scientometrics papers [164].

ID	Top 10 noun phrases	Papers
0	hirsch | h index | g index | citation distribution | hirsch index | index | percentile | variant | google scholar | calculation	4344
1	man | gender difference | scientific collaboration | research collaboration | woman | co authorship network | international committee | gender | medical journal editors | icmje	3154
2	citation classic | article type | randomized controlled trial | year survey | gross domestic product | study design | pubmed database | subspecialty | population size | medline database	1652
3	open access | institutional repository | open access publishing | altmetric | oa journal | self archiving | open access journal | mendeley | repository | twitter	1651
4	author keyword | nanotechnology | patent citation | patent | chinese academy | nanotechnology research | nanoscience | keywords plus | productive journal | uspto	1231
5	interdisciplinarity | bibliographic coupling | co word analysis | research front | aca | map | intellectual structure | visualization | co citation | cluster	1230
6	peer review process | rejection | reviewer | peer reviewer | peer review | review quality | review process | manuscript | manuscript review | peer review system	932
7	link analysis | hyperlink | web page | inlink | web link | web site | yahoo | search engine | web impact factor | link count	816
8	marketing | operations management | management journal | citation error | finance journal | rpys | business school | quotation error | management discipline | reference accuracy	810
9	economics department | economist | economics journal | academic economist | economic research | economic | jel | american economic review | economics profession | top economics journal	492

Table 2.2: Labels of the second scattering. Scattered from cluster 6 of the first scattering.

ID	Top 10 noun phrases	Papers
0	conclusion | method | purpose | journal | author | manuscript | article | quality | background | editor	387
1	proposal | paper | referee | reliability | example | order | peer review | evaluation | science | application	270
2	nih | health | funding | grant application | national institute | grant | application | medical research council | cost | grant proposal	104
3	ecology | peer review system | concern | ecologist | model | simulation | publication process | researcher | system | evolution	104
4	scientific article | megajournal | traditional peer review | transparency | plos | oamj | oamjs | scientific soundness | scientific community | open access	67

Table 2.3: Top 5 papers for cluster 1 in the second scattering. The papers are ranked by number of citations. The citation counts were obtained from the citation network of the initial set of papers.

Rank	Title	Cit.	Year	Source
1	Scientific Peer Review	108	2011	ANNUAL REVIEW OF INFORMATION SCIENCE AND TECHNOLOGY
2	Bias in peer review	79	2013	JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY
3	Improving the peer-review process for grant applications – Reliability, validity, bias, and generalizability	72	2008	AMERICAN PSYCHOLOGIST
4	Selection of research fellowship recipients by committee peer review. Reliability, fairness and predictive validity of Board of Trustees' decisions	58	2005	SCIENTOMETRICS
5	Selecting manuscripts for a high-impact journal through peer review: A citation analysis of communications that were accepted by Angewandte Chemie International Edition, or rejected but published elsewhere	48	2008	JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY

Table 2.4: Top 5 papers for cluster 2 in the second scattering. The papers are ranked by number of citations. The citation counts were obtained from the citation network of the initial set of papers.

Rank	Title	Cit.	Year	Source
1	Big Science vs. Little Science: How Scientific Impact Scales with Funding	31	2013	PLOS ONE
2	Peer review for improving the quality of grant applications	23	2007	COCHRANE DATABASE OF SYSTEMATIC REVIEWS
3	Percentile Ranking and Citation Impact of a Large Cohort of National Heart, Lung, and Blood Institute-Funded Cardiovascular R01 Grants	20	2014	CIRCULATION RESEARCH
4	Peering at peer review revealed high degree of chance associated with funding of grant applications	18	2006	JOURNAL OF CLINICAL EPIDEMIOLOGY
5	Big names or big ideas: Do peer-review panels select the best science proposals?	17	2015	SCIENCE

Table 2.5: Top 5 papers for cluster 4 in the second scattering. The papers are ranked by number of citations. The citation counts were obtained from the citation network of the initial set of papers.

Rank	Title	Cit.	Year	Source
1	Deep impact: unintended consequences of journal rank	23	2013	FRONTIERS IN HUMAN NEUROSCIENCE
2	Alternatives to peer review: novel approaches for research evaluation	12	2011	FRONTIERS IN COMPUTATIONAL NEUROSCIENCE
3	Journal acceptance rates: A cross-disciplinary analysis of variability and relationships with journal measures	11	2013	JOURNAL OF INFORMETRICS
4	Open evaluation: a vision for entirely transparent post-publication peer review and rating for science	11	2012	FRONTIERS IN COMPUTATIONAL NEUROSCIENCE
5	Toward a new model of scientific publishing: discussion and a proposal	10	2011	FRONTIERS IN COMPUTATIONAL NEUROSCIENCE

2.4.3 Visualizing the clusters

We visualize clusters using a packed bubble chart. We developed an algorithm to create these charts (see below). The input of our algorithm is an undirected network. In this network, nodes represent clusters of papers, the weight of a node indicates the number of papers in a cluster, and the weight of an edge between two nodes indicates the relatedness of two clusters in terms of citation links.

2.4.3.1 Bubble chart algorithm

Our bubble chart algorithm determines the coordinates of the bubbles, where each bubble is a node in a network. The objective of our bubble chart algorithm is to obtain a visualization in which the bubbles do not overlap, the empty space is minimized, and the positions of the nodes relative to each other reflect their relatedness as accurately as possible. We base our algorithm on the VOS layout algorithm [119] used in the VOSviewer software, but we make modifications in order to avoid overlapping bubbles and to minimize the empty space.

The area of a node is proportional to the weight of the node. Therefore, the radius of a node is the square root of w, where w is the weight of the node. Nodes connected by edges with a high weight should be close together. To achieve this, we minimize a weighted sum of the squared Euclidean distances between all pairs of nodes, which is similar to the VOS layout algorithm [119]. The weighting considers the weight of the edges between pairs of nodes. This weighted sum can be understood as the stress V of the network layout, and our objective is to minimize this stress. Mathematically, the stress function V is given by

$$V(x_1, \ldots, x_n) = \sum_{i<j} s_{ij} \|x_i - x_j\|^2 \tag{2.1}$$

where x_i denotes the coordinates of node i in a two-dimensional space, $\| * \|$ is the Euclidean norm, and s_{ij} is the weight of the edge between nodes i and j. To avoid overlapping nodes, we add for all pairs on nodes i and j the constraint

$$\|\mathbf{x}_i - \mathbf{x}_j\| \geq r_i + r_j \tag{2.2}$$

where r_i is the radius of node i. Minimization of the stress function in Equation 2.1 subject to the constraint in Equation 2.2 is not straightforward, so we developed a minimization algorithm for it.

2.4.3.2 Minimization algorithm

The best strategy to minimize Equation 2.1 while satisfying Equation 2.2 in a network of two nodes (nodes 1 and 2) is to place the nodes adjacent to each other. When we fix the coordinates of node 1, the coordinates where node 2 can be placed form a circle $c(1, 2)$ around node 1 (Figure 2.3A). This circle has a radius equal to the sum of the radius of node 1 and the radius of node 2. Now, we also fix the coordinates of node 2 and add node 3 to the network layout. We can use the same strategy to get its coordinates. The adjacent coordinates for node 3 form the circles $c(1, 3)$ and $c(2, 3)$ (Figure 2.3B). Therefore, the available coordinates to place node 3 are the intersection points of $c(1, 3)$ and $c(2, 3)$ (Figure 2.3C).

When we add node 4 to the network layout, the available coordinates for this node are no longer all the intersection points of the circles $c(i, j)$, because some coordinates would cause nodes to overlap (Figure 2.3D). Of the available coordinates, we select the ones that result in the lowest stress. We can find these coordinates by calculating the weighted sum of the squared Euclidean distances between node 4 and each node that has already been assigned to coordinates. We proceed in the same way for all other nodes.

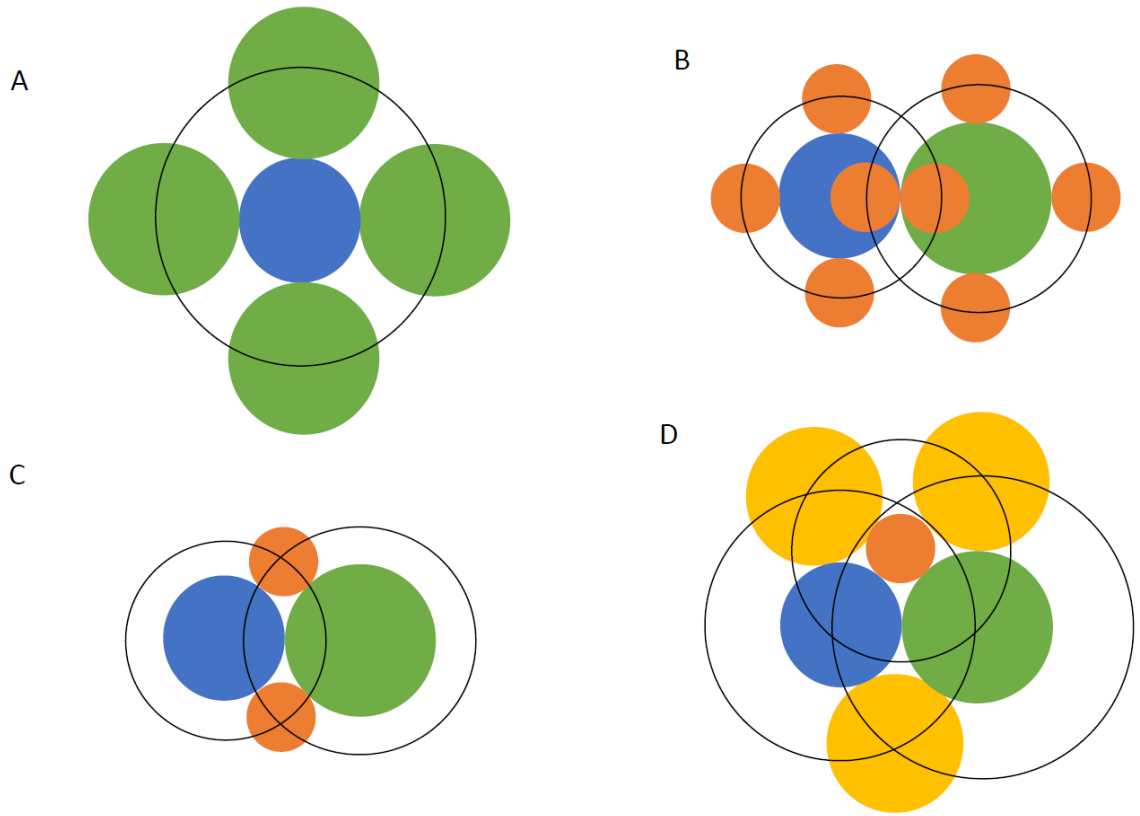

Figure 2.3: Illustration of the minimization algorithm. A: The coordinates for node 2 (green) form a circle around node 1 (blue). B: The coordinates for node 3 (orange) form a circle around node 1 (blue) and another circle around node 2 (green). C: The available coordinates for node 3 (orange) are given by the intersection of the circles in B. D: The available coordinates for node 4 (yellow) no longer include all the intersection points of the circles.

Our minimization algorithm obtains the coordinates of the nodes by adding them one-by-one to the network layout. However, we found that the value of the stress at the end of an algorithm run is highly dependent on the order in which the nodes had been added. To improve our minimization

algorithm, we added a step in which we create several lists of the nodes in a different order. For each list, we run the minimization procedure and in the end we return the network layout with the lowest stress.

We order the nodes in the lists as follows. For each node in the network, we create a list with that node as the first node. The next node in the list is the one that is most strongly related to the nodes already in the list. We repeat this process until all nodes have been added to the list.

Our minimization algorithm is a heuristic approach to the minimization of Equation 2.1 and does not guarantee that the global minimum of Equation 2.1 will be found. The pseudocode of the algorithm is provided in the appendix.

2.5 Conclusion

We have proposed a tool for scientific literature search based on the Scatter/Gather approach. The tool visualizes the structure of the search results using a packed bubble chart. We have presented a case study demonstrating the use of the tool and we have provided a technical specification of the algorithms included in the tool, in particular the algorithm for creating packed bubble charts.

Compared to traditional literature search tools that present the search results as a list of documents (e.g. Google Scholar), we expect the advantage of our tool to be in the emphasis it puts on showing the structure of the search results. We expect this to be important especially when users are searching not for one specific paper but for a larger set of papers offering a broad understanding of a certain scientific domain. In future work, we plan to test the performance of the tool for different information retrieval tasks.

2.6 Data availability

We made available a graphical user interface prototype of the tool, which we named SciMacro (for Science Macroscope) [15].

2.7 CRediT author statement

Juan Pablo Bascur: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing.
Nees Jan van Eck: Conceptualization, Methodology, Supervision, Writing – review & editing.
Ludo Waltman: Conceptualization, Methodology, Resources, Supervision, Writing – review & editing.

2.8 Appendix

```
-----
INPUT: list INLIST containing nodes (x_0,...,x_n).
    Each node possesses:
    A node identity id(x)
    A radius r(x)
    A list of edges E(x) containing (e_0,...,e_n), with each edge e possessing a weight
    w(e) and a node identity id(e) of the node it connects to
    A coordinate c(x) that contains nothing
OUTPUT: list OUTLIST containing nodes (x_0,...,x_n) possessing non-empty coordinates c(x)
-----
Create list MASTERLIST containing nothing
For each node x_i in list INLIST (x_0,...,x_n):
```

```
    Complete subroutine S_ORDER(x_i, (x_0, ..., x_n))
    Create list Z_i containing nothing
    Set coordinate c(x_i0) of node x_i0 as (0, 0)
    Append node x_i0 to list Z_i
    Set coordinate c(x_i1) of node x_i1 as ((r(x_i0) + r(x_i1), 0)
    Append node c(x_i1) to list Z_i
    Complete subroutine S_COOR(Z_i, (x_i2, ..., x_in))
    Append list Z_i to list MASTERLIST
Return list OUTLIST in MASTERLIST (Z_0,...,Z_n), where OUTLIST is the list with lowest
graph stress V as defined in the equation 2.1 V(OUTLIST)
------
Subroutine S_ORDER creates an order of nodes

S_ORDER(x_i, (x_0, ..., x_n)):
Create list X_i containing nothing
Append node x_i to list X_i as node x_i0
Create list Y_i containing nodes (x_0, ..., x_n)
Remove node x_i from list Y_i
While list Y_i containing something:
    For each node x_j in Y_i:
        Declare tw_j is the total weight from x_j to all the nodes in X_i
    Declare x_tw is the node with greatest tw_j
    Append node x_tw to list X_i as node x_ij
    Remove node x_tw from list Y_i
-----
Subroutine S_COOR gets the coordinates of the nodes for nodes x_>1

S_COOR(Z_i, (x_i2, ..., x_in)):
For each node x_ij in (x_i2, ..., x_in):
    Create empty list TEMP_ij
    For each order-independent pair of nodes (x_ijm,x_ijn) in list Z_i, where m > n:
        Complete subroutine S_TEST(x_ij, x_ijm, x_ijn, Z_i, TEMP_ij)
    Append node temp_ij to list Z_i, where temp_ij is the temporal node with lowest node
    stress v in list TEMP_ij
-----
Subroutine S_TEST tests if the node x_ij can be adjacent to nodes (x_ijm, x_ijn), get the
coordinates of center of these adjacent positions, test if the node x_ij on that coordinates
overlaps with other nodes and get the stress of the node x_ij on that coordinates.

S_TEST(x_ij, x_ijm, x_ijn, Z_i, TEMP_ij):
Declare temporary node temp_ijm with coordinate c(x_ijm) and radius (r(x_ij) + r(x_ijm))
Declare temporary node temp_ijn with coordinate c(x_ijn) and radius (r(x_ij) + r(x_ijn))
If temp_ijm and temp_ijn DO overlap:
    Declare coordinates coor_ijmn1 and coor_ijmn2 are the coordinates of the intersection
    between the borders of temp_ijm and temp_ijn
    For coor_ijmnk in list (coor_ijmn1, coor_ijmn2):
        Declare temporary node temp_ijmnk is a node with the parameters of node x_ij, except
        that its coordinate c(temp_ijmnk) is coor_ijmnk
        If node temp_ijmnk DOES NOT overlaps with any node in Z_i:
            Declare node stress v_ijmnk is the total stress of the node temp_ijmnk with
            every node in the list Z_i
        Append temp_ijmnk to list TEMP_ij
```
