Fig. 2. Schematic overview of the cluster membership assignment for instance $x_i$. Graph instances are represented by $x_1, ..., x_n$, clusters by $C_1, ..., C_k$.

- $\forall x \in V': \alpha'(x) = \alpha(x)$
- $\forall (x, y) \in V' \times V': \beta'((x, y)) = \beta((x, y))$

Given two arbitrary labeled graphs $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$, a common subgraph of $g_1$ and $g_2$, $cs(g_1, g_2)$, is a graph $g = (V, E, \alpha, \beta)$ such that there exists a subgraph isomorphism from $g$ to $g_1$ and from $g$ to $g_2$. This can be generalized to sets of graphs. The set of common subgraphs of a set of graphs $\{g_1, ..., g_n\}$ is then denoted by $cs(\{g_1, ..., g_n\})$. Moreover, given two graphs $g_1$ and $g_2$, a graph $g$ is called a maximum common subgraph of $g_1$ and $g_2$ if $g$ is a common subgraph of $g_1$ and $g_2$ and there exists no other common subgraph of $g_1$ and $g_2$ that has more vertices than $g$. Finally, we define the size of a graph as the number of its vertices, i.e. $|V|$.

2.2 Problem Definition

Structural clustering is the problem of finding groups of graphs sharing some structural similarity. Instances with similar graph structures are expected to be in the same cluster provided that the common subgraphs match to a satisfactory extent. Only connected subgraphs are considered as common subgraphs. The similarity between graphs is defined with respect to some user-defined size threshold. The threshold is set such that the common subgraphs shared among a query graph and all cluster instances make up a specific proportion of the size of each graph. A graph is assigned to a cluster provided that there exists at least one such common subgraph whose size is equal or bigger than the threshold. In this way, an object can simultaneously belong to multiple clusters (overlapping clustering) if the size of at least one common subgraph with these clusters is equal or bigger than the threshold. If an object does not share a common subgraph with any cluster that meets the threshold, this object is not included in any cluster (non-exhaustive clustering). A graphical overview is shown in Figure 2. For one graph after the other, it is decided whether it belongs to an existing cluster or whether a new cluster is created.
Formally, we frame the problem of structural clustering as follows. Given a set of graph objects \( X = \{x_1, ..., x_n\} \), we need to assign them into clusters which may overlap with each other. In clustering these objects, one objective is considered: to maximize the average number of objects contained in a cluster, such that at any time for each cluster \( C \) there exists at least one common subgraph that makes up a specific proportion, \( \theta \), of the size of each cluster member. Considering the state of a cluster \( C = \{x_1, ..., x_m\} \) at any point in time, the criterion can formally be defined as:

\[
\exists s \in cs(\{x_1, ..., x_m\}) \forall x_i \in C : |s| \geq \theta |x_i|
\]

where \( s \) is a subgraph and \( \theta \in [0, 1] \) is a user-defined similarity coefficient. According to this goal, a minimum threshold for the size of the common subgraphs shared by the query graph \( x_{m+1} \) and the instances in cluster \( C \) can be defined as

\[
\text{minSize} = \theta \max(|x_{\text{max}}|, |x_{m+1}|),
\]

where \( \theta \in [0, 1] \) and \( x_{\text{max}} \) is the largest graph instance in the cluster. To obtain meaningful and interpretable results, the minimum size of a graph considered for cluster membership is further constrained by a \( \text{minGraphSize} \) threshold. Only graphs whose size is greater than \( \text{minGraphSize} \) are considered for clustering. Thus, the identification of the general cluster scaffold will not be impeded by the presence of a few graph structures whose scaffold is much smaller than the one the majority of the cluster members share. This will be especially useful in real-world applications that often contain small fragments (see the minimum size column in Table 1).

### 2.3 Algorithm

The clustering algorithm works as follows. Let \( \text{minGraphSize} \) be the minimum threshold for the graph size and \( \text{minSize} \) be the minimum threshold for the size of the common subgraphs specified by the user and defined in Equation 2. In the first step, an initial cluster is created containing the first graph object that is larger than \( \text{minGraphSize} \). In the following steps, each instance is compared against all existing clusters. In case the query instance meets the \( \text{minGraphSize} \) threshold and shares at least one common subgraph with one or more clusters that meets the cluster criterion in Equation 2, the instance is added to the respective cluster. Unlike many traditional clustering algorithms, an object is allowed to belong to no cluster, since it is possible that an object is not similar to any cluster. Thus, in this case, a new singleton cluster is created containing the query instance. The proposed clustering algorithm has two main advantages over many clustering algorithms. First, the algorithm works in an online mode, since it does not keep all the examples in memory at the same time, but processes them one by one in a single pass. Second, in contrast to many clustering algorithms which assume that the number of clusters is known beforehand, our algorithm does not require the specification of the number of clusters a priori.

\[1\] In slight abuse of notation, we use the same indices as above.