ISSN 1064-2269, Journal of Communications Technology and Electronics, 2015, Vol. 60, No. 12, pp. 1403–1428. © Pleiades Publishing, Inc., 2015.
Original Russian Text © M.Sh. Levin, 2015, published in Informatsionnye Protsessy, 2015, Vol. 15, No. 2, pp. 215–248.

INFORMATION TECHNOLOGY IN ENGINEERING SYSTEMS

# Combinatorial Clustering: Literature Review, Methods, Examples[1]

**M. Sh. Levin**

*Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Bol'shoi Karetnyi per. 19, str. 1, Moscow, 127051 Russia*

*e-mail: mslevin@acm.org*

Received May 28, 2015

**Abstract**—The paper addresses clustering problems from combinatorial viewpoints. A systemic survey is presented. The list of considered issues involves the following: (1) literature analysis of basic combinatorial methods and clustering of very large data sets/networks; (2) quality characteristics of clustering solutions; (3) multicriteria clustering models; (4) graph based clustering methods (minimum spanning tree based clustering methods, clique based clustering as detection of cliques/quasi-cliques, correlation clustering, detection of network communities); and (5) fast clustering approaches. Mainly, the presented material is targeted to networking. Numerical examples illustrate models, methods and applications.

*Keywords*: clustering, classification, combinatorial optimization, multicriteria decision making, heuristics, network applications

**DOI:** 10.1134/S1064226915120177

## 1. INTRODUCTION

In recent decades, the significance of clustering/classification problems has been increased. Some contemporary trends in clustering researches are as follows: (a) clustering of very-large data, (b) study and implementation of fast clustering methods, and (c) multicriteria clustering models/methods. It may be reasonable to point out main types of clustering applications in communication networks, wireless sensor networks: (1) clustering and positioning/location in sensor/communication network (nodes, hubs, station) (e.g., [23, 36, 103, 211]), (2) design of hierarchal structures for networks, design of network topology (e.g., [42, 43, 81, 104, 123]), detection/assignment of cluster heads in communication/sensor networks (e.g., [211]), (3) clustering based routing, including multi-hop routing (e.g., [89, 93, 100, 103, 211]), (4) hierarchical channel access methods (e.g., [31, 89, 103, 105, 116]).

This article focuses on combinatorial clustering approaches. The following is examined: (1) literature analysis for basic combinatorial methods/models for clustering and for clustering in large scale data sets/networks; (2) quality characteristics of clustering solutions; (3) multicriteria clustering models; (4) graph based clustering methods (minimum spanning tree based clustering methods, clique based clustering, correlation clustering, network communities based clustering); (5) fast clustering approaches. Many numerical examples illustrate considered problems and methods. Two applied examples for networking are briefly described: (1) clustering scheme based on network node types, (2) clustering based scheduling in multi-beam antenna systems. The article is based on materials from electronic preprint [124]. Table 1 contains a list of basic contemporary combinatorial clustering approaches.

## 2. ON CLUSTERING IN LARGE SCALE DATA SETS/NETWORKS

In recent years, the significance of clustering in large-scale data bases and analysis and modeling in large networks has been increased, for example: (i) clustering of large data sets (e.g., [21, 91, 94, 178]); (ii) detection of communities in large networks (e.g., [39, 73, 87, 88, 117, 162, 208]); (iii) detection of communities in mega-scale networks (e.g., [22, 196]); (iv) tracking evolving communities in large networks (e.g., [88]). Table 2 illustrates some dimensional layers (classification) of data sets/networks.

## 3. QUALITY OF CLUSTERING SOLUTION

Here "hard" clustering problem is examined. Consider initial items/elements of element set $A = \{a_1, ..., a_j, ..., a_n\}$. Two types of initial information for clustering can be examined: *1.* there are m parameters/criteria and measurement of $a \in A$ is based on vector estimate $\bar{x} = (x_1, ..., x_1, ..., x_m)$; *2.* binary relation(s) over element set $A$ (including weighted binary relation(s); this is a structure over obtained clusters of a graph).

[1] The article was translated by the authors.

**Table 1.** Combinatorial approaches to clustering

| No. | Approaches, algorithmic schemes | Source(s) |
|---|---|---|
| 1 | Some surveys: | |
| 1.1 | General | [67, 96, 134] |
| 1.2 | Graph clustering | [172] |
| 1.3 | Approximate graph partitioning | [58] |
| 1.4 | Cross-entropy method for clustering, partitioning | [112, 169, 184] |
| 1.5 | Cell formation (in industrial engineering) | [72, 175, 181] |
| 1.6 | Clustering ensemble algorithms | [195] |
| 1.7 | Multicriteria classification and sorting methods | [168, 214] |
| 2 | Basic combinatorial optimization problems: | |
| 2.1 | Minimal spanning tree approach | [75, 138, 154, 160, 186, 197, 201] |
| 2.2 | Partitioning based clustering | [11, 25, 44, 52, 58, 180] |
| 2.3 | Assignment/location based clustering | [71] |
| 2.4 | Graph matching | [173] |
| 2.5 | Dominant set based clustering | [36, 81, 158, 211] |
| 2.6 | Covering based clustering | [6, 137, 171] |
| 2.7 | Clique based clustering | [6, 21, 30, 54, 70, 109, 176] |
| 2.8 | Structural clustering (detection of communities) | [4, 144, 146, 163, 202] |
| 3 | Correlation clustering | [2, 18, 51, 111, 183] |
| 4 | Graph-based data clustering with overlaps | [62] |
| 5 | Segmentation problems | [107] |
| 6 | Cluster graph modification problems | [176, 177] |
| 7 | Multi-criteria decision making in clustering-sorting | [65, 165, 166, 214] |
| 8 | Consensus clustering: | |
| 8.1 | Voting-based consensus of cluster ensembles | [12, 170] |
| 8.2 | Consensus partitions | [76] |
| 9 | Algorithmic schemes: | |
| 9.1 | Enumerative methods: | |
| 9.1.1 | Branch-and-bound methods | [38] |
| 9.1.2 | Dynamic programming | [210] |
| 9.2 | Local optimization heuristics: | |
| 9.2.1 | Simulated annealing algorithms | [29, 149, 174] |
| 9.2.2 | Tabu search algorithms | [149, 182] |
| 9.2.3 | Ant colonies algorithms | [99, 204] |
| 9.2.4 | PSO methods | [35, 185, 193] |
| 9.2.5 | Variable neighborhood search | [82, 83, 84, 85] |
| 9.3 | Genetic algorithms, evolutionary strategies | [14, 47, 90, 153, 190] |
| 9.4 | Hyper-heuristic approach | [41, 114, 189] |

**Table 2.** Dimensional layers of data sets/networks

| No. | Type of studied data sets/networks | Number of objects/network nodes | Examples of applications | Some source(s) |
|---|---|---|---|---|
| 1 | Simplified data sets/ networks (e.g., small groups) | ~10…60 | (i) student group,<br>(ii) sport club network,<br>(iii)laboratory group,<br>(iv)Web page structure,<br>(v)product assortment (product variety) | [212] |
| 2 | Simple data sets/ networks | ~100 | (i) university department,<br>(ii) animal network,<br>(iii) big firm department,<br>(iv)department of government organization,<br>(v)network of books/articles (close by topic(s)),<br>(vi) social network of bottlenose dolphins,<br>(vii) supply chain network,<br>(viii) network of software system components | [146] |
| 3 | Traditional data sets/networks | ~1 k | (i) citation networks,<br>(ii) university network,<br>(iii) collaboration network,<br>(iv) urban systems,<br>(v) consumers bases,<br>(vi) multiple server computer systems | [69] |
| 4 | Large data sets/networks | ~10 k | (i) research society network,<br>(ii) sensor networks,<br>(iii) manufacturing technology networks | [146] |
| 5 | Very large data sets/networks | ~100 k | (i) client bases,<br>(ii) VLSI,<br>(iii) medical patients bases | [39] |
| 6 | Mega-scale data sets/networks | ~1 M | (i) university library,<br>(ii) bases of editorial houses | [196] |
| 7 | Super-scale data sets/networks | ~10 M | (i) library networks,<br>(ii) Internet-based shops,<br>(iii)protein sequence databases | [22] |
| 8 | To-day's / prospective Web-based data sets/networks | ~100 M…1 B | (i) World Wide Web,<br>(ii) social networks (e.g., Twitter, Facebook) | |

Note, the first type of initial information can be transformed into the second type. A clustering solution consists of the following two parts:

(1) Clusters $\hat{X} = \{X_1, ..., X_\iota, ... X_\lambda\}$, i.e. dividing set $A$ into clusters: $X_\iota \subseteq A \; \forall \iota = \overline{1, \lambda}$; $\eta_\iota = |X_\iota|$ is the cluster size (i.e., cardinality for cluster $X_\iota$, $\iota = \overline{1, \lambda}$).

(2) Structure over clusters (if needed). Let $\Gamma(\hat{X})$ be a structure over the clusters of the clustering solution $\hat{X}$, i.e., there exists digraph $G = \hat{X}, \Gamma(\hat{X})$. Let $\Gamma(X_\iota)$ be the structure over the elements of cluster $X_\iota$ ($\forall X_\iota \in \hat{X}$).

The list of basic quality characteristics is the following (Table 3):

**1. Quality of clusters** (i.e., local quality parameters in clustering solution):

**1.1.** Intra-cluster distance (i.e., general proximity between elements in each cluster):

$$I^{\text{intra}}(X_\iota)(\iota = \overline{1, \lambda}).$$

**Table 3.** List of quality characteristics

| No. | Quality type | Notation | Description |
|---|---|---|---|
| I | Cluster | $X_\iota$ | $1 \le \iota \le \lambda$ |
| 1.1 | Intra-cluster distance | $I^{\text{intra}}(X_\iota)$ | Proximity between elements of cluster |
| 1.2 | Size of cluster | $\lvert X_\iota \rvert$ | Number of elements in cluster $X_\iota$ |
| 1.3 | Quality of cluster form | | Closeness to predefined form (e.g., ball, ellipsoid) (if needed) |
| 1.4 | Size of cluster region | | Difference between "max" and "min" coordinates (by parameters) |
| 1.5 | Quality of cluster content | | Configuration of element types (if needed) |
| 1.6 | Quality of cluster structure | | Proximity of structure over cluster elements to predefined structure (if needed) |
| II | Clustering solution | $\hat{X}$ | $\hat{X} = \{X_1, \ldots, X_\iota, \ldots, X_\lambda\}$ |
| 2.1 | Total intra-cluster quality | $Q^{\text{intra}}(\hat{X})$ | Integration of intra-cluster parameters |
| 2.2 | Total inter-cluster quality | $Q^{\text{inter}}(\hat{X})$ | Integration of inter-cluster parameters |
| 2.3 | Number of clusters ($\lambda$) | $Q^{\text{num}}(\hat{X})$ | Number of clusters in clustering solution |
| 2.4 | Closeness to cluster size | $Q^{\text{bal}}(\hat{X})$ | Balance by cluster size, closeness to predefined balance vector |
| 2.5 | Quality by forms of clusters | $Q^{\text{form}}(\hat{X})$ | Integration of cluster form parameters |
| 2.6 | Parameter of cluster regions | $Q^{\text{reg}}(\hat{X})$ | Integration of cluster regions sizes (by coordinates) |
| 2.7 | "Correlation clustering functional" | $Q^{\text{corr}}(\hat{X})$ | Integration of maximum agreement (in each cluster) and minimum disagreements (between clusters) [17, 18] |
| 2.8 | Quality of modularity | $Q^{\text{mod}}(\hat{X})$ | Parameter of network modularity [69, 144] |
| III | Quality of structure over clusters | $Q^{\text{struc}}(\hat{X})$ | Closeness to predefined structure |
| IV | Multicriteria quality | $\overline{Q}(\hat{X})$ | Integrated vector of quality e.g., $\overline{Q}(\hat{X}) = (Q^{\text{intra}}(\hat{X}), Q^{\text{inter}}(\hat{X}), Q^{\text{bal}}(\hat{X}))$ |

*Version 1.* Quantitative parameter as integration of quantitative element proximities (distances) in the cluster.

*Version 2.* Multiset parameter as integration of ordinal estimates of element proximities [121, 123]. The approach is illustrated by example.

**Example 1.** Example for three clusters is depicted in Fig. 1: $X_1 = \{1, 2, 3, 4\}$, $X_2 = \{5, 6, 7\}$, $X_3 = \{8, 9, 10, 11, 12\}$. Ordinal scale [1, 2, 3] for estimates of element similarity is used:

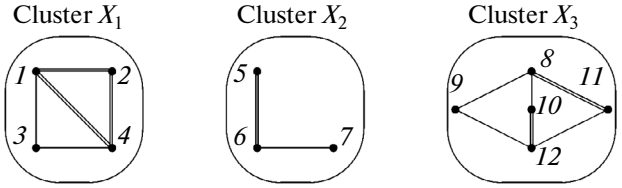



**Fig. 1.** Local intra-cluster quality (for cluster).

1 corresponds to "very similar",

2 corresponds to "medium level",

3 corresponds to "very different" (in this case the edge in Fig. 1 is absent).

Ordinal proximities of edges are presented in Table 4. The resultant multiset intra-cluster parameters for clusters are: $I^{\text{intra}}(X_1) = (2, 3, 1)$, $I^{\text{intra}}(X_2) = (1, 1, 1)$, $I^{\text{intra}}(X_3) = (4, 2, 4)$.

**1.2.** Number of elements in cluster (or in each cluster, i.e., cluster size) corresponds to constraints, for example: $\pi^- \le \eta_\iota = \lvert X_\iota \rvert \le \pi^+$ ($\pi^-$, $\pi^+$ are predefined limits of the cluster size) ($\forall X_\iota \in \hat{X}$).

The quality parameter corresponds to external requirement (from the viewpoint of applied problem(s), e.g., teams, communication systems).

**1.3.** Quality of cluster form (e.g., body, envelope, cover), for example: sphere/ball, ellipsoid, globe (i.e., closeness to the required cluster form).

**1.4.** Quality as constraint for size of cluster region (limits for interval for coordinates of cluster elements). Let us consider cluster $X = \{x_1, ..., x^{\xi}, ..., x^{\phi}\}$, parameter estimates of each cluster element $x^{\xi}$ are (vector estimate, parameters $i = \overline{1, m}$): $\overline{x^{\xi}} = (x_1^{\xi}, ..., x_i^{\xi}, ..., x_m^{\xi})$. Constraints are (by each parameter $\forall i = \overline{1, \phi}$) (Fig. 2):

$$\left| \max_{\xi = \overline{1, \phi}} x_i^{\xi} - \min_{\xi = \overline{1, \phi}} x_i^{\xi} \right| \le d_i, \quad \forall i = \overline{1, m}.$$

The quality parameter corresponds to external requirement (from the viewpoint of applied problem(s), e.g., communication systems).

**1.5.** Quality of the cluster contents/structure (if needed), for example (a composite "team"): 1 element of the 1st type, 3 elements of the 2nd type, 2 elements of the 3rd type, 1 element of the 4th type. Here proximity of the obtained cluster content to the required content can be considered.

**1.6.** Quality of cluster structure (if needed) for cluster $X_\iota$ ($\forall X_\iota \in \hat{X}$), i.e., proximity $\delta(\Gamma(X_\iota), \Gamma^0(X_\iota))$, where $\Gamma^0(X_\iota)$ is the predefined structure over the cluster elements.

**2. Total quality for clustering solution** (i.e., for cluster set):

**2.1.** Total intra-cluster quality for clustering solution $Q^{\text{intra}}(\hat{X})$ is an integrated measure of intra-cluster parameters $\{I^{\text{intra}}(X_\iota))$ of all clusters in clustering solution (i.e., $\iota = \overline{1, \lambda}$).

*Version 1.* Total qualitative quality parameter for qualitative local estimates:

$$Q^{\text{intra}}(\hat{X}) = \frac{1}{\lambda} \sum_{\iota = \overline{1, \lambda}} I^{\text{intra}}(X_\iota).$$

Note, integration process can be based on summarization and some other operations (maximization, minimization, etc.).

**Table 4.** Ordinal proximities (intra-cluster, edge $\{i_1, i_2\}$)

| $i_1$ | $i_2$: | 2 | 3 | 4 | 6 | 7 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 2 | 1 | 2 | | | | | | |
| 2 | | | 3 | 2 | | | | | | |
| 3 | | | | 1 | | | | | | |
| 5 | | | | | 2 | 3 | | | | |
| 6 | | | | | | 1 | | | | |
| 8 | | | | | | | 1 | 1 | 2 | 3 |
| 9 | | | | | | | | 3 | 3 | 1 |
| 10 | | | | | | | | | 3 | 2 |
| 11 | | | | | | | | | | 1 |

**Table 5.** Ordinal proximities (intra-cluster, edge $(i_1, i_2)$)

| $i_1$ | $i_2$: | 2 | 3 | 5 | 6 | 8 | 9 |
|---|---|---|---|---|---|---|---|
| 1 | | 2 | 2 | | | | |
| 2 | | | 1 | | | | |
| 4 | | | | 2 | 1 | | |
| 5 | | | | | 1 | | |
| 7 | | | | | | 2 | 1 |
| 8 | | | | | | | 1 |

*Version 2.* Total multiset quality parameter for multiset local estimates. The approach is illustrated by example.

**Example 2.** Example for three clusters is depicted in Fig. 3 (for simplification the cardinality of clusters is the same): $X_1 = \{1, 2, 3\}$, $X_2 = \{4, 5, 6\}$, $X_3 = \{7, 8, 9\}$; clustering solution is: $\hat{X} = \{X_1, X_2, X_3\}$. Ordinal scale [1, 2, 3] for estimates of element similarity is used: 1 corresponds to "very similar", 2 corresponds to "medium level", 3 corresponds to "very different" (in this case the corresponding edge is absent). Ordinal proximities of edges in clusters are presented in Table 5.

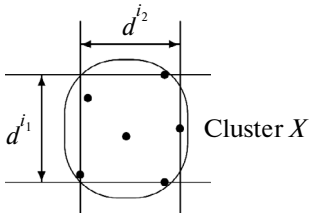



**Fig. 2.** Size of cluster region.

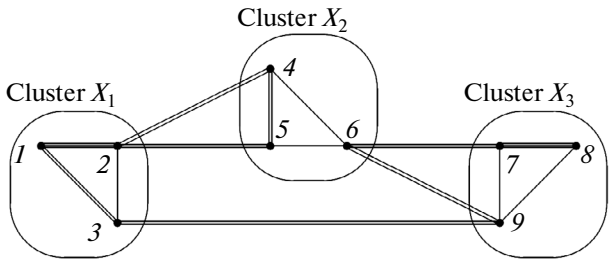



**Fig. 3.** Intra- and inter-cluster qualities.

**Table 6.** Ordinal proximities (inter-cluster, edge $(i, j)$)

| $i$ | $j$: | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|
| 1 | | 3 | 3 | 3 | 3 | 3 | 3 |
| 2 | | 2 | 2 | 3 | 3 | 3 | 3 |
| 3 | | 3 | 3 | 3 | 3 | 3 | 2 |
| 4 | | | | | 3 | 3 | 3 |
| 5 | | | | | 3 | 3 | 3 |
| 6 | | | | | 2 | 3 | 2 |

The resultant multiset intra-cluster parameters for clusters are:

$$I^{\text{intra}}(X_1) = (1, 2, 0),$$

$$I^{\text{intra}}(X_2) = (2, 1, 0), \quad I^{\text{intra}}(X_3) = (2, 1, 0).$$

Integration of the above-mentioned intra-cluster multiset estimates can be based on two methods (e.g., [121, 123]):

(a) summarization (by the vector components): $Q^{\text{intra}}(\hat{X}) = (5, 4, 0)$, the obtained integrate estimate corresponds to an extended lattice;

(b) searching for a median multiset estimate: $Q^{\text{intra}}(\hat{X}) = (2, 1, 0)$.

**2.2.** Total inter-cluster quality for clustering solution $Q^{\text{inter}}(\hat{X})$ is an integrated measure of inter-cluster parameters ($I^{\text{intra}}(X_{\iota_1}, X_{\iota_2})$) of all cluster pairs in clustering solution (i.e., $\iota_1 = \overline{1, \lambda}$, $\iota_2 = \overline{1, \lambda}$, $\iota_1 \neq \iota_1$).

*Version 1.* Total qualitative quality parameter for qualitative local estimates as integration of all qualitative two-cluster inter-cluster proximities/distances:

$$Q^{\text{inter}}(\hat{X}) = \frac{1}{\lambda(\lambda - 1)} \sum_{\iota_1 = \overline{1, \lambda}, \iota_2 = \overline{1, \lambda}, \iota_1 \neq \iota_2} I^{\text{inter}}(X_{\iota_1}, X_{\iota_2}).$$

Note, integration process can be based on summarization and some other operations (maximization, minimization, etc.).

*Version 2.* Total multiset quality parameter for multiset local estimates. The approach is illustrated by example.

**Example 3.** The example is based on data from previous example 2 (i.e., Fig. 3). Table 6 contains inter-cluster ordinal proximities.

Inter-cluster multiset estimates are:

$$I^{\text{inter}}(X_1, X_2) = (0, 2, 7),$$

$$I^{\text{inter}}(X_1, X_3) = (0, 1, 8), \quad I^{\text{inter}}(X_2, X_3) = (0, 2, 7).$$

Integration of the above-mentioned intra-cluster multiset estimates can be based on two methods (e.g., [121, 123]):

(a) summarization (by the vector components): $Q^{\text{inter}}(\hat{X}) = (0, 5, 22)$;

(b) searching for a median multiset estimate: $Q^{\text{inter}}(\hat{X}) = (0, 2, 7)$.

**2.3.** Total number of clusters in clustering solution $Q^{\text{num}}(\hat{X})$, for example: $\Upsilon^- \leq \lambda(\hat{X}) \leq \Upsilon^+$ ($\Upsilon^-$, $\Upsilon^+$ are predefined limits of the total cluster number). The quality parameter corresponds to external requirement (from the viewpoint of applied problem(s)). This is connected to 1.2.

**2.4.** Closeness of element cluster sizes in clustering solution to the predefined cluster size constraints, i.e., balance (or imbalance) of cluster cardinalities $Q^{\text{bal}}(\hat{X})$, for example: $\pi^- \leq |X_\iota)| \leq \pi^+$ ($\pi^-$, $\pi^+$ are general limits of each cluster size. Evidently, here the balance/imbalance (i.e., out-of-balance) estimate of a clustering solution can be consider as the number of clusters that corresponds (or does not correspond) to the constraints. The estimates can be examined as a vector-like estimate or a multiset estimate, for example: the number of "good" clusters (with "good/right" cluster size), the number of quasi-good clusters (with quasi-right cluster size), and the number of other clusters. This parameter corresponds to external requirement (from the viewpoint of applied problem(s)). Now let us describe the version of the vector-like estimate. The notations are as follows: (a) $\pi^0(\hat{X})$ is the number of clusters in $\hat{X}$ in which the cluster size $X_\iota$ complies with the predefined limits, (b) $\pi^{+l}(\hat{X})$ is the number of clusters in $\hat{X}$ where the cluster size $X_\iota$ more then $\hat{\pi}^+$ (upper limit) by $l$ elements, (c) $\pi^{-l}(\hat{X})$ be the number of clusters in $\hat{X}$ where the cluster size $X_\iota$ less then $\hat{\pi}^-$ (bottom limit) by $l$ elements. As a result, the following vector estimate can be considered:

$$Q^{\text{bal}}(\hat{X}) = (\pi^{l-}(\hat{X}), \ldots, \pi^{-1}(\hat{X}), \pi^0(\hat{X}), \pi^1(\hat{X}), \ldots, \pi^{l+}(\hat{X})).$$

Note, a close type of the vector estimate (vector proximity) has been suggested for comparison of rankings in [118]. The approach is illustrated by example.

**Example 4.** Initial set of objects is: $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$, clustering solution is: $\hat{X}$: $X_1 = \{1, 5, 7\}$, $X_2 = \{2\}$, $X_3 = \{3, 6, 10, 13, 17\}$, $X_4 = \{11, 12\}$, $X_5 = \{4, 12, 14, 15\}$, $X_6 = \{8, 16\}$. The following constrains for cluster size are considered: $\hat{\pi}_1 = 2$, $\hat{\pi}_2 = 3$. Vector estimate for balance of cluster cardinalities is: $Q^{\text{bal}}(\hat{X}) = (\pi^{-1}(\hat{X}), \pi^0(\hat{X}), \pi^1(\hat{X}), \ldots, \pi^2(\hat{X})) = (1, 3, 1, 1)$.

The considered approach is close to Y-balanced partitioning (clustering solution $\hat{X}$) when size of each

obtained cluster $|X_\iota| \approx \frac{n}{\Upsilon(\hat{X})} (\forall X_\iota \in \hat{X})$ where $\Upsilon(\hat{X})$ (i.e., $\lambda$) is the number of obtained clusters.

**2.5.** Total quality for balance (or imbalance) of cluster forms (i.e., cluster bodies/covers) in a clustering solution $Q^{\text{form}}(\hat{X})$, for example: majority of clusters of a clustering solution have the same (or about the same) bodies (e.g., spheres/balls, ellipsoids, globes).

Evidently, it is possible to consider a measure of imbalance, analogically as in parameter 2.3.

**2.6.** Total quality $Q^{\text{reg}}(\hat{X})$ as constraints for size of cluster regions (limits for interval of cluster element coordinates for each cluster). Let us consider cluster $X_\iota = \{x^{\iota,1}, ..., x^{\iota,\xi}, ..., x^{\iota,\phi_\iota}\}$. Parameter estimates of each cluster element $x^{\iota,\xi}$ are (vector estimate, parameters $\iota = \overline{1,m}$ and clusters $\iota = \overline{1,\lambda}$): $\overline{x^{\iota,\xi}} = (x_1^{\iota,\xi}, ..., x_i^{\iota,\xi}, ..., x_m^{\iota,\xi})$. Constraints are (by each parameter $\forall i = \overline{1,\lambda}$) (Fig. 2):

$$\left| \max_{\xi = 1,\phi_\iota} x_i^\xi - \min_{\xi = 1,\phi_\iota} x_i^\xi \right| \le d_i, \quad \forall i = \overline{1,m}, \ \forall \iota = \overline{1,\lambda}.$$

The quality parameter corresponds to external requirement (from the viewpoint of applied problem(s), e.g., communication systems).

**2.7.** The "correlation clustering functional" to maximize the intra-cluster agreement (attraction) and the inter-cluster disagreement (repulsion) has been proposed in [17, 18] ($Q^{\text{corr}}(\hat{X})$). Here, partitioning a fully connected labeled graph is examined (label "+" corresponds to edge between similar vertices, label "–" corresponds to edge between different vertices). The optimization functional $Q^{\text{corr}}(\hat{X})$ is an integration (i.e., summarization) of two components: (i) the maximizing number of "–" edges between clusters (i.e., minimizing disagreements), (b) the number of "+" edges insides the clusters (i.e., maximizing agreements) (e.g., [2, 15, 17, 18, 51, 111, 183]). Weighted versions of the "correlation clustering functional" are considered as well (e.g., [32, 33, 51]).

**2.8.** Modularity of clustering solution $Q^{\text{mod}}(\hat{X})$ is defined as follows (e.g., [69, 142, 144, 146]) (Fig. 4). Let $G = (A, E)$ be an initial graph, where $A$ is the set of nodes, $E$ is the set of edges. Clustering solution for graph $G$ is: $\hat{X} = \{X_1, ..., X_\iota, ..., X_\lambda\}$. Let $A^\iota$ be the set of nodes in cluster $X_\iota$ ($\iota = \overline{1,\lambda}$). Let $E^\iota$ be the set of internal edges in cluster $X_\iota$ ($\iota = \overline{1,\lambda}$), i.e., all corresponding nodes belong to $A^\iota$. Let $\tilde{E}^\iota$ be the set of external edges for cluster $X_\iota$ ($\iota = \overline{1,\lambda}$), i.e., the only one corresponding node belong to $A^\iota$. The definitions are illustrated in Fig. 4 for a four cluster solution (cluster $X_3$). Thus, the following parameters for each cluster $X_\iota$ can be used:

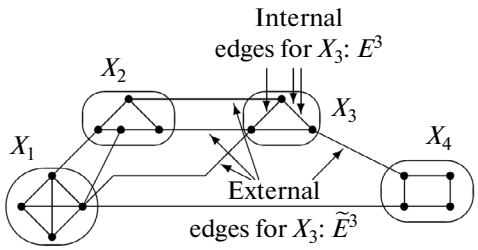



**Fig. 4.** Modularity in graph clustering.

(a) $e_\iota = \frac{|E^\iota|}{|E|}$ (% edges in module $\iota$),

(b) $a_\iota = \frac{|\tilde{E}^\iota| + |E^\iota|}{|E|}$ (% edges with at least one end in module $\iota$).

Further, general modularity of clustering solution for graph $G$ is:

$$Q^{\text{mod}}(\hat{X}) = \sum_{\iota=1}^{\lambda} (e_\iota - (a_\iota)^2).$$

Clustering problem to maximize the modularity is NP-hard [27]. The approach is illustrated by example.

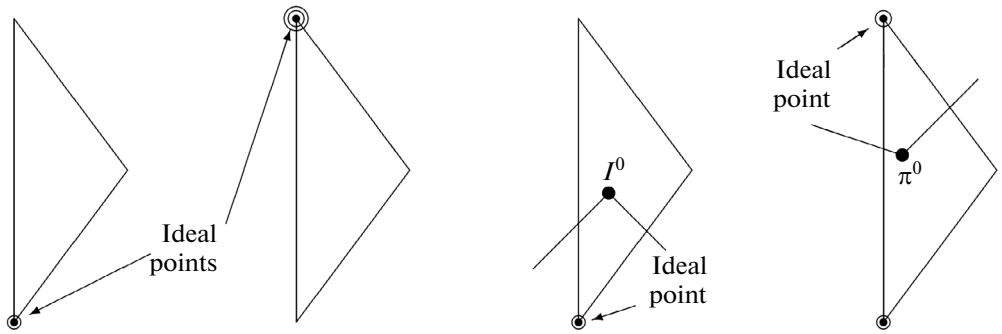
**Example 5.** Let us consider modularity parameters for clustering solution from Fig. 4. Here, $|E| = 26$. Parameters for clusters are:

(1) $|E^1| = 6$, $|\tilde{E}^1| = 4$, $e_1 = 0.23$, $a_1 = 0.38$; (2) $|E^2| = 4$, $|\tilde{E}^2| = 4$, $e_2 = 0.15$, $a_2 = 0.3$; (3) $|E^3| = 3$, $|\tilde{E}^3| = 4$, $e_3 = 0.115$, $a_3 = 0.27$; (4) $|E^4| = 4$, $|\tilde{E}^4| = 2$, $e_4 = 0.15$, $a_4 = 0.23$. The resultant modularity parameter for clustering solution is: $Q^{\text{mod}}(\hat{X}) = (0.23 - 0.14) + (0.15 - 0.09) + (0.115 - 0.073) + (0.15 - 0.053) = 0.09 + 0.06 + 0.42 + 0.097 = 0.667$.

**3. Quality of structure over clusters** (e.g., tree, hierarchy) (if needed) ($Q^{\text{struc}}(\hat{X})$). Here a proximity of the obtained structure $\Gamma(\hat{X})$ in clustering solution $\hat{X}$ and a predefined structure $\Gamma^0$ is examined: $Q^{\text{struc}}(\hat{X}) = \delta(\Gamma(\hat{X}), \Gamma^0)$. Clearly, various scales for assessment of the proximities can be used (e.g., qualitative, ordinal, vector-like, multiset) (e.g., [120, 123]).

**4.** Generally, it is reasonable to consider multicriteria quality of clustering solutions that integrates the above-mentioned clustering characteristics, for example:

$$Q(\hat{X}) = (Q^{\text{inter}}(\hat{X}), Q^{\text{intra}}(\hat{X}), \pi(\hat{X})).$$




(a) Total intra-cluster quality $Q^{\text{intra}}(\hat{X})$ (b) Total inter-cluster quality $Q^{\text{intra}}(\hat{X})$ (c) Local intra-cluster quality $I^{\text{intra}}(X_\iota)$, $\forall_\iota$ (d) Total balance by cluster sizes $Q^{\text{bal}}(\hat{X})$

**Fig. 5.** Illustration for quality posets (lattices).

As a result, the clustering problem can be formulated as generalized multicriteria optimization problem (i.e., Pareto-efficient solutions have to be searched for), for example:

$$\min Q^{\text{intra}}(\hat{X}), \quad \max Q^{\text{inter}}(\hat{X})$$
$$s.t.\ Q^{\text{bal}}(\hat{X}) \preccurlyeq \pi^0, \quad Q^{\text{struc}} = \delta(\Gamma(\hat{X}), \Gamma^0) \le \delta^0.$$

In the case of multiset estimates, the multiple criteria optimization clustering problem can be considered on the basis of quality lattices (poset-like scales) as follows (i.e., Pareto-efficient solutions over posets have to be searched for):

$\min Q^{\text{intra}}(\hat{X})$ (by lattice, Fig. 5a)

$\max Q^{\text{inter}}(\hat{X})$ (by lattice, Fig. 5b)

*s.t.* $I^{\text{intra}}(X_\iota) \succcurlyeq I^0$, $\forall \iota = \overline{1, \lambda}$, $I^0$ is reference multiset estimate $\forall X_\iota$ (by lattice, Fig. 5c),

$Q^{\text{bal}}(\hat{X}) \preccurlyeq \pi^0$, $\pi^0$ is reference multiset estimate (balance by cluster size, by lattice, Fig. 5d),

$Q^{\text{struc}} = \delta(\Gamma(\hat{X}), \Gamma^0) \le \delta^0$, (closeness to predefined general structure $\Gamma^0$).

Thus, Fig. 5 illustrates the integrated "discrete space" (poset) for total multiset based vector quality of clustering solution $\hat{X}$.

In the case of "soft" clustering problems, it is necessary to examine measures of solution "softness" (e.g., total parameter for intersection of clusters).

## 4. GRAPH BASED CLUSTERING

### *4.1. Minimum Spanning Tree Based Clustering*

The preliminary building of minimum trees is widely used in many combinatorial problems (e.g., [67]). The algorithmic complexity estimate for this spanning problem over graph equals $O(n\log n)$ ($n$ is the number of graph vertices). Minimum spanning tree based clustering algorithms have been studied and applied by many researchers (e.g., [74, 75, 113, 138, 154, 160, 181, 197, 201]). The basic stages of the algorithms are as follows:

*Stage 1.* Calculation of distance/proximity matrix $Z$.

*Stage 2.* Design of the corresponding graph $G$.

*Stage 3.* Building of the minimum spanning tree $T$ for graph $G$.

*Stage 4.* Clustering of the vertices of tree $T$ (e.g., by algorithm of deletion of branches, by algorithm of hierarchical clustering).

*Stage 5.* Stopping.

Further, the usage of hierarchical clustering at stage 4 is considered. Complexity estimates for minimum spanning tree clustering algorithm (by stages) are presented in Table 7.

**Table 7.** Complexity estimates of stages for minimum spanning tree based clustering

| Stage | Description | Complexity estimate (running time) |
|---|---|---|
| Stage 1 | Calculate distance matrix $Z$ | $O(n^2)$ |
| Stage 2 | Design the corresponding graph $G$ | $O(n^2)$ |
| Stage 3 | Building the minimum spanning tree | $O(n\log n)$ |
| Stage 4 | Clustering of the tree vertices | $O(n\log n)$ |
| Stage 5 | Stopping | $O(1)$ |

**Table 8.** Proximities for example (edge $(i_1, i_2)$)

| $i_1$ | $i_2$: | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.3 | 1.4 | 1.45 | ★ | ★ | ★ | ★ | ★ | ★ | ★ | ★ |
| 2 | | | 0.3 | ★ | ★ | 2.6 | 0.2 | 1.8 | ★ | ★ | ★ | ★ |
| 3 | | | | 0.4 | ★ | ★ | 1.65 | 0.25 | ★ | ★ | ★ | ★ |
| 4 | | | | | 0.4 | ★ | ★ | 0.45 | 1.9 | ★ | ★ | ★ |
| 5 | | | | | | ★ | ★ | ★ | 0.35 | 1.5 | ★ | ★ |
| 6 | | | | | | | 0.1 | ★ | ★ | ★ | 1.4 | ★ |
| 7 | | | | | | | | 0.41 | ★ | ★ | 0.4 | ★ |
| 8 | | | | | | | | | 0.9 | ★ | 2.1 | ★ |
| 9 | | | | | | | | | | 0.15 | ★ | 0.5 |
| 10 | | | | | | | | | | | ★ | 2.0 |
| 11 | | | | | | | | | | | | 2.5 |

Stages 3, 4, 5 correspond to the situation when a graph is examined as initial data. In this case, complexity of the algorithm equals $O(n\log n)$.

**Example 6.** Initial set of objects is: $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. Table 8 contains proximity matrix (symbol "★" corresponds to a very big value).

Figure 6 depicts corresponding graph, $G = (A, E)$, Figure 7 depicts spanning tree $T = (A, E')$, and clustering solution. Generally, it is reasonable to point out threshold based modification of graph $G = (A, E)$ over object set $A$: deletion of edges by condition: the weight ">" the threshold. Decreasing the threshold leads to decreasing the cardinality of $E$. This process can be very useful for analysis and processing of initial data in clustering. Let us consider an illustration of the above-mentioned process on the basis graph from example 6 (basic proximity matrix from Table 8): (i) threshold equals 2.6: graph $G = (A, E)$ in Fig. 6; (ii) threshold equals 1.4: graph $G^1 = (A, E^1)$ in Fig. 8; (iii) threshold equals 0.5: graph $G^2 = (A, E^2) = T = (A, E')$ in Fig. 7 (here: spanning tree); (iv) threshold equals 0.3: graph $G^3 = (A, E^3)$ in Fig. 9.

As a result, a useful structure can be found. The described by example procedure is an auxiliary problem. Another significant problem consists in analysis of the obtained graph: (a) connectivity, (b) similarity to tree (or hierarchy, clique). Now, a modified version of adaptive minimum spanning tree clustering algorithm is examined. Initial data are: (a) set of objects/alternatives $A = \{A_1, ..., A_i, ..., A_n\}$, (b) set of parameters/criteria $\bar{C} = \{C_1, ..., C_j, ..., C_m\}$, (c) estimate matrix $X = \{x_{ij}\}$, $i = \overline{1, n}$, $j = \overline{1, m}$, $x_{ij}$ is qualitative estimate of $A_i$ on criterion $C_j$. The algorithm is:

*Stage 1.* Calculation of proximity matrix $Z = \{z_{ik}\}$, $i = \overline{1, n}$, $k = \overline{1, n}$, where $z_{ik}$ is estimate of proximity (distance) between $A_i$ and $A_k$ (Euclidean metric is used). Evidently, $z_{ii} = 0$, $\forall i = \overline{1, n}$.

*Stage 2.* Transformation of matrix $Z$ into ordinal matrix $Y = \{y_{ik}\}$. Let us consider the maximum and minimum values of elements of matrix $Z$:

$$z^{\min} = \min_{\forall i = \overline{1, n}, i = \overline{1, k}} \{z_{ik}\},$$

$$z^{\max} = \max_{\forall i = \overline{1, n}, i = \overline{1, k}} \{z_{ik}\}.$$

Thus an interval is obtained $[z^{\min}, z^{\max}]$ and $d = z^{\max} - z^{\min}$. Now an additional integer parameter $\delta$ (e.g., 3, 4, 5, 6) is used. Let $\delta = 5$. Then elements of

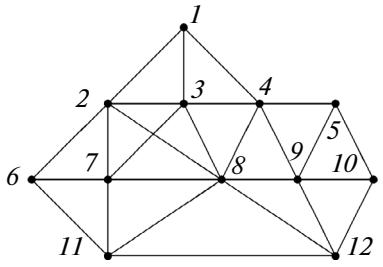



**Fig. 6.** Graph $G = (A, E)$.

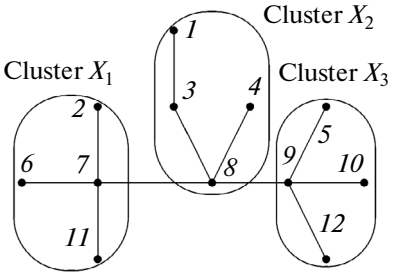



**Fig. 7.** Spanning tree $T = (A, E')$.

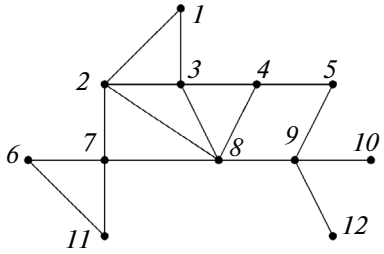



**Fig. 8.** Graph $G^1 = (A, E^1)$.

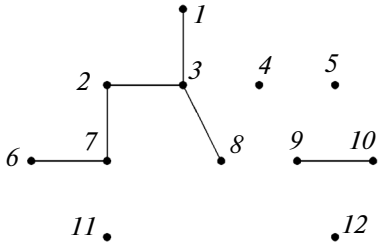



**Fig. 9.** Graph $G^3 = (A, E^3)$.

new matrix $Y$ (i.e, adjacency matrix) are based on the following calculation:

$$y_{ik} = \begin{cases} 0, & \text{if } 0.0 \le z_{ik} \le d/\delta, \\ 1, & \text{if } d/\delta < z_{ik} \le 2d/\delta, \\ 2, & \text{if } 2d/\delta < z_{ik} \le 3d/\delta, \\ 3, & \text{if } 3d/\delta < z_{ik} \le 4/\delta, \\ 4, & \text{if } 4d/\delta < z_{ik} \le d. \end{cases}$$

*Stage 3.* Obtaining an interconnected graph over elements $A$ (iterative approach):

Let $\Delta = 1, 2,...$ be an integer algorithmic parameter (for the algorithm cycle).

*Step 3.1.* Initial value $\Delta = 1$.

*Step 3.2.* Transformation of ordinal matrix $Y$ into Boolean matrix $B = \{b_{ik}\}$:

$$b_{ik} = \begin{cases} 1, & \text{if } y_{ik} < \Delta, \\ 0, & \text{if } y_{ik} \ge \Delta. \end{cases}$$

*Step 3.3.* Building a graph over elements $A$: $G^\Delta = (A, \Gamma^\Delta)$, where $\Gamma^\Delta$ is the set of edges, edge $(A_i, A_k)$ exists if $b_{ik} = 1$.

*Step 3.4.* Analysis of connectivity for graph $G^\Delta = (A, \Gamma^\Delta)$. If the graph is connected, then GOTO *Step 3.6.*

*Step 3.5.* $\Delta = \Delta + 1$ and GOTO *Step 3.2.*

*Step 3.6.* Building of minimum spanning tree for graph $G^\Delta = (A, \Gamma^\Delta)$: $T^\Delta = (A, \hat{E}^\Delta)$. Here, several well-known algorithms can be used, for example: Borovka's algorithm Prim's algorithm, Kruskal's algorithm [7, 66, 67, 45, 161, 209]. Complexity estimate of the algorithms is: $O(p\log n)$ (or less [209]) ($p$ is the number of edges, $n$ is the number of vertices).

*Step 3.7.* Clustering set $A$ on the basis of spanning tree $T^\Delta = (A, \hat{E}^\Delta)$ while taking into account an algorithmic parameter: a number of elements a in each obtained cluster $\alpha' \le \alpha \le \alpha''$, for example $\alpha' = 4$, $\alpha'' = 6$. The constrains above have to be based on applied analysis.

*Stage 4.* Stop.

Complexity estimates for the described adaptive algorithm (by stages) is presented in Table 9.

The general complexity estimate (running time) of the described adaptive algorithm equals $O(n^2)$.

**Table 9.** Complexity estimates for adaptive minimum spanning tree based algorithm

| Stage/step | Description | Complexity estimate |
|---|---|---|
| Stage 1 | Calculation of distance matrix $Z$ | $O(n^2)$ |
| Stage 2 | Transformation of matrix $Z$ into ordinal matrix $Y$ | $O(n^2)$ |
| Stage 3 | Design of interconnected graph over elements $A$ | $O(n^2)$ |
| Step 3.1 | Specifying the start of the cycle | $O(1)$ |
| Step 3.2 | Transformation of matrix $Y$ into Boolean matrix $B$ | $O(n^2)$ |
| Step 3.3 | Building of graph $G$ that corresponds to matrix $B$ | $O(n^2)$ |
| Step 3.4 | Analysis of connectivity of graph $G$ | $O(n)$ |
| Step 3.5 | Correction of cycle parameter | $O(1)$ |
| Step 3.6 | Building of minimum spanning tree $T$ for graph $G$ | $O(n\log n)$ |
| Step 3.7 | Clustering of vertices of spanning tree $T$ (limited cluster size) | $O(n)$ |
| Stage 4 | Stopping | $O(1)$ |

Generally, the problem of $k$-balanced partitioning a tree is NP-hard ($k$ is the number of elements in each cluster of clustering solution) [61].

Note, the obtained clustering solution has a property: "modularity". This can be very important for many applied problems (e.g., close cardinalities of clusters/groups: local region elements in communication network, student teams).

### *4.2. Clique Based Clustering*

Here, an initial graph $G = (A, E)$ is examined as initial data. In a clique (complete graph/subgraph), each vertex is connected to all other the vertices $A$ quasi-clique can be examined, for example, as a clique without one-two edges. The cliques (or quasi-cliques) form a very strong clusters (from the viewpoint of interconnection). The problem of finding a maximal clique in a graph is a well-known NP-hard problem (e.g., [67, 101]). Thus, heuristics or enumerative methods have been used for the problem. Clique-based clustering process can be organized as a series of clique problems [67]:

*Stage 1.* Finding the "maximal clique" (or maximal "quasi-clique") in graph $G = (A, E)$: subgraph $H = (B, V)$ ($H \subseteq A$, $V \subseteq E$).

*Stage 2.* Forming a cluster from subgraph $H$ and compression of initial graph $G$: $G' = (A', E')$, ($A' = A \backslash H$, $E' = E \backslash \{V \cup W\}$, where $W$ is a set of external edges of clique, i.e., the only one vertex belongs to set $H$) (Fig. 10).

*Stage 3.* If $G'$ is empty GO TO Stage 4 otherwise GO TO Stage 1.

*Stage 4.* Stop.

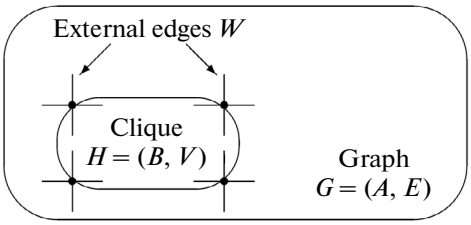



**Fig. 10.** Illustration of clique in graph.

The above-mentioned solving scheme is based on series of NP-hard problems. Evidently, it is possible to find several "maximal cliques" concurrently. Some sources on researches on clique finding and clique based clustering are presented in Table 10.

Clique partitioning problem for a given graph $G = (A, E)$ with edge weights consists in partitioning the graph into cliques such that the sum of the edge weights over all cliques formed is as large as possible (e.g., [109, 148]).

There are some close problems over graphs/digraphs, for example, independent set problems and dominating set problems which are used in clustering as well (e.g., [36, 42, 43, 46, 81, 92, 158]). Recently, the significance of dynamic problems over data streams has been increased including clique/quasi-clique finding in graph streams (e.g. [5, 40, 77, 122]).

On the other hand, clique-based approaches can be considered as density-based and grid-based clustering methods. In some recent works, subgraph as clique/qiasi-clique is considered as one of network community structures (network community based clustering [69, 144, 145, 163]).

**Table 10.** Detection of cliques/quasi-cliques and clustering

| No. | Research | Source(s) |
|---|---|---|
| 1 | Detection and analysis of cliques in graphs: | |
| 1.1 | Cliques in graphs | [98, 136] |
| 1.2 | Finding of clique/quasi-clique in graphs | [1, 10, 30, 59, 67, 150, 156] |
| 1.3 | Maximum-weight clique problem | [13, 16, 26, 151, 155] |
| 1.4 | Finding all cliques of undirected graph | [28] |
| 1.5 | Enumeration maximal cliques of large graph | [9] |
| 2 | Clustering based on cliques: | |
| 2.1 | Clique based clustering | [6, 21, 54, 70, 96, 131, 134, 176, 191] |
| 2.2 | Clique partitioning problem (clique partition of maximum weight) | [49, 109, 148] |
| 3 | Clique-based multiple clustering: | |
| 3.1 | Ensemble clustering with voting active clusters | [192] |
| 3.2 | Cliques for combining multiple clusterings | [133] |
| 4 | Clique based methods over data streams: | |
| 4.1 | $K$-clique clustering in dynamic networks | [54] |
| 4.2 | Clique-based fusion of graph streams | [122] |

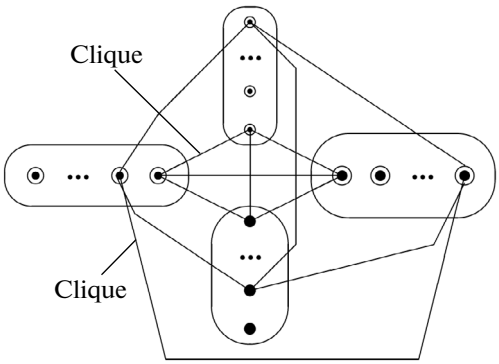



**Fig. 11.** Cliques in four-partite graph.

In recent decades, several new combinatorial problems as clique clustering in multipartite graphs have been suggested (e.g., [34, 48, 86, 118, 119, 123, 194]). Figure 11 illustrates this kind of problems. Table 11 contains a list of the research directions in the above-mentioned field.

### *4.3. Correlation Clustering*

Correlation clustering provides a method for partitioning a fully connected labeled graph (label "+" corresponds to edge for similar vertices, label "–" corresponds to edge for different vertices) while taking into account two objectives for the obtained clusters:

(i) minimizing disagreements (i.e., minimizing the number of "–" edges within the clusters ($Q^{\text{disagr}}(\hat{X}) \to \min$) or maximizing the number of "–" between clusters),

(ii) maximizing agreements (i.e., the number of "+" edges insides the clusters) ($Q^{\text{agr}}(\hat{X}) \to \max$) (e.g., [2, 15, 18, 20, 51, 111, 183]).

In the basic above-mentioned problem formulation, the objective functions are summarized. In other words, binary scale [–1, +1] is used for each edge as a weight (zero value is not used). Here it is not necessary to specify the preliminary number of clusters (e.g., as in $k$-means clustering). The correlation clustering problem formulation is motivated from documents/web pages clustering. This combinatorial model belongs to NP-complete class (e.g., [8, 17, 18]). Various versions of correlation problem formulations are examined: (a) weighted versions of the "correlation clustering functional" are considered as well (e.g., [32, 33, 51]), (b) correlation clustering with partial information (e.g., [50]), (c) correlation clustering with noisy input (e.g., [130]), etc.

Let us consider the weighted version of the problem. Let $A = \{A_1, ..., A_j, ..., A_n\}$ be the initial set of elements. As a result, $(n-1)^2$ elements pairs can be considered: $G = \{g_1, \ldots, g_{(n-1)^2}\}$. Each element of $G$ corresponds to element pair $(A_{j_1}, A_{j_2})$ and an element of proximity matrix $Z = \|z_{j_1,j_2}\|$. Further, it is possible to replace scale [–1, +1] for each edge (i.e., for each element from $G$ or element of proximity matrix $Z$) by two quantitative scales for weights: negative quantitative scale [$-w-$, ..., 0) instead of "–1" and positive quantitative scale (0, ..., $w^+$] instead of "+1". Evidently, element pair set is divided into two separated subsets $G = G^- \cup G^+$ (without intersection, i.e., $|G^- \cap G^+| = 0$) where ®$g^- \in G^-$ weight estimate corresponds to negative quantitative scale above, where $\forall g^+ \in G^+$ weight estimate corresponds to negative quantitative scale above. The clustering solution is: $\hat{X} = \{X_1, ..., X_\iota, ..., X_\lambda\}$. For this solution, two total quality parameters above are examined:

(i) total agreements quality as (summarization by all intra-cluster pairs with positive edge weight) $Q^{\text{agr}}(\hat{X})$ (maximization);

**Table 11.** Research directions in multi-partite graphs

| No. | Research | Source(s) |
|---|---|---|
| 1 | Problem of compatible representatives | [108] |
| 2 | Morphological clique (ordinal estimates) | [118, 119, 123] |
| 3 | Morphological clique (multiset estimates) | [121, 123] |
| 4 | Clustering in multipartite graph | [34, 194] |
| 5 | Bipartite and multipartite clique problems | [48] |
| 6 | Morphological clique over graph streams | [122] |
| 7 | Coreset problems | [60, 86] |
| 8 | Coresets in dynamic data streams | [64] |
| 9 | Detection of communities in $k$-partite networks | [127] |

**Table 12.** Complexity estimates of stages for greedy agglomerative heuristic

| Stage | Description | Complexity estimate (running time) |
|---|---|---|
| Stage 1 | Calculation of distance matrix $Z$ | $O(n^2)$ |
| Stage 2 | Calculation of positive/negative weights | $O(n^2)$ |
| Stage 3 | Specifying the initial solution | $O(1)$ |
| Stage 4 | Searching for the best element pair (by Pareto-efficient improvement of objective function) | $O(n^2)$ |
| Stage 5 | Analysis of algorithm end, recalculation of objective function | $O(n)$ |
| Stage 6 | Transition of computing process | $O(1)$ |
| Stage 7 | Stopping | $O(1)$ |

(ii) total disagreements quality (summarization by all intra-cluster pairs with negative edge weight) $Q^{\text{disagr}}(\hat{X})$ (for minimization, by module).

As a result, the weighted version of correlation clustering problem is (Fig. 12).

Find clustering solution $\hat{X}$ such that: (i) $Q^{\text{arg}}(\hat{X}) \to$ max and (ii) $\left|Q^{\text{disagr}}(\hat{X})\right| \to$ min.

Heuristics and approximation algorithms (e.g., PTAS) have been proposed for the problem versions (e.g., [15, 17, 18]). Clearly, agglomerative (hierarchical) clustering scheme (i.e., selection of an element pair from set $B$ for next joining for improvement of a current clustering solution) can be used here as a simple greedy algorithm (Bottom-Up process of selection of element pair with the best improvement of objective vector function and corresponding joining the elements) (Fig. 12).

*Stage 1.* Calculation of the matrix of element pair $\forall(A(j_1), A(j_2)), A(j_1) \in A, A(j_2) \in A, j_1 \neq j_2$ proximities ("distances").

*Stage 2.* Transformation of element pair proximities into positive (for similar elements) or negative (for dissimilar elements) weights (e.g., mapping).

*Stage 3.* Specifying the initial clustering solution $\hat{X}^0$ as composition of initial elements, vector objective function $\bar{f}^0 = (Q^{\text{disagr}}(\hat{X}^0), Q^{\text{agr}}(\hat{X}^0)) = (0,0)$ (initial value, initial index $\gamma = 0$).

*Stage 4.* Searching for the element pair with the best improvement of vector objective function $\bar{f}$ (i.e., searching for Pareto-efficient point(s)). Integration of the corresponding both elements into a cluster or inclusion of the corresponding element into the cluster with the second element (i.e., new clustering solution) $\hat{X}^q$ ($q$ is parameter of algorithm iteration). Recalculation of the current value of vector objective function: $\bar{f}^\gamma = (Q^{\text{disagr}}(\hat{X}^\gamma), Q^{\text{agr}}(\hat{X}^\gamma))$.

*Stage 5.* If all elements are processed then GO TO Stage 7.

*Stage 6.* Increasing index $\gamma = \gamma + 1$, while constraint $\left|Q^{\text{disagr}}(\hat{X})\right| \le q$ is satisfied Go To Stage 4, else GO TO Stage 7.

*Stage 7.* Stop.

Complexity estimates of greedy heuristic above for two-objectives correlation clustering (by stages) are presented in Table 12.

Table 13 contains a list of main research directions in correlation clustering.

On the other hand, it is possible to use a multiset based problem formulation. It is possible to replace scale [−1, +1] (or two quantitative scales above) for each edge (i.e., for each element from $G$ or element of proximity matrix $Z$) by two ordinal scales: negative ordinal scale $[-k^-, ..., -1]$ instead of "−1" and positive ordinal scale $[+1, ..., k^+]$ instead of "+1". Note, calculation of edge weights upon the above-mentioned scales is sufficiently easy (e.g., mapping of the quantitative estimate into the ordinal scale). For the clustering solution $\hat{X} = \{X_1, ..., X_\lambda\}$ two total quality parameters can be calculated as follows: (i) total agreements quality as multiset estimate (summarization by the component for all intra-cluster pairs with positive edge

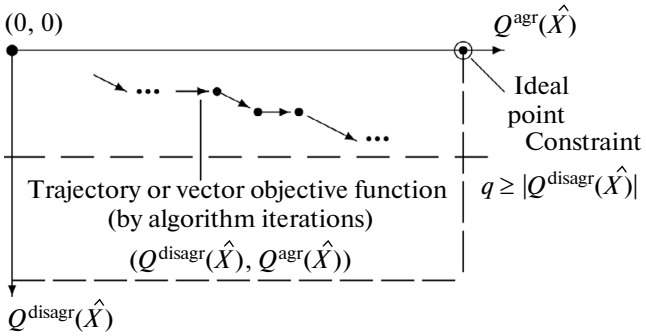



**Fig. 12.** "Space" of solution quality.

**Table 13.** Correlation clustering

| No. | Research direction | Source(s) |
|---|---|---|
| 1 | Basic problem formulations and complexity | [8, 15, 17, 18, 20, 111] |
| 2 | Surveys | [8, 18, 111] |
| 3 | Comparing methods for correlation clustering | [56] |
| 4 | Approximation algorithms (including PTAS) | [15, 17, 18, 68] |
| 5 | Weighted versions of correlation clustering problems | [32, 33, 51] |
| 6 | Correlation clustering with fixed number of clusters | [68] |
| 7 | Maximizing agreements via semidefinite programming | [183] |
| 8 | Minimizing disagreements on arbitrary weighted graphs | [57] |
| 9 | Global correlation clustering | [3] |
| 10 | Correlation clustering with partial information | [50] |
| 11 | Correlation clustering with noisy input | [130] |
| 12 | Error bounds for correlation clustering | [97] |
| 13 | Robust correlation clustering | [2, 110] |
| 14 | Correlation clustering in image segmentation | [106] |

weight) $Q^{\text{agr}}(\hat{X})$ (maximization); (ii) total disagreements quality as multiset estimate (summarization by the component for all intra-cluster pairs with negative edge weight) $Q^{\text{disagr}}(\hat{X})$ (for minimization). As a result, the multiset based correlation clustering problem is (Fig. 13).

Find clustering solution $\hat{X}$ such that $Q^{\text{agr}}(\hat{X}) \to \max$ and $\left|Q^{\text{disagr}}(\hat{X})\right| \to \min$.

### *4.4. Network Communities Based Clustering*

In recent decades, "network communities based clustering" as a new research direction has been organized (e.g., [63, 69, 87, 117, 142, 143, 144, 145, 146, 163]). The largest connected components are examined as "network communities", for example: cliques, quasi-cliques, cliques/quasi-cliques with leaves, chains of cliques/quasi-cliques, integrated groups of clisues/quasi-cliques (Fig. 14). The network example in Fig. 14 does not contains overlaps (i.e., without intersection of community structures). Figure 15 illustrates the overlaps.

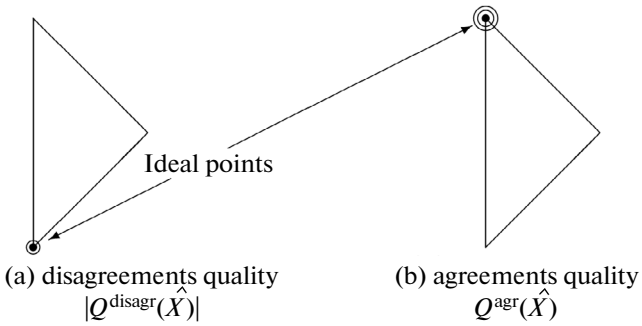



**Fig. 13.** Multiset quality posets (lattices).

The detection of "network communities structures" corresponds to complex combinatorial optimization models (e.g., linear/nonlinear integer programming, mixed integer programming). The models belong to NP-hard problems (e.g., [27, 39, 63, 145]). Table 14 contains a list of basic research directions in community network based clustering.

Modularity of a graph can be defined as a normalized tradeoff between edges covered by clusters and squared cluster degree sums [27, 146]. The problem is formulated as combinatorial optimization model. For the modularity maximization, several main algorithms are pointed out [27]: (a) greedy agglomeration [39, 142], (b) spectral division [144, 199], (c) simulated annealing [79, 164], (d) extremal optimization [55]. An example of modularity algorithm as greedy agglomerative heuristic is the following [142]:

*Stage 1.* Trivial clustering: each node corresponds to its own cluster,

*Stage 2.* Cycle by cluster pairs:

*Stage 2.1.* Calculation of possible increase of modularity for merging each cluster pairs.

*Stage 2.2.* Merging the two clusters with maximum possible increase.

*Stage 2.3.* If increasing of modularity by merges of cluster pair is impossible then GO TO Stage 3.

*Stage 2.4.* Go To Stage 2.2.

*Stage 3.* Stop.

In this algorithm, algorithmic complexity estimate equals $O((p + n)n)$ or $O(n^2)$ [142].

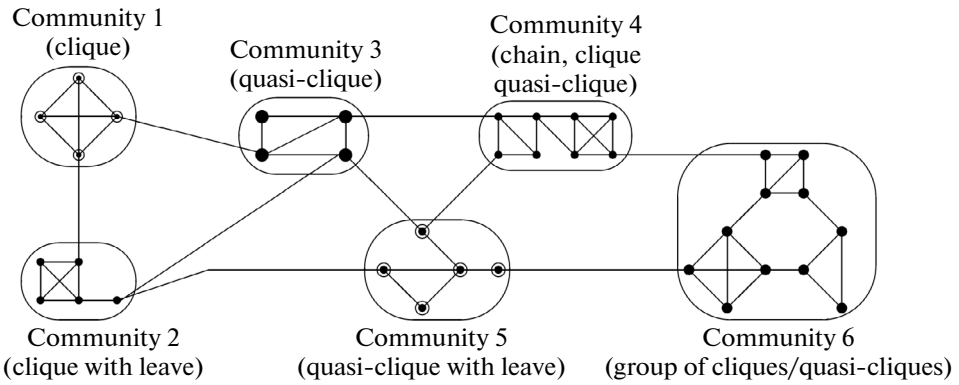



**Fig. 14.** Illustration for network communities.

The general scheme of Girvan-Newman (GN) algorithm based on edge betweenness is [69]:

*Step 1.* Calculation of the betweenness score for each the edges.

*Step 2.* Deletion of the edges with the highest score.

*Step 3.* Performance analysis for the network's components.

*Step 4.* If all edges are deleted and the system breaks up into N non-connected nodes Go TO Step 5. Otherwise GO TO Step 1.

*Step 5.* Stop.

Algorithmic complexity estimate of the algorithm equals $O(p^2n)$ ($p$ is the number of edges) [69]).

## 5. TOWARDS FAST CLUSTERING

Many applications based on very large data sets/networks require fast clustering approaches (e.g., [39, 142, 179, 188, 196]). In Table 15, basic ideas for fast clustering schemes are pointed out. Generally, many fast clustering schemes consist of two basic levels (global level and local level): (a) partition of the initial problems into local problems (i.e., decreased dimension, limited type of objects/elements) (global level), (b) clustering of local clustering problems (local level), (c) composition/integration of local clustering solutions into a resultant global clustering solution (global level).

In Table 16, a list of basic fast local clustering algorithms (i.e., fast sub-algorithms) is presented.

## 6. APPLIED EXAMPLES

### *6.1. Analysis of Network*

Network analysis can be based on dividing the network nodes into classes by their structural properties. Figure 19 illustrates network node types.

As a result, a multi-type object clustering strategy can be applied. The types of networks nodes can be obtained by analysis of their connections (the number and types of neighbors), for example: (a) multi-connected nodes (type 1), (b) connected nodes (type 2), (c) outliers (type 3), (d) isolated nodes (type 4). Let $G = (A, E)$ be the examined network (graph), where $A = \{A_1, ..., A_i, ..., A_n\}$ is the set of nodes, $E$ is the set of edges ($|E| = h$). The following clustering scheme can be considered:

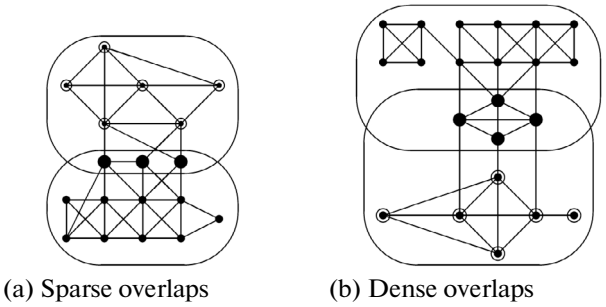



**Fig. 15.** Illustration for overlaps in structures.

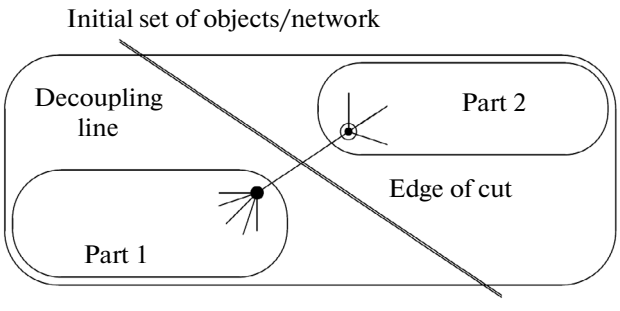



**Fig. 16.** Edge betweenness for decoupling.

**Table 14.** Community network based clustering

| No. | Research direction | Source(s) |
|---|---|---|
| 1 | Basic issues: | |
| 1.1 | Basic problem formulation | [27, 63, 69, 117, 146, 144, 145, 163, 198, 207, 208] |
| 1.2 | Basic surveys | [27, 24, 63, 117, 135, 146, 144, 145, 163] |
| 1.3 | Problems complexity | [27, 39, 63, 145] |
| 1.4 | Overlapping (fuzzy) community structures | [73, 198, 200, 205, 206, 207] |
| 1.5 | Analysis/evaluation of community structures | [117, 146, 198, 208] |
| 2 | Main algorithms/solving schemes: | |
| 2.1 | Algorithm based on edge betweeness (divisive algorithm) | [69] |
| 2.2 | Modularity algorithm as greedy agglomerative heuristic | [142, 152] |
| 2.3 | “Karate Club” algorithm | [146] |
| 2.4 | Kernighan-Lin method and variants | [102] |
| 2.5 | Overlapping communities (clique percolation, local expansion, dynamic algorithms, etc.) | [73, 200, 205, 206, 207] |
| 2.6 | Spectral clustering algorithms, modifications | [208] |
| 2.7 | Genetic algorithms | [126] |
| 2.8 | Agent-based algorithms | [80] |
| 3 | Modularity clustering (maximum modularity): | |
| 3.1 | Surveys | [27, 144, 213, 208] |
| 3.2 | Tripartite modularity (three vertex types) | [139, 140] |
| 3.3 | Modularity in $k$-partite networks | [127] |
| 3.4 | Greedy agglomeration algorithm | [39, 142] |
| 3.5 | Spectral division algorithm | [144, 199] |
| 3.6 | Simulated annealing algorithms | [79, 164] |
| 3.7 | Detecting communities by merging cliques | [203] |
| 3.8 | Extremal optimization scheme (mathematical programming) | [4, 55] |
| 3.9 | Global optimization approach | [132] |
| 3.10 | Memetic algorithm | [141] |
| 3.11 | Random works algorithms | [162] |
| 3.12 | Multi-level algorithms | [53, 147] |
| 4 | Large networks: | |
| 4.1 | Communities in large networks | [22, 39, 73, 87, 88, 117, 162, 208] |
| 4.2. | Communities in mega-scale networks | [196] |
| 4.3 | Communities in super-scale networks | [22] |
| 4.4 | Tracking evolving communities in large networks | [88] |
| 5 | Applications: | |
| 5.1 | World Wide Web | [53, 117, 139] |
| 5.2 | Journal/article networks, citation networks, etc. | [37, 63, 167] |
| 5.3 | Social networks (friendship, collaboration, etc.) | [63, 69, 78, 142, 145, 146, 196, 208] |
| 5.4 | Biological networks | [63, 69, 146] |
| 5.5 | Purchasing network | [39] |
| 5.6 | CAD applications | [146] |
| 5.7 | Antenna-To-Antenna network (mobile phone network) | [23, 128] |

**Table 15.** Main approaches to fast clustering

| No. | Approach | Solving schemes | Source(s) |
|---|---|---|---|
| 1 | Aggregation of object/network nodes | Hierarchical clustering (Bottom-Up, step-by-step node aggregation) | [95, 96, 179] |
| 2 | Division of objects/network nodes (partition/decomposition): | Top-Down scheme | |
| 2.1 | Pruning of objects/network nodes (Fig. 16) | 1. Selection of basic edge betweenness in graph and decoupling (Top-Down scheme)<br>2. Clustering in each graph part (if needed) | [69, 179] |
| 2.2 | Multi-level schemes (partition, clustering, integration of solutions): | 1. Partitionof object set/network<br>2. Clustering of local regions<br>3. Composition of local solutions | [187, 188] |
| 2.2.1 | "Basic" objects (special "key" objects/nodes) based clustering (Fig. 17) | 1.Detection of "basic" objects/nodes (e.g., by filtering)<br>2. Clustering of "basic" objects/nodes<br>3. Joining other elements/nodes to obtained clusters | |
| 2.2.2 | Grid-based clustering | Dividing the space into cells | [111, 116, 125] |
| 2.2.3 | Grid-based clustering in data streams | Online clustering of data streams | [129, 157] |
| 2.2.4 | Grid-based clustering (composition): multiple division of objects "space"/network into cells/regions (e.g., axis-parallel subspaces), region-based clustering, composition of local solutions (Fig. 18) | 1. Gridover object "space"/network<br>2. Analysisof grid regions<br>3. Selection of "non-empty' regions<br>4. Clustering in "dense" regions<br>5. Clustering in "sparse" regions (while taking into account solutions in "dense" regions)<br>6. Composition of regions solutions | |
| 2.2.5 | Grid-based clustering (extension): multiple division of objects "space"/network and "extension" of clustering solutions (with condensing of clusters, as in dynamic programming) (Fig. 18) | 1.Gridover object "space"/network<br>2. Analysisof grid regions<br>3. Selection of "non-empty" regions<br>4. Clusteringin "dense" regions<br>5. Extension of "dense" regions by neighbor region(s) and extension of clustering solution(s) | |
| 2.2.6 | Division of object "space"/network by types ($k$-partite network) (close to 2.2.1) | 1. Detection of objects by types<br>2. Clustering for each part<br>3. Composition of clustering solutions | [127, 139, 140] |
| 3 | Composite (multistage, concurrent, multi-techniques) approaches | Composition/combination of various approaches | [115] |

*Stage 1.* Building the list of nodes (with info on neighbors) $O(n)$.

*Stage 2.* Selection of multi-neighbor nodes (type 1) (complexity estimate equals $O(n)$). Result: subset $B_1 \subset A$.

*Stage 3.* Selection of outlier nodes (i.e., leaves, type 3) (complexity estimate equals $O(n)$). Result: subset $B_3 \subset \{A \backslash B_1\}$.

*Stage 4.* Selection of other nodes (type 2) (complexity estimate equals $O(n)$). Result: subset $B_2 \subset A$, $B_2 = \{A \backslash \{B_1 \cup B_3\}$.

*Stage 5.* Clustering of multi-neighbor nodes $B_1$ (complexity estimate equals about $O(|B_1|^2)$ (about $O((n/3)^2)$). Thus, a preliminary clustering solution is: $\hat{X}^1 = \{X_1^1, \ldots, X_l^1, \ldots, X_{q1}^1\}$. Now, a macro-network can be built: $G^1 = (\hat{X}^1, E^1)$, where $\hat{X}^1$ is the node set that corresponds to the obtained clustering solution (i.e., set of clusters), $E^1$ is a built set of edges. (Note, the obtained clusters can be used as centroids in $k$-means clustering at the next step(s)).

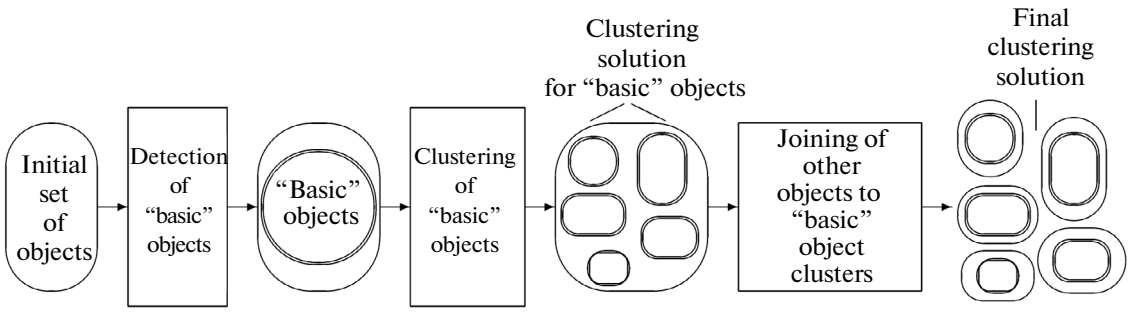



**Fig. 17.** "Basic" objects based clustering framework "Grid" Object "space"/network.

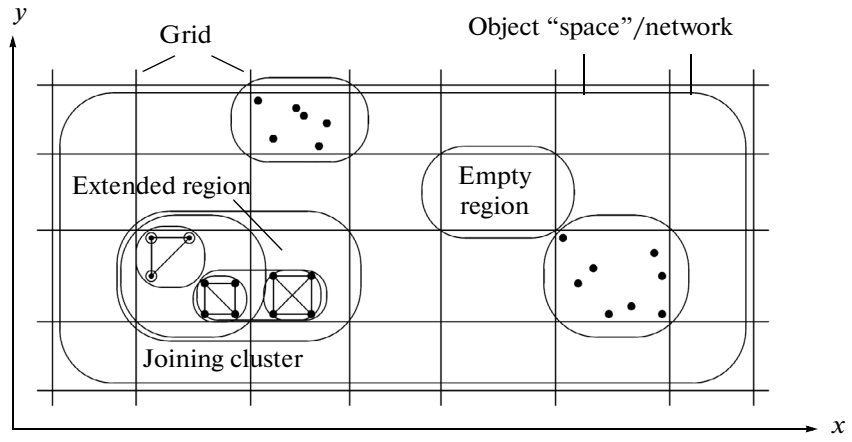



**Fig. 18.** "Grid" over object "space"/network.

*Stage 6.* Clustering of nodes of type 2, i.e., set $B_2$ (if needed). The corresponding clustering solution is: $\hat{X}^2 = \{X_1^2, \ldots, X_l^1, \ldots X_{q2}^2\}$. Now, a macro-network can be built: $G^2 = (\hat{X}^2, E^2)$, where $\hat{X}^2$ is the node set that corresponds to the obtained clustering solution (i.e., set of clusters), $E^2$ is a built set of edges.

*Stage 7.* Matching of two graphs $G^1 = (\hat{X}^1, E^1)$ and $G^2 = (\hat{X}^2, E^2)$. The matching process can be based on edges from initial network or on the usage of additional parameters (e.g., node coordinates). As a result, integrated clustering solution can be obtained $\hat{X}^{12} = \{X_1^{12}, \ldots, X_l^{12}, \ldots, X_{q12}^{12}\}$.

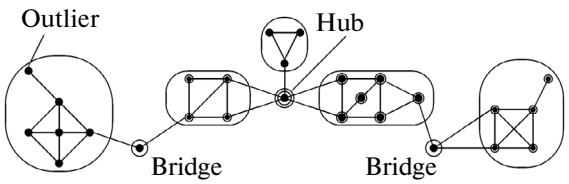



**Fig. 19.** Illustration for network nodes.

*Stage 8.* Joining outliers ($B_3$) to clusters of solution $\hat{X}^{12}$. As a result, integrated clustering solution can be obtained $\hat{X}^{123}$.

### *6.2. Scheduling in Multi-Beam Antenna Communication System*

There are the following initial information (Fig. 20): (a) multi-beam antenna system (and its positioning), (b) number of antenna beams: (c) set of communication nodes $A = \{A_1, \ldots, A_i, \ldots, A_n\}$ (including node positions, required communication resources, etc.), (d) volume of transmitted data is about the same for each $A_i$ (for simplification). The problem is:

Design a schedule for connection of antenna system to communication nodes while taking into account the following: (i) minimization of total con-

**Table 16.** List of some fast local clustering algorithms

| No. | Fast scheme | Description | Complexity estimate (running time) | Source(s) |
|---|---|---|---|---|
| 1 | Basic agglomerative(hierarchical) algorithm | Bottom-up joining the closest object pair | $O(n^3)$ | [96] |
| 2 | Balanced by cluster size hierarchical algorithm | Bottom-up joining the closest object pair under constraints for cluster size | $O(n^3)$ | |
| 3 | Minimum spanning tree based algorithm | Clustering the spanning tree nodes | $O(n\log n)$ | [74, 75, 138] [154, 160, 181] [197, 201] |
| 4 | Balanced by cluster size minimum spanning tree based algorithm | Clustering the spanning tree nodes under constraints for cluster size | $O(n\log n)$ | |
| 5 | Graph clustering algorithm | Detection of network communities (edge betweeness of the graph) | $O(p^2n)$ | [69] |
| 6 | Modularity graph clustering algorithm | Modularity based detection of network communities | $O((p+n)n)$ or $O(n^2)$ | [142] |
| 7 | Algorihtms based on grid over "space of object coordinates" (partition space techniques) | Assignment of objects into local regions of "space of object coordinates" | $O(n + n' \times n'')$ $(n' \ll n, n'' \ll n)$ | [188] |
| 8 | Clustering based on cores decomposition of networks | Preliminary cores decomposition of covering graph | $O(n^2) + O(h)$ | [19] |

nection time, (ii) providing the best communication quality (by the minimum interference between neighbor (by angle) connections, i.e.,

$$\max_{i \in A} \min_{i_1, i_2 \in A} D^{\text{angular}}(A_{i_1}, A_{i_2}),$$

where $D^{\text{angular}}(A_{i_1}, A_{i_2})$ is angular separation (or angle between beams to the nodes).

The pointed out proximity measure is defined as follows (e.g., [159]). Let $x = (x_1, ..., x_\iota, ..., x_m)$ and $y = (y_1, ..., y_\iota, ..., y_m)$ be coordinate/parameter vectors for two items (i.e., $x$ and $y$). The angular separation proximity is: $D^{\text{angular}}(x, y) = \dfrac{\sum_{i=\iota}^{m} x_\iota y_\iota}{[\sum_{i=\iota}^{m} x_\iota^2 \sum_{\iota=1}^{m} y_\iota^2]^{1/2}}$. The similarity measure corresponds to the angle between the item vectors in directions of $x$ and $y$. The heuristic solving scheme is:

*Stage 1.* Linear ordering of communication nodes by their angle (Fig. 20, node 1 is the 1st).

*Stage 2.* Dividing of the obtain list of nodes into μ equal by size groups (the last group can have less elements) and numeration as follows:

*group 1*: {(1, 1), (1, 2),..., (1, $k$)},

*group 2*: {(2, 1), (2, 2),..., (2, $k$)},

...

group μ: {(μ, 1), (μ, 2), ..., (μ, $k$)}.

Here $k = \left\lceil \dfrac{n}{\mu} \right\rceil$.

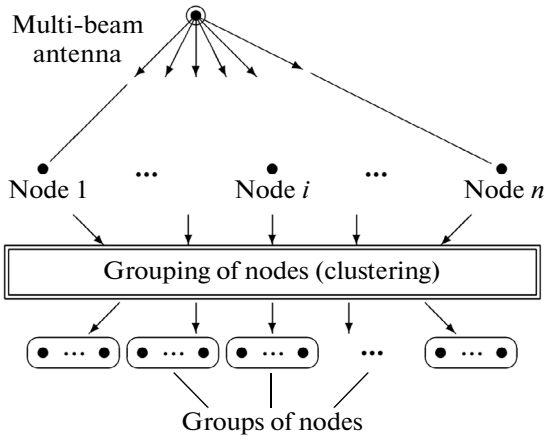



**Fig. 20.** Multi-beam antenna system.

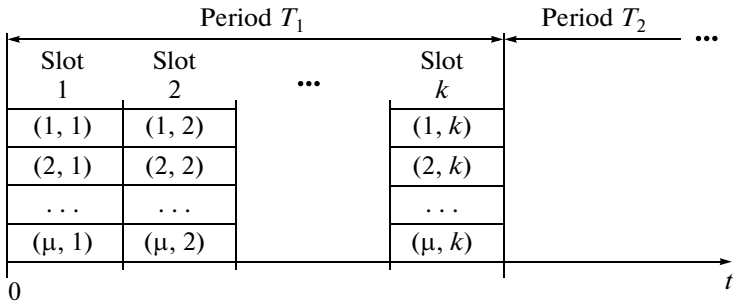



**Fig. 21.** Scheme of schedule.

*Stage 3.* Generation of scheduling by the rules: Slot $j$ ($j = \overline{1, k}$): the $j$th element from each group ($\zeta = 1, 2, ..., \mu$), i.e., elements $\{\zeta, j\}$ (Fig. 21).

*Stage 4.* Stop.

## 7. CONCLUSIONS

In the article, an "engineering viewpoint" to combinatorial clustering approaches is described. First, a literature review presented: (a) on basic combinatorial methods/models for clustering, (b) on clustering in large scale data sets/networks. A special attention is targeted to quality characteristics of clustering solutions and multicriteria clustering. Many contemporary clustering methods consist in graph based clustering. Here, illustrative solving schemes and numerical examples are described: (i) minimum spanning tree based clustering methods, (ii) clique based clustering, (iii) correlation clustering, (iv) network communities based clustering. In addition, a description of fast clustering methods contains main ideas and an analysis of basic literature sources. A couple of applied examples are described: (1) analysis of networks and (2) scheduling of multi-beam antenna systems. It is reasonable to point out some important future research directions: *1.* design of new composite combinatorial solving frameworks; *2.* design of special decision support tools (modular solving environments) for structural clustering problems; *3.* study of dynamic clustering problems; and *4.* applications of combinatorial clustering problems in networking (design, covering, routing, etc.).

## 8. ACKNOWLEDGMENTS

This research was done in Institute for Information Transmission Problems (IITP RAS) and supported by Russian Science Foundation grant 14-50-00150 "Digital technologies and their applications". The author thanks Prof. Andrey I. Lyakhov for preliminary engineering description of problem: design of communication schedule for multiple beam antenna system.

## REFERENCES

1. J. Abello, M. G. C. Resende, and S. Sudarsky, "Massive quasi-clique detection," in *Lecture Notes in Computer Science (LNCS),* Vol. 2573: *Proc. 5th Latin American Symp. on Theoretical Informatics (LATIN 2002), Cancun, Mexico, Apr. 3–6, 2002*, Ed. By Rajsbaum (Springer-Verlag, Berlin, 2002), pp. 598–612.
2. E. Achtert, C. Bohm, H.-P. Kriegel, P. Kroger, and A. Zimek, "Robust, complete, and efficient correlation clustering," in *Proc. 7th SIAM Int. Conf. on Data Mining (SDM), Minneapolis, MN, 2007* (SIAM, 2007), pp. 413–418.
3. E. Achtert, C. Bohm, J. David, P. Kroger, and A. Zimek, "Global correlation clustering based on the hough transform," Stat. Anal. Data Mining, **1**, 111–127 (2008).
4. G. Agarwal and D. Kempe, "Modularity maximizing network communitites using mathematical programming," Eur. Phys J. **66**, 4009–418 (2008).
5. *Data Streams: Models and Algorithms*, Ed. by C. C. Aggarwal (Springer, New York, 2007).
6. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data," Data Mining Knowl. Discov. **11** (5), 5–33, (2005).
7. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison Wesley, Reading, MA, 1974).
8. N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information: Ranking and clustering," J. ACM **55** (5), art. No. 23, (2008).
9. E. Akkoyunlu, "The enumeration of maximal cliques of large graph," SIAM J. Comput. **2** (1), 1–6 (1973).
10. N. Alon, M. Krivelevich, and B. Sudakov, "Finding a large hidden clique in a random graph," in *Proc. 9th, Ann. ACM-SIAM Symp. on Discr. Alg., San Francisco, CA, 1998* (ACM, 1998), pp. 594–598.
11. C. J. Augeri and H. H. Ali, "New graph-based algorithms for partitioning VLSI circuits," in *Proc. IEEE Int. Symp. on Cirquits and Systems (ISCAS'04), Vancouver, Canada, May 23–26, 2004* (IEEE, New York, 2004), Vol. 4, pp. 521–524.
12. H. Ayad and M. S. Kamel, "On voting-based consensus of cluster ensembles," Pattern Recogn. **43**, 1943–1953 (2010).
13. L. Babel, "A fast algorithm for the maximum weight clique problem," Computing **52**, 31–38 (1994).

14. G. Babu and M. Nurty, "Clustering with evolution strategy," Pattern Recogn. Lett. **14**, 763–769, (1993).
15. S. Bagon and M. Galun, "Optimizing large scale correlation clustering," Electr. Prepr., 9 p., Dec. 13, (2011); http://arxiv.org/abs/1112.2903 [cs.CV]
16. E. Balas, V. Chvatal, and J. Nesetril, "On the maximum weight clique problem," Math., Oper. Res. **12**, 522–535 (1987).
17. N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," in *Proc. 43rd Symp. on Foundations of Computer Science (FOCS 2002), Vancouver, BC, Canada, Nov. 16–19, 2002* (IEEE, New York, 2002), pp. 238–250.
18. N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," Mach. Learn. **56**, 89–113 (2004).
19. V. Batagelj and M. Zavershik, "An O(m) algorithm for cores decomposition of networks," Electr. Prepr., 10 p., Oct. 25, (2003); http://arxiv.org/abs/0310.0049 [cs.DS]
20. A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," J. Comput. Biology **6**, 281–292, (1999).
21. P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, (Springer-Verlag, New York, 2006), 25–71.
22. V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," Electr. Prepr., 12 p., July 25, 2008; http://arxiv.org/abs/0803.0476 [physics.soc-ph]
23. V. D. Blondel, M. Esch, C. Chan, F. Clerot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki, "Data for development the d4d challenge on mobile phone data," Electr. Prepr., 10 p., Jan. 28, (2012); http://arxiv.org/abs/1210.0137 [cs.CY]
24. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex netwoprks: Structure and dynamics," Phys. Rep. **424**, 175–208 (2006).
25. D. Boley, M. Gini, R. Gross, S. Han, K. Hastings, G. Kapyris, V. Kumar, B. Mobasher, and J. Moor, "Partitioning-based clustering of web document categorization," Decision Support Syst. (DSS) **27**, 329–341 (1999).
26. I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The maximum clique problem," in *Handbook of Combinatorial Optimization*, Ed. by D.-Z. Du and P. M. Pardalos (Springer, New York, 1999), (Suppl. vol. A), pp. 659–729.
27. U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikolosk, and D. Wagner, "On modularity clustering," IEEE Trans. Knowl. Data Eng. **20**, 172–188, (2008).
28. C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph," Commun. ACM **16**, 575–577, (1973).
29. D. Brown and C. Huntley, "A practical application of simulated annealing to clustering," Pattern Recogn. **25**, 401–412, (1992).
30. S. Butenko and W. Wilhelm, "Clique-detection models in computational biochemistry and genomics," Eur. J. Operat. Res. (EJOR) **173** (1), 1–17, (2006).
31. Z. Cai, M. Lu, and X. Wang, "Channel access-based self-organized clustering in ad hoc networks," IEEE Trans. Mobile Comput. **2**, 102–113 (2003).
32. M. Charikar, V. Guruswami, and A. Wirth, "Clustering with quantitative information," in *Proc. 44th Symp. on Foundations of Computer Science (FOCS 2003), Cambridge, MA, USA, Oct. 11–14, 2003*, (IEEE, New York, 2003), pp. 524–533.
33. M. Charikar, V. Guruswami, and A. Wirth, "Clustering with quantitative information," J. Comput. Syst. Sci. **71**, 360–383 (2005).
34. I. Charon and O. Hundry, "Optimal clustering in multipartite graph," Disc. Appl. Math. **156**, 1330–1347 (2008).
35. C.-Y. Chen and F. Ye, "Particle swam optimization algorithm and its application to cluster analysis," in *Proc. 2004 IEEE Int. Conf. on Networking, Sensing and Control, Taipei, Mar. 21–23, 2004* (IEEE, New York, 2004), vol. 2, 789–794.
36. Y. P. Chen and A. L. Liestman, "Maintaining weakly-connected dominating sets for clustering ad hoc networks," Ad Hoc Netw. **3**, 629–642 (2005).
37. P. Chen and S. Redner, "Community structure of the physical review citation network," J. Informetrics **4**, 278–290 (2010).
38. C. H. Cheng, "A branch-and-bound clustering algorithm," IEEE Trans. Syst. Man Cybern. **25**, 895–898 (1995).
39. A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," Phys. Review E **70**, No. 066111, 2004.
40. J. Coble, D. J. Cook, and L. B. Holder, "Structure discovery in sequentially-connected data streams," Int. J. Artif. Intell. Tools **15**, 917–944 (2006).
41. C. Cobos, M. Mendoza, and Leon E., "A hyper-heuristic approach to design and tuning heuristic methods for web document clustering," in *Proc. 2011 IEEE Cong. on Evolutionary Computation (CEC), New Orleans, USA, June 5–8, 2011* (IEEE, New York, 2011), pp. 1350–1358.
42. D. Cokuslu, K. Erciyes, and O. Dagdeviren, "A dominating set based clustering algorithm for mobile ad hoc networks," in *Lecture Notes in Computer Science (LNCS)*, Vol. 3991: *Proc. 6th Int. Conf. on Computational Science (ICCS'2006), Reading, UK, May 28–31, 2006*, Ed. by V. N. Alexandrov, et al. (Springer-Verlag, Berlin, 2006), pp. 571–578.
43. D. Cokuslu and K. Erciyes, "A hierarchical connected dominating set based clustering algorithm for mobile ad hoc networks," in *Proc. 15th Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'07), Istanbul, Turkey, Oct. 2007* (IEEE, New York, 2007), pp. 60–66.
44. A. Condon and R. M. Karp, "Algorithms for graph partitining on the planted partition model," Random Struct. Alg. **18**, 116–140 (2001).
45. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 3rd ed. (MIT Press, Cambridge, 2009).
46. D. G. Corneil and Y. Perl, "Clustering and domination in perfect graphs," Disc. Appl. Math. **9**, 27–39, 1984.
47. M. C. Cowgill, R. J. Harvey, and L. T. Watson, "A genetic algorithm approach to cluster analysis," Comput. Math. Appl. **37** (7), 99–108, (1999).
48. M. Dawande, P. Keskinocak, J. M. Swaminathan and S. Tayur, "On bipartite and multipartite clique problems," J. Algorithms **41**, 388–403 (2001).

49. S. G. de Amorim, J.-P. Barthelemy, and C. C. Ribeiro, "Clustering and clique partitioning: Simulated anealing and tabu search approaches," J. Classif. **9** (1), 17–41 (1992).
50. E. D. Demaine and N. Immorlica, "Correlation clusteirng with partial information," in *Approximation, Randomization, and Combinatorial Optimization* (Algorithms and Techniques, Springer-Verlag, 2003), pp. 1–13.
51. E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica, "Correlation clustering in general weighted graphs," Theor. Comp. Sci. **361**, 172–187 (2006).
52. C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max algorithm for graph partitioning and data clustering," in *Proc. IEEE Int. Conf. on Data Mining (ICDM'01), San Jose, Nov. 2001* (IEEE, New York, 2001), pp. 107–111.
53. H. N. Djidjev, "A scalable multilevel algorithm for graph clustering and community structure detection," in *Lecture Notes in Computer Science (LNCS)*, Vol. 4936: *Proc. 4th Int. Workshop on Algorithms and Models for the Web-Graph (WAW'06), Banff, Canada, Nov. 30–Dec. 1, 2006*, Ed. by W. Aiello et al. (Springer, Berlin, 2008), pp. 117–128.
54. D. Duan, Y. Li, R. Li, and Z. Lu, "Incremantal K-clique clustering in dynamic social networks," Artif. Intell. Rev. **38**, 129–147 (2012).
55. J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," Phys. Rev. E **72**, 027104 (2005).
56. M. Elsner and W. Schudy, "Bounding and comparing methods for correlation clustering beyond ILP," in *Proc. NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing, Boulder, Co, May 31–June 5, 2009* (North Am. Chap. Ass. Comput. Linguistics, 2009), pp. 19–27.
57. D. Emanuel and A. Fiat, "Correlation clustering - minimizing disagreements on arbitrary weighted graphs," in *Proc. 11th Ann. Eur. Symp. on Algorithms, ESA-2003, Budapest, Hungary, Sept., 2003* (Springer-Verlag, Berlin, 2003), pp. 208–220.
58. G. Even, J. Naor, S. Rao, and B. Schieber, "Fast approximate graph partitioning algorithms," SIAM J. Comput. **28**, 2187–2214 (1999).
59. U. Feige and R. Krauthgamer, "Finding and certifying a large clique in a semi-random graph," Random Struc. Alg. **16**, 195–208, (2000).
60. D. Feldman and M. Langberg, "A unified framework for approximating and clustering data," in *Proc. 43rd ACM Symp. on Theory of Computing, (STOC 2011), San Jose, CA, USA, June 6–8, 2011* (ACM, 2011), pp. 569–578.
61. A. E. Feldman and L. Foschini, "Balanced partitions of trees and applications," Algorithmica **71**, 354–376 (2015).
62. M. R. Fellows, J. Guob, C. Komusiewicz, R. Niedermeier and J. Uhlmann, "Graph-based data clustering with overlaps," Disc. Optim. **8**, 2–17 (2011).
63. S. Fortunato, "Community detection in graphs," Electr. Prepr., 103 p., Jan. 25, (2010); http://arxiv.org/abs/0906.0612v2 [physics.soc-ph]
64. G. Frahling and C. Sohler, "Coresets in dynamic geometric data streams," in *Proc. 37th ACM Symp. on Theory of Computing (STOC 2005), Baltimore, MD, USA, May 22–24, 2005* (ACM, 2005), pp. 209–217.
65. E. M. Furems, "Dominance-based extension of STEPCLASS for multiattribute nominal classification," Int. J. Inform. Technol. Dec. Making **12**, 905–925 (2013).
66. H. N. Gabow, Z. Galil, T. Spencer, and R. E. Tarjan, "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs," Combinatorica **6**, 109–122 (1986).
67. M. R. Garey and D. S. Johnson, *Computers and Intractability. The Guide to the Theory of NP-Completeness* (W. H. Freeman, San Francisco, 1979).
68. I. Giotis and V. Guruswami, "Correlation clustering with a fixed number of clusters," in *Proc. 17th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'06), Miami, FL, Jan. 22–26, 2006* (SIAM, New York, 2006), pp. 1167–1176.
69. M. Girvan and M. E. J. Newman, "Community structure in social an biological networks. Community structure in social and biological networks," Proc. Natl. Acad. Sci. USA (PNAS) **99**, 8271–8276 (2002).
70. J. Gramm, J. Guo, F. Huffner, and R. Niedermeier, "Graph-modeled data clustering: Fixed-parameter algorithm for clique generation," Theory Comput. Syst. **38**, 373–392 (2005).
71. J. Goldberger and T. Tassa, "A hierarchical clustering algorithm based on the Hungarian method," Pattern Recogn. Lett. **29**, 1632–1638 (2008).
72. B. Goldengorin, D. Krushinsky, and P. M. Pardalos, *Cell Formation in Industrial Engineering: Theory, Algorithms and Experiments* (Springer-Verlag, New York, 2013).
73. P. K. Gopalan and D. M. Blei, "Efficient discovery of overlapping communites in massive networks," Proc. Natl. Acad. Sci. USA (PNAS) **110**, 14534–14539 (2013).
74. J. Gower and G. Ross, "Minimum spanning trees and single linkage cluster analysis," J. Royal Stat. Soc., Ser. C: Appl. Stat. **18**, 54–64 (1969).
75. O. Grygorash, Y. Zhou, and Z. Jorgensen, "Minimum spanning tree based clustering algorithms," in *Proc. 18th IEEE Int. Conf. in Tools with Artificial Intelligence (ICTAI'06), Arlington, VA, USA, Nov. 13–15, 2006* (IEEE, New York, 2006), pp. 73–81.
76. A. Guenoche, "Consensus partitions: a constructive approach," Adv. Data Anal., Classif. **5**, 215–229 (2011).
77. S. Guha, N. Mishra, R. Motwani, and L. O'Callagham, "Clustering data streams," in *Proc. 41st Ann. Symp. Foundations of Computer Science (FOCS), Redondo Beach, CA, Nov. 12–14, 2000* (IEEE Computer Society, 2000), pp. 359–366.
78. R. Guimera, L. Dadon, A. Diaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," Phys. Rev. E **68**, 065103 (2003).
79. R. Guimera, M. Sales-Pardo, and L. A. N. Amaral, "Modularity from fluctuations in random graphs and complex networks," Phys. Rev. E **70**, 025101 (2004).
80. I. Gunes and H. Bingol, "Coomunity detection in complex networks using agents," Electr. Prepr., 5 p., Oct. 23, (2006); arXiv:cs/0610129 [cs.MA]
81. B. Han and W. Jia, "Clustering wireless ad hoc networks with weakly connected dominating set," J. Paral. Distr. Comput. **67**, 727–737 (2007).

82. P. Hansen and N. Mladenovic, "Variable neighborhood search for the p-median," Location Sci. **5**, 207–226 (1997).
83. P. Hansen, J. Brimberg, D. Urosevic, and N. Mladenovic, "Data Clustering using Large p-Median Models and Primal-Dual Variable Neighborhood Search," Les Cahiers du GERAD, G-2007-41, June (2007).
84. P. Hansen, J. Brimberg, D. Urosevic, and N. Mladenovic, "Primal-dual variable neighborhood search for the simple plant-location problem," INFORMS J. Comput. **19**, 552–564 (2007).
85. P. Hansen, J. Brimberg, D. Urosevic, and N. Mladenovic, "Solving large p-median clustering problems by primal-dual variable neighborhood search," Data Min. Knowl. Discov. **19**, 351–375 (2009).
86. S. Har-Peled and S. Mazumdar, "On coresets for k-mean and k-median clustering," in *Proc. 36th Annual ACM Symp. on Theory of Computing, Chicago, IL, USA, June 13–16, 2004* (ACM, 2004), pp. 291–300.
87. J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Natural communities in large linked networks," in *Proc. 9th ACM SIGKDD Int. Conf. on Knowl. Discov. Data Mining (KDD'03), New York, NY, USA, 2003* (ACM, New York, 2003), pp. 541–546.
88. J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," Proc. Natl. Acad. Sci. USA **101** (Suppl. 1), 5249–5353 (2004).
89. T. C. Hou and T.-J. Tsai, "An access-based clustering protocol for multihop wireless ad hoc networks," IEEE J. Selec. Areas Commun. **19**, 1201–1210 (2001).
90. E. R. Hruschka, R. G. B. Campello, A. A. Freitas, and A. P. L. Carvalho, "A survey of evolutionary algorithms for clustering," IEEE Trans. Syst. Man Cybern., Part C **39** (2), 133–155 (2009).
91. Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," Data Mining Knowl. Discov. **2**, 283–304 (1998).
92. R. K. R. Indukuri and S. V. Penumathsa, "Dominating sets and spanning tree based clustering algorithms for mobile ad hoc networks," Int. J. Adv. Comput. Sci. Appl. **2**, 75–81 (2011).
93. A. S. Ivanov, A. I. Lyakhov, and E. M. Khorov, "Analytical model of batch flow multihop transmission in wireless networks with channel reservation," Autom. Remote Control, **76** (7), 1179–1192 (2015).
94. A. K. Jain, "Data clustering: 50 years beyond k-means," Pattern Recogn. Lett. **31**, 651–666 (2010).
95. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data* (Prentice Hall, Upper Saddle River, 1988).
96. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," ACM Comput. Surv. **31**, 264–323 (1999).
97. T. Joachims and J. Hopcroft, "Error bounds for correlation clustering," in *Proc. 22nd Int. Conf. on Machine Learning (ICML'05), Bonn, Germany, Aug. 7–11, 2005* (ACM, 2005), pp. 385–392.
98. *DIMACS Ser. in Disc. Math, and Theor. Comp. Sci.*, Vol. 26: *Cliques, Coloring, and Satisfiability*, Ed. by D. S. Johnson and M. A. Trick (AMS, Providence, 1996).
99. R. Jovanovic, M. Tuba, and S. Voss, "An ant colony optimization algorithm for partitioning graphs with supply and demand," Electr. prep. 21 p., March 3 (2015); http://arxiv.org/abs/1503.00899 [cs.AI]
100. I. Kargin, E. Khorov, and A. Lyakhov, "A mathematical method to estimate packet loss ratio for a multipath route with error correlation" Probl. Inform. Transmission, (2015) (in press).
101. R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, Ed. by R. E. Miller and J. W. Thatcher (Plenum, New York, 1972), pp. 85–103.
102. B. Kernigham and S. Lin, "An efficient heuristic procedure for partitioning graphs," Bell Syst. Techn. J. **49**, 291–307 (1970).
103. E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on IEEE 802.11ah: an enabling networking technology for smart cities," Comput. Commun. **58**, 53–69 (2015).
104. E. M. Khorov, A. G. Kiruanov, A. A. Kureev, and A. Lyakhov, "Study of mechanism for building a logical network topology in MANET," J. Commun. Technol. Electron. **60** (12), (2015). (In Press)
105. E. Khorov, A. Krotov, and A. Lyakhov, "Modeling machine type communication in IEEE 802.11ah network," in *Proc. IEEE Int. Conf. on Communications-Workshop on 5G & Beyond - Enabling Technologies and Applications, London, UK, June, 2015* (IEEE, New York, 2015).
106. S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo, "Higher-order correlation clustering for image segmentation," in Adv. Neural Inf. Process. Syst. **25**, 1530–1538 (2011).
107. J. M. Kleinberg, C. Papadimitriou, and P. Raghavan, "Segmentation problems," in *Proc. 30th ACM Symp. on Theory of Computing (STOC'1998), New York, NY, USA, 1998* (ACM, New York, 1998), 473–482 (1998).
108. D. E. Knuth and A. Raghunathan, "The problem of compatible representatives," SIAM J. on Disc. Math. **5**, 422–427 (1992).
109. G. Kochenberg, F. Glover, B. Alidaee, and H. Wang, "Clustering of microarray data via clique partitioning," J. Combin. Optim. **10**, 77–92 (2005).
110. H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "A general framework for increasing the robustness of PCA-based correlation clustering algorithms," in *Proc. 20th Int. Conf. Scientific and Statistical Database Management (SSDBM), Hong Kong, China, 2008* (Springer-Verlag, Berlin, 2008), pp. 418–435.
111. H.-P. Kriegel, P. Kroger, and A. Zimek, "Clustering high dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," ACM Trans. on Knowledge Discovery from Data (KDD) **3** (1), 1–58 (2009).
112. D. P. Kroese, R. Y. Rubinstein, and T. Taimre, "Application of the cross-entropy method for clustering and vector quantization," J. Global Optim. **37** (1), 137–157 (2007).
113. V. Kumar, M. Steinbach, and P.-N. Tan, *Introduction to Data Mining* (Addison-Wesley, 2005).
114. A. C. Kumari and K. Srinivas, Software module clustering using a fast multi-objective hyper-heuristic evolutionary algorithm," Int. J. of Appl. Inform. Syst. **5** (6), 12–18 (2012).
115. M. Kyperountas, A. Tefas, and I. Pitas, "Dynamic training using multistage clustering for face recognition," Pattern Recogn. **41**, 894–905 (2008).

116. Y. C. Lai, P. Lin, W. Liao, and C. M. Chen, "A region-based clustering mechanism for channel access in vehicular ad hoc networks," IEEE J. Selec. Areas Commun. **29**, 83–93 (2011).

117. J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," Internet Math. **6**, 29–123 (2009).

118. M. Sh. Levin, *Combinatorial Engineering of Decomposable Systems* (Kluwer, Dordrecht, 1998).

119. M. Sh. Levin, *Composite Systems Decisions* (Springer-Verlag, New York, 2006).

120. M. Sh. Levin, "Aggregation of composite solutions: strategies, models, examples," Electr. Prepr., 72 p., (Nov. 29, 2011); http://arxiv.org/abs/llll.6983 [cs.SE]

121. M. Sh. Levin, "Multiset estimates and combinatorial synthesis," Electr. prep., 30 p. (May 9, 2012); http://arxiv.org/abs/1205.2046 [cs.SY]

122. M. Sh. Levin, "Clique-based fusion of graph streams in multi-function system testing," Informatica **23**, 391–404 (2012).

123. M. Sh. Levin, *Modular System Design and Evaluation* (Springer-Verlag, New York, 2015).

124. M. Sh. Levin, "Towards combinatorial clustering: preliminary research survey," Electr. Prepr., 102 pp., (May 28, 2015); http://arxiv.org/abs/1505.07872 [cs.AI]

125. N. P. Lin, C.-I. Chang, H.-E. Chueh, H.-J. Chen, and W.-H. Hao, "A deflected grid-based algorithm for clustering analysis," WSEAS Trans. Comput. **3** (7), 125–132 (2007).

126. X. Liu, D. Li, S. Wang, and Z. Tao, "Effective algorithm for detecting community structure in complex networks based on GA and clustering," in *Proc. 7th Int. Conf. on Comput. Sci. ICCS'07, Beijing, China, May 27–30, 2007*, Ed. by Y. Shi et. al. (Springer-Verlag, Berlin, 2007).

127. X. Liua and T. Murata, "Detecting communities in k-partite k-uniform (hyper)networks," J. Comput. Sci. Technol. **26**, 778–791 (2011).

128. X. Liua, T. Murata, and K. Wakita, "Extending modularity by capturing the similarity attraction feature in the null model," Electr. Prepr., 10 p., (Feb. 12, 2013); http://arxiv.org/abs/1210.4007 [cs.SI]

129. Y. Lu, Y. Sun, G. Xu, and G. Liu, "A grid-based clustering algorithm for high-dimensional data streams," in *Lecture Notes in Computer Science (LNCS)*, Vol. 3584: *Advanced Data Mining and Applications (Proc. 1st Int. Conf. ADMA, Wuhan, China, July 22–24, 2005)* (Springer-Verlag, Berlin, 2005), pp. 824–831.

130. C. Mathieu and W. Schudy, "Correlation clustering with noisy input," in *Proc. 21st Ann. ACM-SIAM Symp. on Discrete Algorithms, Austin, TX, USA, Jan., 2010* (SIAM, 2010), pp. 712–728.

131. A. Mehrotra and M. A. Trick, "Cliques and clustering: A combinatorial approach," Oper. Res. Lett. **22** (1), 1–12 (1998).

132. A. Medius, G. Acuna, and C. O. Dorso, "Detection of community strcuture in networks via global optimization," Physica A **358**, 396–405 (2005).

133. S. Mimaroglu and M. Yagci, "CLICOM: Cliques for combining multiple clusterings," Expert Syst. Appl. **39**, 1889–1901 (2012).

134. B. Mirkin and I. Muchnik, "Combinatorial optimization in clustering," in *Handbook of Combinatorial Optimization*, Ed. by D.-Z. Du and P. M. Pardalos (Springer-Verlag, New York, 1999) vol. 2, pp. 261–329.

135. M. Mitchell, "Complex systems: Network thinking," Artif. Intell. **179**, 1194–1212 (2006).

136. J. W. Moon and L. Moser, "On cliques in graphs," Israel J. Math. **3** (1), 23–28 (1965).

137. E. Muller, I. Assent, S. Gunnemann, R. Krieger, and T. Seidl, "Relevant subspace clustering: Mining the most interesting non-redundent concepts in high dimensional data," in *Proc. 9th IEEE Int. Conf. on Data Mining, Miami, Florida, USA, Dec. 6–9, 2009* (IEEE, New York, 2009), pp. 377–386.

138. A. C. Muller, S. Nowozin, and C. H. Lampert, "Information theoretic clustering using minimum spanning trees," in *Lecture Notes in Computer Science*, Vol. 7476: *Proc. Joint 34th DAGM & 36th OAGM Symp. Pattern Recognition, Graz, Aug., 2012*, Ed. by A. Pinz et al. (Springer-Verlag, Berlin, 2012), pp. 205–215.

139. T. Murata, "Detecting communities from tripartite networks," in *Proc. World Wide Web. Conf. (WWW'2010), Raleigh, North Carolina, USA, Apr. 26–30, 2010*, (Springer-Verlag, Berlin, 2010), pp. 1159–1160.

140. T. Murata, "Modularity for heterogeneous networks," in *Proc. 21th ACM Conf. on Hypertext and Hypermedia (HyperText'2010), Toronto, Canada, June 13–16, 2010* (ACM, 2010), pp. 129–134.

141. L. M. Naeni, R. Berretta, and P. Moscano, "MA-Net: A reliable memetic algorithm for community detection by modularity optimization," in *Proc. 18th Asia Pac. Symp. on Intell. & Evol. Syst. Nov. 2014*, Ed. by H. Handa et. al., (Springer-Verlag, 2015), Vol. 1, pp. 311–323.

142. M. E. J. Newman, "Fast algorithm for detecting community structure in networks," Electr. Prepr., 5 p., (Sep. 22, 2003); http://arxiv.org/abs/0309508 [cond-mat.stat-mech]

143. M. E. J. Newman, "Detecting community structure in networks," Eur. Phys. J. B 38(2), 321–330 (2004).

144. M. E. J. Newman, "Modularity and community structure in networks," Proc. Natl. Acad. Sci. USA **103**, 8577–8582 (2006).

145. M. E. J. Newman, *Networks: an Introduction* (Oxford Univ. Press, Oxford, 2010).

146. M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," Electr. Prepr., 16 p., (Aug. 11, 2003); http://arxiv.org/abs/0308217 [cond-mat.stat-mech]

147. A. Noack and R. Rotta, "Multi-level algorithms for modularity clustering," Electr. Prepr., 12 p., (Dec. 22, 2008); http://arxiv.org/abs/0812.4073 [cs.DC]

148. M. Oosten, J. G. C. Rutten, and F. C. R. Spieksma, "The clique partioning problem: Facets and patching facets," Networks **38**, 209–226 (2001).

149. I. H. Osman and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search," Int. Trans, Oper. Res. **1**, 317–336 (1994).

150. R. E. Osteen and J. T. Tou, "A clique-detection algorithm based on neighborhoods in graphs," Int. J. Comput. Inf. Sci. **2**, 257–268 (1973).
151. P. R. J. Ostergard, "A new algorithm for the maximum-weight clique problem," in *Proc. Electr. Notes in Disc. Math., 6th Twente Workshop on Graphs and Combinatorial Optimization, 1999* (Univ. Twente, Enschede, Netherlands, 1999), vol. 3, pp. 153–156.
152. M. Ovelgonne and A. Geyer-Schulz, "A comparison of agglomerative hierarchical algorithms for modularity clustering," in *Challenges at the Interface of Data Analysis, Computer Science, and Optimization, 2012* (Springer-Verlag, Berlin, 2012), pp. 225–232.
153. T. Ozyer and R. Alhajj, "Parallel clustering of high dimensional data by integrating multi-objective genetic algorithm with divide and conquer," Appl. Intell. **31**, 318–331 (2009).
154. N. Paivinen, "Clustering with a minimum spanning tree of scale-free-like structure," Pattern Recogn. Lett. **26**, 921–930 (2005).
155. P. M. Pardalos and J. Xue, "The maximum clique problem," J. Global Optim. **4**, 301–328 (1994).
156. P. Pardalos, M. Batzyn, and E. Maslov, "Cliques and quasi-cliques in large graphs: theory and applications," in *Proc. Int. Conf. on Disc. Optim. & Oper. Res. DOOR-2013, Novosibirsk, June 24–28, 2013* (Sobolev Inst. Math., Novosibirsk, 2013).
157. N. H. Park and W. S. Lee, "Statistical grid-based clustering over data streams," ACM SIGMOD Record **33**, 32–37 (2004).
158. M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," IEEE Trans. Pattern. Anal. Mach. Intell. **29**, 167–172 (2007).
159. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules* (Wiley, Hoboken, NJ, 2005).
160. S. J. Peter and S. P. Victor, "A novel algorithm for dual similarity clusters using minimum spanning tree," J. Theor. Appl. Inform. Technol. **14**, 60–66 (2010).
161. S. Pettie and V. Ramashandran, "An optimal minimum spanning tree algorithm," J. ACM **49**, 16–34 (2002).
162. P. Pons and M. Latapy, "Computing communities in large networks using random works," J. Graph. Alg. Appl. **10**, 191–218 (2006).
163. M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks". Notices AMS **56**, 1082–1097, 1164 (2009).
164. J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," Phys. Rev. E **74**, 016110, (2006).
165. C. Rocha, L. C. Dias, and I. Dimas, "Multicriteria classification with unknown categories: A clustering-sorting approach and an application to conflict management," J. Multi-Cri. Dec. Anal. **20**, 13–27 (2013).
166. C. Rocha and L. C. Dias, "MPOC - an agglomerative algorithm for multicriteria partially ordered clustering," Quart. J. Operat. Res. (4OR) **11**, 253–273 (2013).
167. M. Rosvall and C. T. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks," Proc. Natl. Acad. Sci. USA (PNAS) **104**, 7327–7331 (2007).
168. B. Roy, *Multicriteria Methodology for Decision Aiding* (Kluwer, Dordrecht, 1996).
169. R. Y. Rubinstein, "Cross-entropy and rare-events for maximal cut and partition problems," ACM Trans. Model. Comput. Simul. **12** (1), 27–53 (2002).
170. F. Saeed, N. Salim, and A. Abdo, "Voting-based consensus clustering for combining multiple clusterings of chemical structures," J. Cheminf. **4** (37), 1–8 (2012).
171. J. Salzmann, R. Behnke, M. Gag, and D. Timmermann, "4-MASCLE - improved coverage aware clustering with self healing abilities," in *Proc. IEEE Symp. & Workshops on Ubiquitous, Autonomic and Trusted Computing (UIC-ATC'09), Brisbane, Australia, July 7–9, 2009* (IEEE, New York, 2009), pp. 537–543.
172. S. E. Schaeffer, "Graph clustering," Comput. Sci. Rev. **1**, 27–64 (2007).
173. A. Schenker, M. Last, H. Bunke, and A. Kandel, "Classification of web documents using graph matching," Int. J. Pattern Recognit. Artif. Intell. **18**, 475–496 (2004).
174. S. Selim and K. Alsultan, "A simulated annealing algorithm for the clustering problems," Pattern Recogn. **24**, 1003–1008 (1991).
175. H. M. Selim, R. G. Askin, and A. J. Vakharia, "Cell formation in group technology: review, evaluation and direction for future research," Comput. Ind. Eng. **34** (1), 3–20 (1998).
176. R. Shamir, R. Sharan, and D. Tsur, "Cluster graph modification problems," in *Lecture Notes in Computer Science,* Vol. 2573: *Proc. 28th Int. Workshop on Graph-Theoretic Concepts in Computer Science, Cesky Krumlov, Czech Republic, June 13-15, 2002* (Springer-Verlag, Berlin, 2002), pp. 379–316.
177. R. Shamir, R. Sharan, and D. Tsur, "Cluster graph modification problems," Disc. Appl. Math. **144**, 173–182 (2004).
178. G. Sheikholeslami, C. Chattterjee, and A. Zhang, "WaveCluster: a wavelet-based clustering approach for spatial data in very large databases," The VLDB J. **8**, 289–304 (2000).
179. H. Shiokawa, Y. Fujiwara, and M. Onizuka, "Fast algorithm for modularity-based graph clustering," in *Proc. 27th AAAI Conf. on Artificial Intelligence (AAAI 2013), Bellevue, WA, USA 2013* (AAAI, 2013), 1170–1176.
180. D. A. Spielman and S.-H. Teng, "A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning," SIAM J. Comput. **42**, 1–26 (2013).
181. G. Srinivasan, "A clustering algorithm for machine cell formation in group technology using minimum spanning tree," Int. J. Prod. Res. **32**, 2149–2158 (1994).
182. C. S. Sung and H. W. Jin, "A Tabu-search-based heuristic for clustering," Pattern Recogn. **33**, 849–858 (2000).
183. C. Swamy, "Correlation clustering: maximizing agreements via semidifinite programming," in *15th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA), New Orleans, LA, Jan., 2004* (SIAM, 2004), pp. 526–527.
184. J. Tabor and P. Spurek, "Cross-entropy clustering," Pattern Recogn. **47**, 3046–3059 (2014).
185. J. Tillet, R. Rao, and F. Sahin, "Cluster-head identification in ad hoc sensor networks using particle swam optimization," in *Proc. 2002 IEEE Int. Conf. on Personal Wireless Commun., New Delhi, India, Dec. 2002* (IEEE, Piscataway, 2002), pp. 201–205.

186. A. Trifunovic and W. J. Knottenbelt, "Parallel multilevel algorithms for hypergraph partitioning," J. Paral. Distr. Comput. **68**, 563–581 (2008).
187. C.-F. Tsai and C.-C. Yen, "ANGEL: a new effective and efficient hybrid clustering techniques for large databases," in *Lecture Notes in Computer Science*, Vol. 4426: *Proc. 11th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'2007), Nanjing, China, May 22–25, 2007*, Ed. by Z.-H. Zhou et. al. (Springer-Verlag, Berlin, 2007), pp. 817–824.
188. C.-F. Tsai, H.-F. Yeh, J.-F. Chang, and N.-H. Liu, "PHD: an efficient data clustering scheme using partition space technique for knowledge discovery in large databases," Appl. Intell. **33** (1), 39–53 (2010).
189. C.-W. Tsai, H.-J. Song, and M.-C. Chiang, "A hyperheuristic clustering algorithm," in *Proc. 2012 IEEE Int. Conf. on Systems, Man, Cybernetics (SMC'2012) Seoul, Korea (South), Oct. 14–17, 2012* (IEEE, New York, 2012), pp. 2839–2844.
190. L. Y. Tseng and S. B. Yang, "A genetic approach to the automatic clustering problem," Pattern Recogn. **34**, 415–424 (2001).
191. K. Tsuda and T. Kudo, "Clustering graphs by wieghted substructure mining," in *Proc. 23rd Int. Conf. on Mach. Learn., Pittsburgh, Pennsylvania, USA, June 25–29, 2006* (Carnegie Mellon Univ., Pittsburgh, 2006), pp. 953–960.
192. K. Turner and A. K. Agogino, "Ensemble clustering with voting active clusters," Pattern Recogn. Lett. **29**, 1947–1953 (2008).
193. D. W. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swam optimization," in *Proc. 2003 IEEE Congr. on Evolutionary Computation (CEC'2003), Newport Beach, California, June, 2003* (IEEE, New York, 2003), Vol. 1, pp. 215–220.
194. A. Vashist, C. A. Kulikowsky, and I. Muchnik, "Orthlog clustering on a multipartite graph," IEEE/ACM Trans. Comput. Biology Bioinform. **4**, 17–27 (2007).
195. S. Vega-Pons and J. Ruiz-Schulcloper, "A survey of clustering ensemble algorithms," Int. J. Pattern Recogn. Artif. Intell. **25**, 337–372 (2011).
196. K. Wakita and T. Tsusumi, "Finding community structure in mega-scale social networks," Electr. Prepr., 9 p., Fev. 8 (2007); http://arxiv.org/abs/0702.2048 [cs.CY]
197. X. Wang, X. Wang, and Wikes, "A divide-and-conquer approach for minimum spanning tree-based clustering," IEEE Trans. Knowledge Data Eng. (KDE) **21**, 945–958 (2009).
198. Q. Wang and E. Fleury, "Overlapping community structure and modular overlaps in complex networks," in *Lecture Notes in Social Networks,* Part: *Mining Social Networks and Security Informatics*, Ed. by T. Ozyer et al. (Springer, Berlin, 2013), pp. 15–40.
199. S. White and P. Smyth, "A spectral clustering approach to finding communities in graph," in *Proc. SIAM Data Mining Conf., Trondheim, Norway, Aug. 30–Sept. 2, 2005* (SIAM, 2005), pp. 76–84.
200. J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," ACM Comp. Surv. **45** (4) art. 443 (2013).
201. Y. Xu, V. Olman, and D. Xu, "Minimum spanning trees for gene expression data clustering," Genome Inf. **12**, 24–33 (2001).
202. X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "SCAN: a structural clustering algorithm for networks," in *Proc. Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD-07), San Jose, Aug. 2007* (ACM, 2007), 824–833.
203. B. Yan and S. Gregory, "Detecting communities in networks by merging cliques," in *Proc. 2nd Int. Conf. on Interaction Sciences: Information Technology, Culture and Human (ICIS 2009), Seoul, Korea (South), Nov. 24–26, 2009* (IEEE, New York, 2009), 832–836.
204. Y. Yang and M. S. Kamel, "An aggregated clustering approach using multi-ant colonies algorithms," Pattern Recogn. **39**, 1278–1289 (2006).
205. J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proc. 6th ACM Int. Conf. on Web Search and Data Mining (WSDM'2013), Rome, Feb. 4–8, 2013* (ACM, 2013), 587–596.
206. J. Yang and J. Leskovec, "Overlapping communities explain core-periphery organization of networks," Proc. IEEE **102**, 1892–1902 (2014).
207. J. Yang and J. Leskovec, "Structure and overlaps of ground-truth communities in networks," ACM Trans. Intell. Syst. Technol. (TIST) **15** (2), art. 26 (2014).
208. J. Yang and J. Leskovec, "Designing and evaluation network communities based on ground-truth," Knowl. Inf. Syst. **42** (1), 181–213 (2015).
209. A. C. Yao, "An $O(|E|\log\log|V|)$ algorithm for finding minimum spanning trees," Inf. Process. Lett. **4** (1), 21–23 (1975).
210. D. Y. Yeh, "A dynamic programming approach to the complete set partitioning problem," BIT Numer. Math. **26**, 467–474 (1986).
211. O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," IEEE Networks **20** (3), 20–25 (2006).
212. W. W. Zachary, "An information flow model for conflict and fission in small groups," J. Anthropol. Res. **33**, 452–473 (1977).
213. E. Ziv, M. Middendorf, and C. Wiggins, "Information-theoretic approach to network modularity," Phys. Rev. E **71**, 046117 (2005).
214. C. Zopounidis and M. Doumpos, "Multicriteria classification and sorting methods: a literature review," Eur. J. Operat. Res. (EJOR) **138**, 229–246 (2002).