Table of Contents

[1. CHC Informal Description 2](#_Toc72415052)

[2. CHC Formal Description 4](#_Toc72415053)

[2.1. Constrained Clustering Problem – HDBSCAN\* 4](#_Toc72415054)

[2.1.1. Definition 4](#_Toc72415055)

[2.1.2. Implications 4](#_Toc72415056)

[2.1.3. Projection 4](#_Toc72415057)

[2.1.4. Coordination Protocol 4](#_Toc72415058)

[2.2. Content-Based Reciprocal Recommender System Problem (Palomares, Porcel, Pizzato, Guy, & Herrera-Viedma, 2021) 5](#_Toc72415059)

[2.2.1. Definition 5](#_Toc72415060)

[2.2.2. Implications 5](#_Toc72415061)

[2.2.3. Projection 5](#_Toc72415062)

[2.2.4. Coordination Protocol 8](#_Toc72415063)

[3. Analyzed Algorithms 8](#_Toc72415064)

[3.1. Constrained Optimization Problems 9](#_Toc72415065)

[3.1.1. COP (Fioretto, Pontelli, & Yeoh, 2018) 9](#_Toc72415066)

[3.1.2. DCOP (Fioretto, Pontelli, & Yeoh, 2018) 9](#_Toc72415067)

[3.1.3. ADCOP (Fioretto, Pontelli, & Yeoh, 2018) 10](#_Toc72415068)

[3.1.4. MODCOP (Fioretto, Pontelli, & Yeoh, 2018) 11](#_Toc72415069)

[3.2. Clustering Algorithms 12](#_Toc72415070)

[3.2.1. K-Means (Manning, Raghavan, & Schü, 2008) 12](#_Toc72415071)

[3.2.2. Faster Exact K-Means (Borgelt, 2020) 12](#_Toc72415072)

[3.2.3. Hierarchical Clustering (Nielsen, 2016) 12](#_Toc72415073)

[3.2.4. DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) 12](#_Toc72415074)

[3.3. Constrained Clustering Algorithms 13](#_Toc72415075)

[3.3.1. BCOP-Kmeans 13](#_Toc72415076)

[3.3.2. CoExDBSCAN (Ertl, Meyer, Schneider, & Streit, 2020) 13](#_Toc72415077)

[3.3.3. COP-KMeans (Wagstaff, Cardie, Rogers, & Schrödl, 2001) 13](#_Toc72415078)

[3.3.4. SCOP-KMeans (Wagstaff K. L., 2002) 20](#_Toc72415079)

[3.4. Recommendation Systems (Beel, Gipp, Langer, & Breit, 2015) 20](#_Toc72415080)

[3.4.1. Content Based 20](#_Toc72415081)

[3.4.2. Collaborative Filtering 20](#_Toc72415082)

[3.4.3. Group Recommendation-RS (Boratto, 2016) 20](#_Toc72415083)

[3.5. Reciprocal Recommendation Systems (Palomares, Porcel, Pizzato, Guy, & Herrera-Viedma, 2021) 21](#_Toc72415084)

[3.5.1. CF-RRS 21](#_Toc72415085)

[3.5.2. Hybrid-RRS 21](#_Toc72415086)

[3.5.3. Group Formation-RRS (Yacef & McLaren, 2015) 21](#_Toc72415087)

[3.6. Coalition Structure Generation Problem (Präntare & Heintz, 2020) 21](#_Toc72415088)

[3.6.1. ACS (Unknown, 2021) 21](#_Toc72415089)

[3.6.2. ODP-IP (Michalak, Rahwan, Elkind, Wooldridge, & Jennings, 2016) 22](#_Toc72415090)

[3.6.3. ODSS (Changder, Aknine, Ramchurn, & Dutta, ODSS: Efficient Hybridization for Optimal Coalition Structure Generation, 2020) 22](#_Toc72415091)

[3.7. Clustering Ensemble Algorithm 22](#_Toc72415092)

[3.7.1. ACE (Alqurashi & Wang, Clustering ensemble method, 2019) 22](#_Toc72415093)

[References 23](#_Toc72415094)

# CHC Informal Description

*Context.* Refugees and asylum seekers (TCNs) in Greece need to leave their accommodation (the one that exist via specific programs) when they have the refugee status. Due to the high number of TCNs, it is a challenging task for PRAKSIS to provide help to all TCNs for finding new accommodation facilities. Therefore, they are informed about specific programs that exist, however, due to limitations in the capacity, there is also a high number of them searching for apartments in order to share with other refugees so that to jointly afford related costs.

*Rationale.* This, in turn, requires coordination support of TCNs to find candidates with whom they could share some flat based on most similar preferences.

*MyWELCOME contribution.* In WELCOME, the co-habitation coordination (CHC) scenario is concerned with helping TCNs to form groups of candidates for sharing an apartment according to their given individual and group constraints. The agents, however, neither search for available (rentable space in) rental apartments, nor do they match groups of TCNs with landlords of apartments for rent. These activities may be performed by the proposed groups of TCNs who best match with each other according to their individual preferences and some general hard (group) constraints of sharing some rental apartment in Greece. During self-coordination, TCNs may be assigned to more than one group of candidates for apartment sharing. In the CHC scenario, we consider the following family types: Single man; Single woman; 2-member Nuclear family; 3-member Nuclear family; 4-member Nuclear family; Single-parent (mother) family; Single-parent (father) family; Extended family.

At the end of a CHC process, each participating TCN shall be informed by his/ her agent about the recommended (approximately) optimal cohabitation group for her and the respective contact addresses of the other group members. The WELCOME system should make suggestions to the TCNs for their grouping with respect to co-habitation based on the following constraints:

1. **Individual constraints** specified by the TCNs (preferences)
2. **Age**: A TCN could prefer to belong to a group of certain age group(s), to the extent possible, as they are more likely to share common interests. Recommended age groups for exclusive preference settings by TCNs: *18 – 25, 26 – 33, 34 – 43, 44 – 50, 50 – 65, and 65+.*
3. **Gender**: A TCN could prefer being member of *Male*, *Female* or *Other* gender group.
4. **Family:** A TCN could prefer to belong to a cohabitation group certain family type(s). Recommended family types for exclusive preference settings by TCNs: *Single man, Single woman, 2-member Nuclear family, 3-member Nuclear family, 4-member Nuclear family, Single-parent (mother) family, Single-parent (father) family, and Extended family.*
5. **Nationality**: Although many people do not have a problem when working with other nationalities, some of the TCNs might have. To avoid respective conflicts and communication problems in co-habitation groups, the agent should respect the preference of its TCN about being assigned to a group with other nationalities than her own one, or not. A TCN could prefer being member of *same or mixed* gender group.
6. **Religion**: A TCN could prefer to be in a *mixed or same* religion group.
7. **Ethnicity**: A TCN could prefer to be in a *mixed or same* ethnic group.
8. **Apartment preferences:** A TCN could specify rental flat or apartment related preferences
   1. **Location (area) of apartment**: A TCN could specify one or multiple of the following locations listed below, as preference. Predefined locations: *Ampelokipoi, Menemeni, Kalamaria, Eleftherio-Kordelio, Evosmos, Agios Pavlos, Neapoli, Pefka, Sykies, Nea Efkarpia, Polichni, Stavroupoli, Pylaia, Thessaloniki, Triandria.*
   2. **Accessibility (disabled accessible)**: A TCN could select *yes or no* for availability of accessibility.
   3. **Rental period (min-max/from-to):** A TCN could specify a *start and end date* as a range, for his/her preferred rental period.
   4. **Share with**: A TCN could specify a *minimum and maximum number* as a range as the number of people that he/she prefers to share the apartment with.
9. **Cohabitation group constraints**
10. All members of a cohabitation group have most similar individual constraints (preferences) for rental apartment sharing.
11. All members of a cohabitation group should be able to communicate among each other in at least one common language.

***Please note that the cohabitation group constraints above are key/informal points that the TCNs themselves could adjust (with PRAKSIS) if appropriate.***

# CHC Formal Description

Detailed information about the solutions for CHC scenario will be in this section.

## HDBSCAN\* - Constrained Clustering Problem Solver

### Definition

\*\* Temporary note\*\*

\*\*Detailed information in terms of complexities, characteristics of HDBCSAN\* algorithm can be found in the following link:

<https://cloud.dfki.de/owncloud/index.php/s/RHmTStqcRdkcRC3>

**Also in the file**: Shared->Project WELCOME-> WP3 Agent-Based Semantic Service Coordination->Task 3.2 - MyWelcome Agent Teams-> Scenarios->*Algorithm characteristics\_for\_LCC\_CHC.xlsx*

*\*\**

* What is constrained clustering?
* Why is it used?
* Refer to original paper for formal description (objective function)
* Other constrained clustering algorithm are in Analyzed Algorithms section
* Briefly explain HDBSCAN\* and refer to the paper for details (Campello, Moulavi, Zimek, & Sander, 2015)
* Write the complexities and other characteristics

### Implications

* Reduction implications: how CHC is reduced into constrained clustering?
* Advantages of HDBSCAN\*. Why it can solve CHC

### Projection

* Write the objective function and constraints (ML&CL and system) in the objective function adapted for CHC.

### Coordination Protocol

* Assumptions in the protocol
* Pseudocode of the Coordination protocol of LCC
* Runtime and communication complexities

## Content-Based Reciprocal Recommender System (Palomares, Porcel, Pizzato, Guy, & Herrera-Viedma, 2021)

\*\* Temporary note\*\*

\*\*Detailed information in terms of complexities, characteristics of CB-RRS algorithm can be found in the following link:

<https://cloud.dfki.de/owncloud/index.php/s/RHmTStqcRdkcRC3>

**Also in the file**: Shared->Project WELCOME-> WP3 Agent-Based Semantic Service Coordination->Task 3.2 - MyWelcome Agent Teams-> Scenarios->*Algorithm characteristics\_for\_LCC\_CHC.xlsx*

\*\*

### Definition

* What is RRS?
* Types of RRS: CB-RRS, CF-RRS, Hybrid RRS
* Write about CB-RRS since it is the solution. “*Details about CF-RRS and Hybrid RRS can be found in analyzed algorithms section*”
* Refer to original paper for formal description (objective function)
* Mention the complexities and other characteristics

### Implications

* Reduction implications: how CHC is reduced into CB-RRS?
* Advantages of CB-RRS

### Projection

* The objective function and constraints.
* Projection of constraints, etc.

**Distance between Ranges Formalization**

Following formulas apply to date and number range.

**Compatibility Score**

**Input**: Preferences of user , users (all users except x)

**Output**: List of compatibility scores for each user in

Add into

**Reciprocal Score**

**Input**: asymmetric compatibility matrix

**Output**: symmetric reciprocal matrix

**Recommender**

**Input**:

**Output**:

// all users except user i

// sort recommendations by reciprocal score

**CHC Example**

Example Personal info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | u\_1 | u\_2 | u\_3 | u\_4 | u\_5 |
| **Age** | 19 | 47 | 29 | 60 | 25 |
| **Gender** | Male | Male | Female | Male | Female |
| **Family Type** | Single-man | 4-memb. Nuclear family | Single-parent (mother) | 3-memb. Nuclear family | Single-woman |
| **Nationality** | N\_1 | N\_2 | N\_4 | N\_3 | N\_2 |
| **Religion** | R\_3 | R\_4 | R\_3 | R\_2 | R\_1 |
| **Ethnicity** | E\_2 | E\_1 | E\_3 | E\_1 | E\_4 |

Example Preference info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | u\_1 | u\_2 | u\_3 | u\_4 | u\_5 |
| **Age** | 18-25 | 34-43, 44-50 | 26-33, 34-43 | Don’t mind | 18-25, 26-33 |
| **Gender** | Don’t mind | Don’t mind | Female | Don’t mind | Female |
| **Family Type** | Single-man | 3-memb. Nucl. family, 4-memb. Nucl. family | Single-woman, Single-parent (mother) | 2-memb. Nucl. family, 3-memb. Nucl. family, 4-memb. Nucl. family | Single-woman, Single-parent (mother) |
| **Nationality** | Mixed | Same | Don’t mind | Don’t mind | Same |
| **Religion** | Same | Same | Don’t mind | Same | Don’t mind |
| **Ethnicity** | Don’t mind | Same | Don’t mind | Don’t mind | Don’t mind |
| **Location** | L1 | Don’t mind | L2, L3, L4 | Don’t mind | L1, L2, L3, L4 |
| **Accessibility** | Don’t mind | Don’t mind | Don’t mind | Yes | No |
| **Rent Period** | 1/1/2021-1/7-2021 | 1/3/2021-1/3/2022 | 15/6/2021 – 1/1/2022 | Don’t mind | Don’t mind |
| **Share with** | 1-3 | Don’t mind | 1-3 | 2-5 | Don’t mind |

**Compatibility scores with default weights:**

**ALL VALUES FROM NOW ON ARE MADE UP**

**Compatibility matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | U\_1 | U\_2 | U\_3 | U\_4 | U\_5 |
| U\_1 |  | 0.36 | 0.29 | 0.4 | 0.21 |
| U\_2 | 0 |  | 0.17 | 0.31 | 0.25 |
| U\_3 | 0.2 | 0.24 |  | 0.19 | 0.47 |
| U\_4 | 0.5 | 0.22 | 0.15 |  | 0.19 |
| U\_5 | 0.33 | 0.47 | 0.41 | 0 |  |

**Reciprocal Score Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | U\_1 | U\_2 | U\_3 | U\_4 | U\_5 |
| U\_1 |  | 0 | 0.24 | 0.44 | 0.26 |
| U\_2 |  |  | 0.2 | 0.26 | 0.33 |
| U\_3 |  |  |  | 0.17 | 0.44 |
| U\_4 |  |  |  |  | 0 |
| U\_5 |  |  |  |  |  |

**Recommendations**

### Coordination Protocol

* Assumptions in the protocol
* Pseudocode of the Coordination protocol of LCC
* Runtime and communication complexities

# Analyzed Algorithms

\*\* Temporary note\*\*

\*\*Detailed information in terms of complexities, characteristics of the algorithms listed below, can be found in the following link:

<https://cloud.dfki.de/owncloud/index.php/s/RHmTStqcRdkcRC3>

**Also in the file**: Shared->Project WELCOME-> WP3 Agent-Based Semantic Service Coordination->Task 3.2 - MyWelcome Agent Teams-> Scenarios->*Algorithm characteristics\_for\_LCC\_CHC.xlsx*

\*\*

## Constrained Optimization Problems

### COP (Fioretto, Pontelli, & Yeoh, 2018)

**Definition**:

* *explain briefly in your own words*
* *refer to the original paper for formal description of the algorithm*

COPs consist of variables, values and weighted constraints. COP solving algorithms assign values to variables such that the constraints are satisfied as much as possible. Since it is not always possible to satisfy all of the constraints, COP allows some constraints to be violated to some extent.

**Implications**:

* *Strongest argument whether to be ruled out or to be considered*
* *List the characteristics of algorithm*
* *Refer to the big characteristics table for comparison between algorithms*

COP time complexity is exponential which is not acceptable for LCC scenario. As well as COPs are solved in central manner. Distributed version of COP is in the next section.

### DCOP (Fioretto, Pontelli, & Yeoh, 2018)

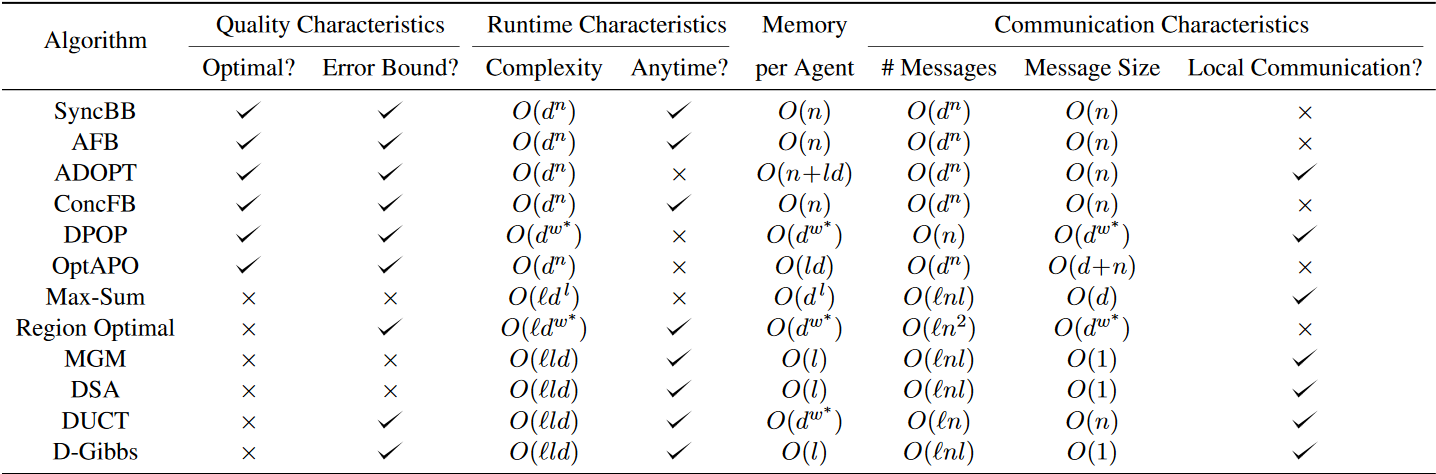
**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

DCOP is an extension of the COP framework to the multi-agent case, where agents control variables and constraints and need to coordinate the value assignment for the variables they control so as to optimize a global objective function. A DCOP is a tuple (A, X, D, F, α), where:

* A is a finite set of agents
* X is a finite set of variables
* D is a finite set of domain sets. Each domain set corresponds to one variable in X.
* F is a finite set of cost functions.
* α is a function which assigns the control of each variable to an agent.

Finding an optimal solution for DCOP is **NP-HARD problem**.



**Figure 1**. List of DCOP algorithms (Fioretto, Pontelli, & Yeoh, 2018)

* **n** refers to the number of variables
* **d** refers to the size of the largest domain
* **w\***refers to the induced with of the pseudo-tree
* **l** refers to the largest number of neighboring agents

**i** refers to the number of iterations in incomplete algorithms

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

Like COP, DCOP doesn’t formulate/generate clusters by nature. In DCOP, each agent assigns predefined values, to its variable/variables in a distributed/decentralized manner. Therefore, the groups which the TCNs can be assigned to are given as the domain for each TCN (variable).

Additionally, all agents controlling variables which are in the same cost function, incur the same cost as each other.

Figure 1 illustrates the list of DCOP algorithms with their properties such as optimality, anytime, computational and communication complexities etc.

In the worst case, all optimal DCOP algorithms will take exponential time: **O(dn)**.

So if there are 20 TCNs and each group can contain at least 2 TCNs, then in the worst case there will be 10 groups to cover all 20 TCNs. That means O(1020) runtime complexity for the optimal DCOP algorithms.

Therefore, DCOP doesn’t fit to LCC problem. Since ADCOP and MODCOP solving algorithms inherit DCOP algorithms, they don’t fit as well. Short information about ADCOP and MODCOP can be found in the next sections.

### ADCOP (Fioretto, Pontelli, & Yeoh, 2018)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

ADCOP is a tuple (A, X, D, F, α) just like DCOP. In ADCOP, each agent can incur different costs from the same cost function. This is the most important difference between DCOP and ADCOP.

**ADCOP Algorithms:**

There are 2 approaches to solve ADCOPs:

* **Two-phase strategy**: where only one side of the constraint (i.e., the cost induced by one agent) is considered in the first phase. The other side(s) (i.e., the cost induced by the other agent(s)) is considered in the second phase once a complete assignment is produced. As a result, the costs of all agents are aggregated.
* **One-phase strategy**: is to systematically check both sides of constraints before reaching a full assignment.

DCOP algorithms are extended with the 2 strategies (one-phase and two-phase) to solve ADCOPs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Extension of** | **Exact** | **Runtime** | **Memory per Agent** | **Number of Messages** | **Message size** |
| SyncABB-2ph | SyncBB | Yes | O(dn) | O(n) | O(dn) | O(n) |
| SyncABB-1ph | SyncBB | Yes | O(dn) | O(n) | O(dn) | O(n) |
| ATWB | AFB | Yes | O(dn) | O(n) | O(dn) | O(n) |
| ACLS | DSA | No | O(ild) | O(l) | O(inl) | O(l) |
| MCS-MGM | MGM | No | O(ild) | O(l) | O(inl) | O(l) |

**Figure 2**. List of ADCOP algorithms (Fioretto, Pontelli, & Yeoh, 2018)

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

Since ADCOP solving algorithms are simply an extended version of DCOP algorithms, they grow exponentially as well. Therefore, LCC cannot be solved as ADCOP.

### MODCOP (Fioretto, Pontelli, & Yeoh, 2018)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

MODCOP is basically an extension of DCOP with multiple conflicting objectives where all objectives need to be optimized simultaneously. It is a tuple where A, X, D and are same as ADCOP.

* is a finite set of objective functions.

Therefore, the goal in MODCOP is to minimize the sum of all objective functions but it typically doesn’t achieve to optimize all of them. Therefore, there are tradeoffs between different objectives.

Algorithms

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Extension of** | **Exact** | **Runtime** | **Memory per Agent** | **Number of Messages** | **Message size** |
| MO-SBB | SyncBB | Yes | O(dn) | O(np) | O(dn) | O(n) |
| Pseudo-tree based algorithm | ADOPT | Yes | O(dn) | O(np) | O(dn) | O(n) |
| B-MOMS | Bounded Max-Sum | No | O(idl) | O(pdn) | O(inl) | O(pdn) |
| DP-AOF | AOF | No | O(dw\*) | O(dw\*) | O(hn) | O(dw\*) |
| MO-DPOPLp | DPOP | No | O(dw\*) | O(dw\*) | O(hn) | O(dw\*) |
| DIPLS | Pareto Local Search | No | O(ikp) | O(np) |  | O(np) |

**Figure 3**. MODCOP algorithms (Fioretto, Pontelli, & Yeoh, 2018)

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

Since MODCOP can be considered as multiple DCOPs and we said that LCC cannot be solved as DCOP, LCC cannot be solved as MODCOP as well.

## Clustering Algorithms

### K-Means (Manning, Raghavan, & Schü, 2008)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### Faster Exact K-Means (Borgelt, 2020)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### Hierarchical Clustering (Nielsen, 2016)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### DBSCAN (Ester, Kriegel, Sander, & Xu, 1996)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

## Constrained Clustering Algorithms

### BCOP-Kmeans

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### CoExDBSCAN (Ertl, Meyer, Schneider, & Streit, 2020)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### COP-KMeans (Wagstaff, Cardie, Rogers, & Schrödl, 2001)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

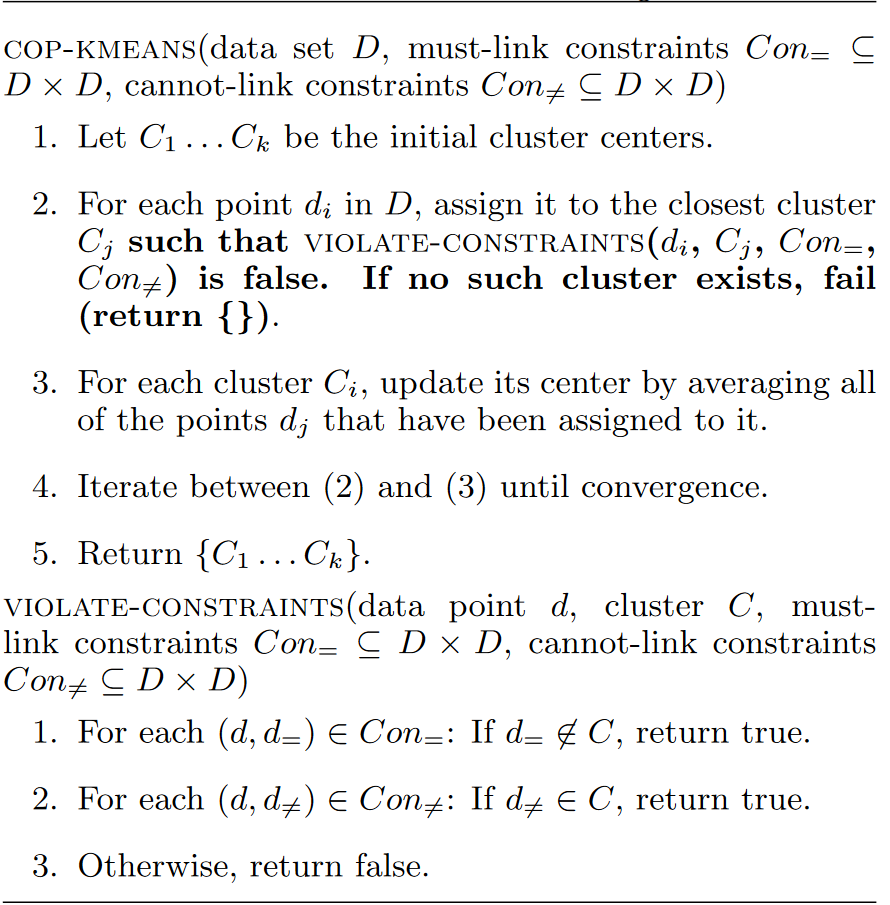
Clustering is an unsupervised method for data analysis. In clustering, the data points are grouped based on a similarity unit. The **objective** of clustering is to group a set of objects in such a way that the objects are similar intra-group and dissimilar inter-group as much as possible such that distinct groups of the objects can be identified.

There are some clustering methods such as hierarchical clustering, k-means clustering and etc. In k-means clustering, the number of groups is required to be known beforehand which is represented by K such that the algorithm divides the data points into K clusters.

Constrained clustering is a method which takes into account the constraints between data points. These constraints are represented either as must-link or cannot-link.

So the definition of constrained k-means clustering is “it divides the data points into k clusters while ensuring that the constraints (must-link and cannot-link) are not violated”. Additionally, it is a polynomial time algorithm unlike the ones we talked about so far. The complexity of COP-KMeans is O(kN) where N and k are the number of data points and clusters, respectively. (Masayuki & Seiji, 2018)

The traditional constrained k-means algorithm is called COP-KMeans and it is described in **figure 4**.



**Figure 4**. Constrained K-means algorithm (Wagstaff, Cardie, Rogers, & Schrödl, 2001)

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

The objective of LCC is to maximize the degree of suitability of working groups. A working group corresponds to a cluster in a clustering problem. In a clustering problem, the objective is to maximize the intra-group similarity and minimize the inter-group dissimilarity. In a constrained clustering problem, in order to improve the quality of the clusters, the constraints are used as the background knowledge about the objects to direct the algorithm into a better solution.

In LCC problem, language instructor didn’t request to form working groups of TCNs based on only their language course performance but she requested to form working groups in which the TCNs would perform well together. Therefore, in LCC\_CKmeans the objective is to minimize the distance and the violations of individual preferences and group conditions which are specified by language instructor.

In other words, LCC\_CKmeans aims to divide TCNs into small groups based on their CPL while satisfying individual preferences and group conditions. Therefore, LCC problem can be solved with COP-Kmeans clustering algorithm. Since the algorithm is polynomial, the solution of LCC might be faster compared to exponential algorithms (e.g. BOSS & FACS).

**Projection: TO BE REMOVED**

The projection of LCC as CKMeans problem consists of 8 steps: the input, generation of must-link (ML) and cannot-link (CL) constraints, calculation of k range and k (number of clusters), initialization of cluster centers, assignment of an agent to a cluster, update of cluster centers, convergence and the output.

1. The input

As the input, the COP-Kmeans algorithm accepts a **set of agents (A)**, **must-link** constraints (ML) and **cannot-link** constraints (CL).

1. Generation of ML and CL

Must-link and cannot-link constraints are represented as pairs of agents. During the coordination, each agent computes a set of individual must-link and individual cannot-link constraints set based on their preferences. To do so, they make use of and functions for individual must-link and individual cannot-link constraints, respectively.   
set is built by making use of and sets. (gender must-link) and (nationality must-link) correspond to must-link constraints for gender and nationality preferences of agent j ,respectively.

set is built by making use of and sets. (gender cannot-link) and (nationality cannot-link) correspond to cannot-link constraints for gender and nationality preferences of agent j ,respectively.

The **complete** must-link and cannot-link constraints sets are computed by **combining all individual** must-link and cannot-link sets of agents.

Individual must-link and cannot-link constraints for **Gender** Preference:

*:* Functioncomputes the must-link constraints of agent j for gender preference.

*:* Function computes the cannot-link constraints of agent j for gender preference.

**Gender linkage** function maps two agents either to their pair or to an empty set. The type of their pair is specified by . Assigning 1 (one) to will result in a must-link constraint and assigning -1 (negative one) to will result in a cannot-link. For example, given 1 to , if gender of agent satisfies the gender preference of agent , then will be the output of the function. However, given -1 (negative one) to , if there is a must-link in between , then the output will be empty set .

Individual must-link and cannot-link constraints for **Nationality** Preference:

*:* Functioncomputes the must-link constraints of agent j for nationality preference.

*:* Function computes the cannot-link constraints of agent j for nationality preference.

**Nationality linkage** function maps two agents either to their pair or to an empty set. The type of their pair is specified by . Assigning 1 (one) to will result in a must-link constraint and assigning -1 (negative one) to will result in a cannot-link. For example, given 1 to , if nationality of agent satisfies the nationality preference of agent , then will be the output of the function. However, given -1 (negative one) to , if there is a must-link in between , then the output will be empty set .

IML/ICL of agent j is the union of gender and nationality must-link/cannot-link constraints of agent j.

: Union of individual must-link constraints of all agents

: Union of individual cannot-link constraints of all agents

1. Calculation of k range and k

According to constraint **B.2**, two variables are defined: minimum () and maximum ( number of TCNs in a cluster. Since the cluster cardinality can be different, a range for the number of clusters (i.e. k) is calculated (. Since there is a range of K, the COP-Kmeans algorithm will be executed for each K value in the range.

1. Initialization of cluster centers

Initialize k cluster centers () where each of them takes a random value in between [0,100].

1. Assignment of agent j to cluster i

In order to assign an agent to a cluster, 2 course constraints (B.2 and B.4) must be satisfied as shown below. As well as, assignment must not violate any must-link or cannot-link constraint.

* 1. Each cluster has at least **two** and at most **five** agents (constraint B.2);

Where:

* 1. Each cluster should have at most **one** agent who **missed the previous** lesson (constraint B.4);

Where:

* 1. ML or CL violation

1. Update of cluster centers

The cluster centers need to be updated at each iteration of COP-Kmeans clustering algorithm. To do so, the following formula is used.

1. Convergence

Convergence is a point where the current cluster centers are same as the previous cluster centers. In COP-Kmeans clustering, the algorithm aborts and produces a result when it converges. Until convergence, it iterates between step 5 and step 6.

1. The output

Once the algorithm converges, it produces a set of clusters.

**COP-KMeans Pseudocode**

**INPUT**: Agent set A, Must-Link constraints , Cannot-link constraints

1. Set the minimum and maximum number for the cardinality of each cluster.
2. Calculate the minimum and maximum values for k. k represents the number of clusters.

1. **WHILE** K is not greater than
2. Let be the initial cluster centers
3. **REPEAT**
4. **FOR** each agent in
5. Assign it to the closest cluster center such that no constraint is violated (=false)
6. **IF** no such cluster exist **THEN**
7. Increment k by one and start again from (3).
8. **ENDIF**
9. **ENDFOR**
10. **ENDIF**
11. **FOR** each cluster in
12. Update cluster centers by averaging all of the points that have been assigned to them
13. **ENDFOR**
14. **UNTIL** convergence of clustering
15. **IF** converged **THEN**
16. **OUTPUT** the set of clusters (return {})
17. **ENDWHILE**
18. **OUTPUT** empty cluster set (return {})

**INPUT**:

**OUTPUT:** true or false

1. **FOR** each pair
2. **IF** is not in cluster **THEN**
3. **RETURN** true
4. **ENDIF**
5. **ENDFOR**
6. **FOR** each pair
7. **IF** is in cluster **THEN**
8. **RETURN** true
9. **ENDIF**
10. **ENDFOR**
11. **RETURN** false

**===========================================================================**

**LCC\_CKMeans Protocol**

1. **Start**
2. Teacher selects a set of TCNs (or a classroom) and initiates the coordination
3. WPM sends a signal to the agents of selected TCNs. The signal contains information about the list of agents and a dedicated agent who is being selected by WPM (randomly or by considering the resources)
4. Dedicated agent collects **personal info** (no preference) from other coordinating agents, builds the complete personal info and **broadcasts** it to the coordinating agents.
5. Upon receiving the complete info, each agent creates 2 sets: **must-link** constraints and **cannot-link** constraints based on its own preferences
6. Dedicated agent collects these pair of sets from each agent, **merges** them appropriately into 2 **complete** **must-link** and **complete cannot-link** constraints sets.
7. (Optional) Dedicated agent broadcasts the complete must-link and complete cannot-link constraints to the agents such that any of them would be able to solve the problem in case anything happens to the dedicated agent.
8. Dedicated agent **runs COP-Kmeans(**D, must-link, cannot-link**)** algorithm and gives the complete info, must-link, cannot-link constraints as the **input**.
9. Upon finding a solution, dedicated agent sends the **result to WPM** and **other agents**.
10. (Optional) Upon receiving the result, WPM might decide **to kill or not kill** the agents who were involved in the coordination.
11. Teacher selects from **rank list**, **modifies** or **approves** the grouping proposal of agents and then either a grouping result or an “*approved*” message is sent to all involved agents by WPM.
12. Each agent informs the TCN by sending the name of the group which the TCN belongs to. Regardless of the login state of the TCN, agent proceeds to the next step. Because once an agent receives the approved grouping from teacher, it stores the info in LAKR such that the TCN can request it anytime via the Welcome app.

Each agent sends a signal to WPM informing that the coordination is **ended**.

### SCOP-KMeans (Wagstaff K. L., 2002)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

## Recommendation Systems (Beel, Gipp, Langer, & Breit, 2015)

Formal definition from page 5 of (Palomares, Porcel, Pizzato, Guy, & Herrera-Viedma, 2021)

### Content Based

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### Collaborative Filtering

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### Group Recommendation-RS (Boratto, 2016)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

## Reciprocal Recommendation Systems (Palomares, Porcel, Pizzato, Guy, & Herrera-Viedma, 2021)

Formal definition from page 5 of (Palomares, Porcel, Pizzato, Guy, & Herrera-Viedma, 2021)

### CF-RRS

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### Hybrid-RRS

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### Group Formation-RRS (Yacef & McLaren, 2015)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

## Coalition Structure Generation Problem (Präntare & Heintz, 2020)

### ACS (Unknown, 2021)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### ODP-IP (Michalak, Rahwan, Elkind, Wooldridge, & Jennings, 2016)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

### ODSS (Changder, Aknine, Ramchurn, & Dutta, ODSS: Efficient Hybridization for Optimal Coalition Structure Generation, 2020)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

## Clustering Ensemble Algorithm

### ACE (Alqurashi & Wang, Clustering ensemble method, 2019)

**Definition**:

* explain briefly in your own words
* refer to the original paper for formal description of the algorithm

The **ACE** algorithm combines different clustering models to get a better clustering result by making use of the cluster similarities. It consists of 3 stages: *Transformation*, *Generating Consensus Clusters* and *Enforcing Hard Clustering*. [Figure 6](#ACE_diagram) illustrates the stages.

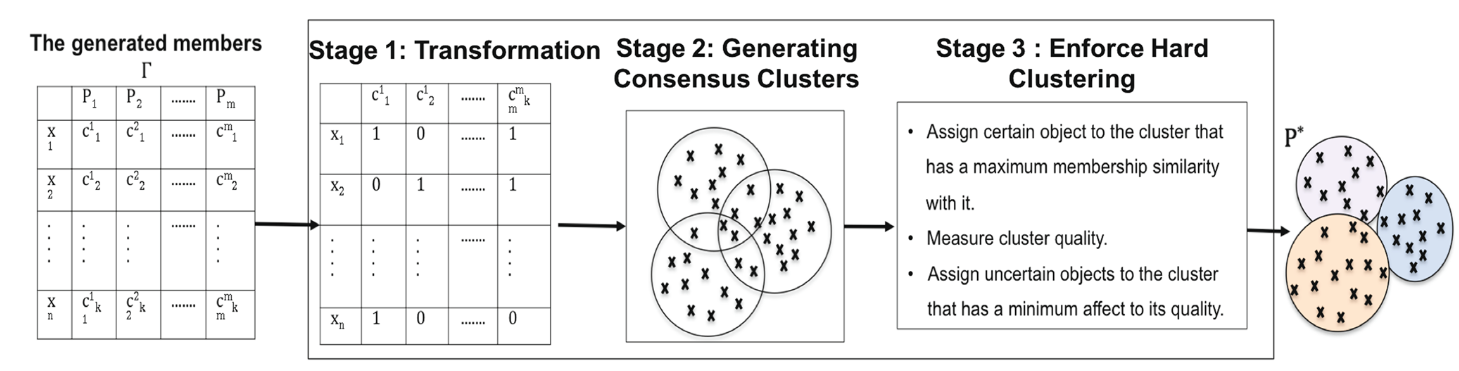
ACE requires a set of clustering models as the input. Therefore, multiple clustering models must be generated before running the algorithm. One simple method to generate a clustering is using k-means clustering. The input is a matrix, consisting of partitions as columns and objects as rows.

1. In *Transformation* stage, ACE transforms the input into a matrix such that columns are clusters and rows are objects. Each cell can take either 0 (non-member) or 1 (member), depending on the membership of an object to a cluster.
2. At *Generating Consensus Clusters* stage, the algorithm calculates the similarities between clusters, then merges the clusters which have a similarity value higher than a predefined threshold. Also it takes a *k* value as the number of clusters which should be the output. Briefly, this stage aims to merge the most similar clusters.
3. At *Enforcing Hard Clustering* stage, the algorithm assigns the objects to the clusters such that each cluster quality is affected as little as possible. Finally, it outputs a set of k clusters as the result.

The lines above briefly explain the ACE algorithm even though there are lots of details about each stage such as calculating cluster and membership similarities, selecting and updating thresholds and etc.

Shortly, the ACE algorithm takes a set of clustering, similarity thresholds and a value *k* as **input** and **outputs** a single clustering which contains *k* clusters.

The time complexity for the worst-case scenario of ACE algorithm is estimated to be equal to , where is the total number of clusters in all the generated members, and is the number of uncertain objects which is in the worst case scenario equal to , and k is the number of pre-defined clusters for the dataset. (Alqurashi & Wang, Clustering ensemble method, 2018)



**Figure 6**. Diagram of the ACE algorithm (Alqurashi & Wang, Clustering ensemble method, 2018)

**Implications**:

* Strongest argument whether to be ruled out or to be considered
* List the characteristics of algorithm
* Refer to the big characteristics table for comparison between algorithms

The objective of ensemble clustering is to maximize the quality of clustering by combining the existing clustering models. To reduce the LCC problem into an ensemble clustering problem, the suitability of a clustering can be interpreted as the quality of the clustering. However, in order to be able to ensemble clustering models, they need to be generated in the first hand. Since there is no existing clustering models in the LCC scenario, they need to be generated (e.g. with COP-Kmeans) and then they can be combined (e.g. with ACE). In the [CKMeans section](#CKMeans_section), it is indicated that LCC problem can be reduced into Constrained K-means clustering problem and can be solved. Therefore, even though LCC can be solved as ensemble clustering, it doesn’t make sense to implement it while CKMeans already solves the problem.

# References

Aknine, S., Ramchurn, S., Changder, N., & Dutta, A. (2021). BOSS: A Bi-directional Search Technique for Optimal Coalition Structure Generation with Minimal Overlapping, Student Abstract. *AAAI.*

Alqurashi, T., & Wang, W. (2018). Clustering ensemble method. *International Journal of Machine Learning and Cybernetics*.

Alqurashi, T., & Wang, W. (2019). Clustering ensemble method. *International Journal of Machine Learning and Cybernetics*, (pp. 1-18). doi:10.1007/s13042-017-0756-7

Ayeni, A. W. (2014). Empirics of Standard Deviation. *Research Presentation, Covenant University*.

Beel, J., Gipp, B., Langer, S., & Breit. (2015). Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, 1-34. From https://www.researchgate.net/publication/280576923\_Research-paper\_recommender\_systems\_A\_literature\_survey

Boratto, L. (2016). Group Recommender Systems: State of the Art, Emerging Aspects and Techniques, and Research Challenges. *ECIR 2016: Advances in Information Retrieval.* From https://link.springer.com/chapter/10.1007/978-3-319-30671-1\_87#citeas

Borgelt, C. (2020). Even Faster Exact k-Means Clustering. *IDA 2020: Advances in Intelligent Data Analysis XVIII.* From https://link.springer.com/chapter/10.1007/978-3-030-44584-3\_8

Campello, R. J., Moulavi, D., Zimek, A., & Sander, J. (2015). Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Transactions on Knowledge Discovery from Data.* Association for Computing Machinery. From https://doi.org/10.1145/2733381

Changder, N., Aknine, S., Ramchurn, S., & Dutta, A. (2020). ODSS: Efficient Hybridization for Optimal Coalition Structure Generation. *Proceedings of the AAAI Conference on Artificial Intelligence, 34(05)*, 7079-7086. From https://doi.org/10.1609/aaai.v34i05.6194

Changder, N., Aknine, S., Ramchurn, S., & Dutta, A. (2021). BOSS: A Bi-directional Search Technique for Optimal Coalition Structure Generation with Minimal Overlapping, Student Abstract. *AAAI.* SMA - Systèmes Multi-Agents, University of Southampton .

Cornuéjols, A., Wemmert, C., Gançarski, P., & Bennani, Y. (2018). Collaborative clustering: Why, when, what and how. *Information Fusion.* MIA-Paris, ICube, LIPN.

Ertl, B., Meyer, J., Schneider, M., & Streit, A. (2020). CoExDBSCAN: Density-based Clustering with Constrained Expansion. *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR*, 104-115.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, (pp. 226-231).

Fioretto, F., Pontelli, E., & Yeoh, W. (2018). Distributed Constraint Optimization Problems and Applications: A Survey. *Journal of Artificial Intelligence Research*.

Manning, C., Raghavan, P., & Schü. (2008). *Introduction to information retrieval.* Cambridge University Press. From https://www.cambridge.org/core/books/introduction-to-information-retrieval/669D108D20F556C5C30957D63B5AB65C

Masayuki, O., & Seiji, Y. (2018). Clustering Using Boosted Constrained k-Means Algorithm. *Frontiers in Robotics and AI*.

Michalak, T., Rahwan, T., Elkind, E., Wooldridge, M., & Jennings, N. R. (2016). A hybrid exact algorithm for complete set partitioning. *Artificial Intelligence 230.* From https://dl.acm.org/doi/abs/10.1016/j.artint.2015.09.006

Nielsen, F. (2016). Hierarchical Clustering. *Introduction to HPC with MPI for Data Science*, (p. chapter 8). From https://www.researchgate.net/publication/314700681\_Hierarchical\_Clustering

Okabe, M., & Yamada, S. (2018). Clustering Using Boosted Constrained k-Means Algorithm. *Frontiers in Robotics and AI.* From https://www.researchgate.net/publication/323637580\_Clustering\_Using\_Boosted\_Constrained\_k-Means\_Algorithm

Palomares, I., Porcel, C., Pizzato, L., Guy, I., & Herrera-Viedma, E. (2021). Reciprocal Recommender Systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation. *Information Fusion*, (pp. 103-127). From https://doi.org/10.1016/j.inffus.2020.12.001

Präntare, F., & Heintz, F. (2020). An anytime algorithm for optimal simultaneous coalition structure generation and assignment. *Autonomous Agents and Multi-Agent Systems 34.* Linkoping University.

Rahwan, T., P. Michalak, T., Elkind, E., Wooldridge, M., & R. Jennings, N. (2013). An Exact Algorithm for Coalition Structure Generation and Complete Set Partitioning. DCS.

Taguelmimt, R., Aknine, S., Boukredera, D., & Changder, N. (2021). FACS: Fast code-based Algorithm for Coalition Structure Generation. *AAAI.* SMA - Systèmes Multi-Agents .

Unknown, U. (2021). Code-based Algorithm for Coalition Structure Generation. *Unknown Journal*.

Wagstaff, K. L. (2002). Intelligent Clustering with Instance-Level Constraints. *PhD thesis,Department of Computer Science, Cornell University*.

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained K-means Clustering with Background Knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning.* Williams College.

Yacef, K., & McLaren, B. (2015). Supporting Learners ’ Group Formation with Reciprocal Recommender Technology. *In 3rd Workshop on Intelligent Support for Learning in Groups, held in conjunction with ITS.* From https://www.semanticscholar.org/paper/Supporting-Learners-%E2%80%99-Group-Formation-with-Yacef-McLaren/ea9d694829841d50c8dbf3c37fbe9f0b4fc883bf