Table of Contents

[1. LCC Informal Description 2](#_Toc72425068)

[2. LCC Formal Description 3](#_Toc72425069)

[2.1. BOSS - Coalition Structure Generation Problem Solver (Aknine, Ramchurn, Changder, & Dutta, 2021) 4](#_Toc72425070)

[2.1.1. Definition 4](#_Toc72425071)

[2.1.2. Implications 5](#_Toc72425072)

[2.1.3. Projection 6](#_Toc72425073)

[2.1.4. Coordination Protocol 10](#_Toc72425074)

[2.2. BCOP-Kmeans - Constrained Clustering Problem Solver (Okabe & Yamada, 2018) 10](#_Toc72425075)

[2.2.1. Definition 10](#_Toc72425076)

[2.2.2. Implications 11](#_Toc72425077)

[2.2.3. Projection 11](#_Toc72425078)

[2.2.4. Coordination Protocol 11](#_Toc72425079)

[3. Analyzed Algorithms 11](#_Toc72425080)

[3.1. Constrained Optimization Problems 11](#_Toc72425081)

[3.1.1. COP (Fioretto, Pontelli, & Yeoh, 2018) 11](#_Toc72425082)

[3.1.2. DCOP (Fioretto, Pontelli, & Yeoh, 2018) 12](#_Toc72425083)

[3.1.3. ADCOP (Fioretto, Pontelli, & Yeoh, 2018) 13](#_Toc72425084)

[3.1.4. MODCOP (Fioretto, Pontelli, & Yeoh, 2018) 14](#_Toc72425085)

[3.2. Clustering Algorithms 15](#_Toc72425086)

[3.2.1. K-Means (Manning, Raghavan, & Schü, 2008) 15](#_Toc72425087)

[3.2.2. Faster Exact K-Means (Borgelt, 2020) 15](#_Toc72425088)

[3.2.3. Hierarchical Clustering (Nielsen, 2016) 15](#_Toc72425089)

[3.2.4. DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) 15](#_Toc72425090)

[3.3. Constrained Clustering Algorithms 15](#_Toc72425091)

[3.3.1. HDBSCAN\* (Campello, Moulavi, Zimek, & Sander, 2015) 15](#_Toc72425092)

[3.3.2. CoExDBSCAN (Ertl, Meyer, Schneider, & Streit, 2020) 16](#_Toc72425093)

[3.3.3. COP-KMeans (Wagstaff, Cardie, Rogers, & Schrödl, 2001) 16](#_Toc72425094)

[1.1.1. SCOP-KMeans (Wagstaff K. L., 2002) 23](#_Toc72425095)

[1.2. Recommendation Systems (Beel, Gipp, Langer, & Breit, 2015) 23](#_Toc72425096)

[1.2.1. Content Based 23](#_Toc72425097)

[1.2.2. Collaborative Filtering 23](#_Toc72425098)

[1.2.3. Group Recommendation-RS (Boratto, 2016) 23](#_Toc72425099)

[1.3. Reciprocal Recommendation Systems (Palomares, Porcel, Pizzato, Guy, & Herrera-Viedma, 2021) 24](#_Toc72425100)

[1.3.1. CB-RRS 24](#_Toc72425101)

[1.3.2. CF-RRS 24](#_Toc72425102)

[1.3.3. Hybrid-RRS 24](#_Toc72425103)

[1.3.4. Group Formation-RRS (Yacef & McLaren, 2015) 24](#_Toc72425104)

[1.4. Coalition Structure Generation Problem (Präntare & Heintz, 2020) 24](#_Toc72425105)

[1.4.1. ACS (Unknown, 2021) 24](#_Toc72425106)

[1.4.2. ODP-IP (Michalak, Rahwan, Elkind, Wooldridge, & Jennings, 2016) 25](#_Toc72425107)

[1.4.3. ODSS (Changder, Aknine, Ramchurn, & Dutta, ODSS: Efficient Hybridization for Optimal Coalition Structure Generation, 2020) 25](#_Toc72425108)

[1.5. Clustering Ensemble Algorithm 25](#_Toc72425109)

[1.5.1. ACE (Alqurashi & Wang, Clustering ensemble method, 2019) 25](#_Toc72425110)

[References 27](#_Toc72425111)

# LCC Informal Description

One of the main challenges in the Catalan language course of the First Reception Service is meeting the different learning speeds of the students (TCNs). To do that, the teachers usually divide the class into smaller **working groups**. A course consists of multiple consecutive lessons (classes), one per day in the course period.

In WELCOME, the language course coordination (LCC) scenario refers to a language course of, **initially**, **three** consecutive lessons (or classes). In each lesson, every student performs exercises of four types of Catalan language learning activities related to the **reading**, **writing**, **grammar**, and **vocabulary** of Catalan. Each student is assigned to one working group per lesson. **At the end of each lesson**, the progress made by the students is individually assessed through means of a specific set of language learning assessment exercises for each of the four types of learning activities. Language **speaking activities** are not considered in WELCOME. The grouping of students per lesson is usually made by the teacher upon various factors, both **course** and **individual constraints**, and depending on the overall **course progress level** of individual students.

In this context, to facilitate the intervention of the teacher, it would be very helpful if the developed WELCOME technology can support the teacher in her language course coordination. The **objective** of the agents of the students would be to coordinate among each other for finding an (approximately) optimal assignment of students to groups with similar level of course progress. In particular, the WELCOME platform should allow to assess the progress of each of the students and make grouping suggestions to the teacher at the end of each lesson based on the following constraints:

1. **Individual constraints** specified by the students (TCNs)
2. **Gender**: Gender can be a major constraint for the participation of women in the language learning course. Although this limitation can be considered less important in large groups (such as a class), when working in smaller groups, female TCNs, who do not want to interact with the other gender, are less likely to participate freely in the exercises of the lessons if they are grouped into a mixed-gender group. This can also happen to male-TCNs, although it is less likely to happen.

Therefore, the agent should respect the preference of its TCN about being assigned to a mixed-gender group or not. In fact, this individual constraint **must be satisfied**.

1. **Nationality**. Although many people do not have a problem when working with other nationalities, some of the TCNs might have. To ensure the progress of all members of a group in a class, the agent should respect the preference of its TCN about being assigned to a group with other nationalities than her own one. This constraint, however, **may be ignored** for sake of finding an overall grouping solution.

Each student (TCN) can select one of 3 following options for each preference (gender and nationality):

1) *Same*: TCN prefers a group in which the gender of all others are same as his/hers.

2) *Mixed:* TCN prefers a group in which there are at least a male and a female.

3) *Don’t mind*: TCN doesn’t care about the gender of others in the group.

1. **Course constraints** to be satisfied by the coordinating agents of the TCNs
   1. Each class (lesson) has at least **two** and at most **twenty** students
   2. Each group in a lesson has at least **two** and at most **five** students
   3. Each group has students with **similar** course progress level
   4. Each group should have at most **one** student who **missed the preceding** lesson. A group can have **more than one** student (who missed the previous lesson) **only if** there are more students who missed the previous lesson than the number of groups.

The **progress assessment scheme** to be used by the agents for their finding of an optimal grouping of students is as follows:

* The language learning skill of a student is measured at the end of each lesson of the course as the equally weighted average sum of percentages of correct answers for each language learning activity (reading, writing, grammar, vocabulary) given by the student in the lesson.
  + The *language learning assessment* score (LLA-X) for a given type X of language learning activity (X in {Reading, Writing, Grammar, Vocabulary}) denotes the percentage of correct answers for assessment exercises of this type by the student. An LLA-X score in the interval of 0-49, 50-59, 60-69, 70-79, 80-89, 90-100 is interpreted as “insufficient”, “sufficient”, “normal”, “good”, “remarkable”, “outstanding” progress, respectively. Besides, the time a student spent to complete the assessment exercises of a given type of language learning activity in the interval of 0-5, 5-10, 10+ minutes is interpreted as „Student is doing OK“, „Student is facing difficulties“ and „Student is struggling“, respectively, in this regard.
  + The *language lesson* (LL) score of an individual student is defined as the equally weighted sum of the LLA-X scores she obtained for performing assessment exercises for the selected four types of language learning activities in multiple-choice form.

The *course progress level* (CPL) score for a student is determined as the averaged sum of the LL scores for all lessons she attended so far.

# LCC Formal Description

In this section, different kind of problem classes are described to compare with LCC problem and to identify which one fits best.

In the Informal Description section of LCC scenario, it is given that the objective of agents is to coordinate among each other for finding an (approximately) optimal assignment of students with similar course progress levels. As well as, the individual and course constraints should be taken into account while making the assignments. For the formal description of LCC problem, the objective is not explicitly given as the maximization or minimization of a specific value. Besides, language instructor hasn’t been requested to identify/group TCNs who perform best in the class. If so, the objective of LCC would be to maximize the aggregation of course progress level of TCNs in a working group. However, the overall goal is to satisfy the constraints while forming the working groups of TCNs in order to provide a more suitable environment (working groups) which are aimed to help them to improve their language learning performance. Hence, the formal objective of LCC problem can be described as to **maximize** the degree of **suitability** of working groups. Since the formal objective is given implicitly, there isn’t any given specific valuation of the degree of suitability of a working group in the informal description section. Therefore, we can define the degree of suitability as the degree of **satisfaction** of constraints (both individual and course constraints).

Furthermore, the information given in this paragraph is agreed with User Partner such that informal terms can be converted into mathematical terms. The course constraints should never be violated. However, in case there are more than 10 TCNs who missed the previous lesson, it will be allowed to have at most 2 TCNs who missed the previous lesson in a group. Because, otherwise there will be no solution. Regarding course constraint B.3, since there are no explicit boundaries (i.e. **similar CPL** doesn’t have a meaning in mathematical terms), B.3 will comprise the 70% of the overall score of a grouping while satisfaction of individual preferences will comprise 30%. Regarding the individual preferences, the gender preference will comprise the 75% of the overall satisfaction of preferences while nationality will comprise only 25% since it is much less important than gender.

## BOSS - Coalition Structure Generation Problem Solver (Aknine, Ramchurn, Changder, & Dutta, 2021)

### Definition

\*\* Temporary note\*\*

\*\*Detailed information in terms of complexities, characteristics of BOSS 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 CSGP?
* How BOSS solves it? How BOSS works, briefly?
* Complexities, characteristics, etc. of BOSS

Coalition Structure Generation (CSG) problem aims to create agent teams to solve problems where single agents cannot solve effectively or cannot solve at all. In a CSG problem, a coalition is a set/group of agents. Each coalition has a value called coalition value which is the calculated based on the utility values of member agents. Utility value is the evaluation of the coalition from the agent perspective. A coalition value represents the potential performance of the coalition. A set of coalitions form a coalition structure (CS). Each CS has a value which is calculated based on the values of member coalitions. So the CSG problem is maximizing the coalition value (performance) and consequently the CS value such that agents/members are grouped in a way which maximizes their performance.

In order to solve a CSG problem, firstly algorithms produce all possible coalition structures where each of them contains a set of coalitions. Then a reward value for each coalition is calculated. Afterwards, a value for each CS is calculated by summing the coalition values. Finally, since each CS has a value, the algorithm can compare them and produce a rank list of CS from best to worst.

There are exact and approximate CSG solving algorithms and they grow exponentially. For example,   
**IP**- O(nn), (Rahwan, P. Michalak, Elkind, Wooldridge, & R. Jennings, 2013)  
**ODP** - O(3n), (Rahwan, P. Michalak, Elkind, Wooldridge, & R. Jennings, 2013)  
**ODP**-**IP** - O(3n), (Rahwan, P. Michalak, Elkind, Wooldridge, & R. Jennings, 2013)  
**BOSS** - O(3n), (Changder, Aknine, Ramchurn, & Dutta, BOSS: A Bi-directional Search Technique for Optimal Coalition Structure Generation with Minimal Overlapping, Student Abstract, 2021)  
**FACS** - O(3n). (Taguelmimt, Aknine, Boukredera, & Changder, 2021)

**BOSS** is an exact algorithm and according to authors it finds an exact solution for 27 agents in approximately 111 minutes. (Changder, Aknine, Ramchurn, & Dutta, BOSS: A Bi-directional Search Technique for Optimal Coalition Structure Generation with Minimal Overlapping, Student Abstract, 2021)

**FACS** is an approximate algorithm which is much faster than BOSS. (Taguelmimt, Aknine, Boukredera, & Changder, 2021)

### Implications

* How LCC is reduced into CSGP?
* Why BOSS can solve it? And why this solution is acceptable?
* Why BOSS but not ODSS or FACS?

Regarding the reduction of LCC into CSG problem, a working group in LCC problem corresponds to a coalition in CSGP. While CSGP maximizes the performance of a coalition, LCC problem maximizes the suitability of a group. A simple reduction from LCC problem to CSGP would imply maximizing the sum of course progress level of member TCNs. However, it would be unknown for the actual session. Besides, a specific way of valuation of a working group wasn’t given by the language instructor. Only that the general conditions and individual preferences should be satisfied for each working group. Therefore, when reducing LCC problem into a CSG problem, the coalition value is interpreted as the degree of suitability rather than the performance of the coalition/working group. Consequently, in LCC\_CSGP it is aimed to maximize the degree of suitability of each coalition by taking into account the system and individual constraints.

Regarding the runtime complexity of the CSGP solving algorithms, they are all exponential as described in the section above. However, since BOSS and FACS algorithms run successfully for maximum 27 agents and in LCC scenario there will be 2 to 20 TCNs in a classroom, the algorithms would be able to solve the LCC\_CSGP. Furthermore, there will be at least a day between 2 consecutive lessons in LCC scenario which is more than enough for BOSS and FACS algorithms to find a solution.

To be fair or equallly distributed CV,

the utlity function will have same max and min for all agents. if all prefernces satisfied, then max, if zero satisfied hen min.

In the formalization, mention about this. the utility increases linear to the amount of satisfied rpeferences. however, satisfying n preference of agent\_1 and satisfiying m preference of agent\_2, where n>m, doesn't mean that u(agent\_1) must be higher than u(agent\_2). Because each agent weights the preferences differently. A single preference might be extremely important for an agent which would mean a much higher utility (profit) for that agent.

How an agent weights its prefs is up the TCN. How the tcn gives weight? don't know yet.

### Projection

Projection of LCC problem as CSGP is divided into the four parts: a) *Coalition Structure Generation*, b) *Filtering*, c) *Evaluation* and d) *Output of results*.

1. **Coalition Structure Generation**: Firstly, given the number of the TCNs, we are able to evaluate the possible CSs. Note that according to the Course Constraints, the number of TCNs should be between **two** and **twenty (constraint B.1)**.
2. **Filtering**: The Coalition Structures should follow the requirements of the Course Constraints in terms of the group size restriction and maximum number of TCNs missed the previous lesson. Hence each of the generated Coalition Structure from the previous step will be filtered by the following criteria:
3. Each Coalition has at least **two** and at most **five** TCNs;

Where:

1. Each Coalition should have at most **one** TCN who **missed the previous** lesson unless the number of TCNs who missed the previous lesson is not more than the maximum number of groups. In other words, if there is no possible solution in which each group has at most 1 TCN who missed the previous lesson, then a group can have at most 2 TCNs who missed the previous lesson for the sake of having a solution.

Where:

1. **Evaluation:** This step aims to evaluate how well a certain Coalition Structure fits the LCC scenario. Namely it assesses the fitness of Individual Constraints and Course Constraints. The formal part evaluates how well each of the agent can be satisfied in the Coalition Structure and the later part evaluates how similar the course progress levels of the TCNs in the same Coalition.

In order to appropriately model the evaluation, we design the following functions.

1. **Utility value function**

The Utility value of an agent for a coalition reflects the satisfaction rate of the agent’s individual preferences. In other words, it measures the degree of how well a coalition fits to the agent. In LCC scenario, each TCN has gender and nationality preferences. To calculate the utility value, each agent independently weights its preferences and the sum of the weights equals to 1 such that all agents contribute fairly to the coalition value. While a satisfaction of a preference will increase, a violation will reduce the utility value by the amount of weight that is defined by the agent itself. Therefore, utility value ranges between -1 and 1. In order to identify whether a preference is satisfied or violated, the agent compares its preference with the corresponding information of coalition. For example, if a coalition contains TCN1 (male), TCN2 (female) and TCN3 (female), then the gender information of the coalition will be “mixed”. And if TCN3 prefers a same gender group, then the gender preference of agent3 will be violated. Since personal information (e.g. gender, nationality) of agents are shared while the individual preferences are not, the coalition information will be built by making use of the personal information of the member agents.

1. **Coalition value function**

Coalition value function calculates the suitability of a coalition. For the LCC scenario, it is designed as the sum of utility values and . According to the Formal Description, will comprise the 70% of Coalition Value while sum of utility values will comprise the rest.

Since each utility value represents the satisfaction rate of a member agent, summing all utility values will give the suitability of the coalition.

One of the course constraints is to have similar course progress level of TCNs in the same group. In order to calculate the closeness of the CPL of members in a coalition, we use standard deviation. Standard deviation is the most popular method to measure the spread in the data. (Ayeni, 2014)

1. **Coalition Structure Value Function**

Coalition Structure Value function is the overall evaluation value of a certain Coalition Structure. It is the sum of all the Coalition Values within the Coalition Structure. With the foundation of the previous functions, the Coalition Structure Value function assesses the quality of each Coalition Structure by taking into the consideration of both how well the individual preferences of each TCN in the same Coalition are satisfied, and how similar the course progress levels of the TCNs in the same Coalition.

We can directly compare the qualities of several Coalition Structures by comparing their corresponding Coalition Structure Values. A better Coalition Structure has a higher Coalition Structure Value. To find the best Coalition Structure result, we need to find the Coalition Structure with the maximum Coalition Structure Value.

**Example**

An example is shown below. The Coalition Structure consists of two Coalitions, each contains three TCNs:

Coalition 1:

|  |  |  |  |
| --- | --- | --- | --- |
| **TCN number** | **TCN1** | **TCN2** | **TCN3** |
| **Gender preference** | Same gender | Same gender | Different gender |
| **Nationality preference** | Same nationality | Different nationality | Different nationality |
| **Gender** | Male | Female | Male |
| **Nationality** | A | A | B |
| **Language Lesson Score** | 85 | 65 | 75 |

Coalition 2:

|  |  |  |  |
| --- | --- | --- | --- |
| **TCN number** | **TCN4** | **TCN5** | **TCN6** |
| **Gender preference** | Different gender | Same gender | Same gender |
| **Nationality preference** | Different nationality | Different nationality | Same nationality |
| **Gender** | Female | Female | Female |
| **Nationality** | A | B | C |
| **Language Lesson Score** | 80 | 81 | 85 |

In coalition 1 (C1), there are 3 TCNs. Two TCNs in C1 are male while TCN2 is female. Therefore, the gender information of C1 is mixed. Similarly, the nationality info of C1 is mixed. In C1, both preferences of TCN3, 1 preference of TCN2 and no preference of TCN1 is satisfied.

Let’s assume that the weights for 3 agents are as following:

Accordingly, the utility values of 3 TCNs would be as following:

Accordingly, coalition value and SD of C1 would be as following:

In coalition 2 (C2), there are 3 TCNs. All TCNs in C2 are female. Therefore, the gender information of C2 is same (female). Similarly, the nationality info of C2 is mixed. In C2, both preferences of TCN5, 1 preference of TCN6 and 1 preference of TCN4 are satisfied.

Let’s assume that the weights for 3 agents are as following:

Accordingly, the utility values of 3 TCNs would be as following:

Accordingly, coalition value and SD of C2 would be as following:

Since coalition values are calculated, the coalition structure value would be:

**Output Results**

As the last step, we calculate the Coalition Structure Value for each generated CS and rank them based on their values (from highest to lowest). Then we select top t CS and output it as the result.

### Coordination Protocol

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

## BCOP-Kmeans - Constrained Clustering Problem Solver (Okabe & Yamada, 2018)

### Definition

\*\* Temporary note\*\*

\*\*Detailed information in terms of complexities, characteristics of BCOP-Kmeans 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)
* For other constrained clustering algorithms check section x.x
* explain the working principle of Bcop-kmeans, briefly in your own words
* refer to the original paper for more details
* Mention the complexities and other characteristics

### Implications

* Reduction implications: how LCC is reduced into constrained clustering?
* Advantages of BCOP-Kmeans
* It is selected because of the following reasons…
* It is extended and the pseudocode of LCC\_BCOP-kmeans can be checked in Appendix y.

### Projection

* The objective function and constraints (ML&CL and system) in the objective function.

### 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)

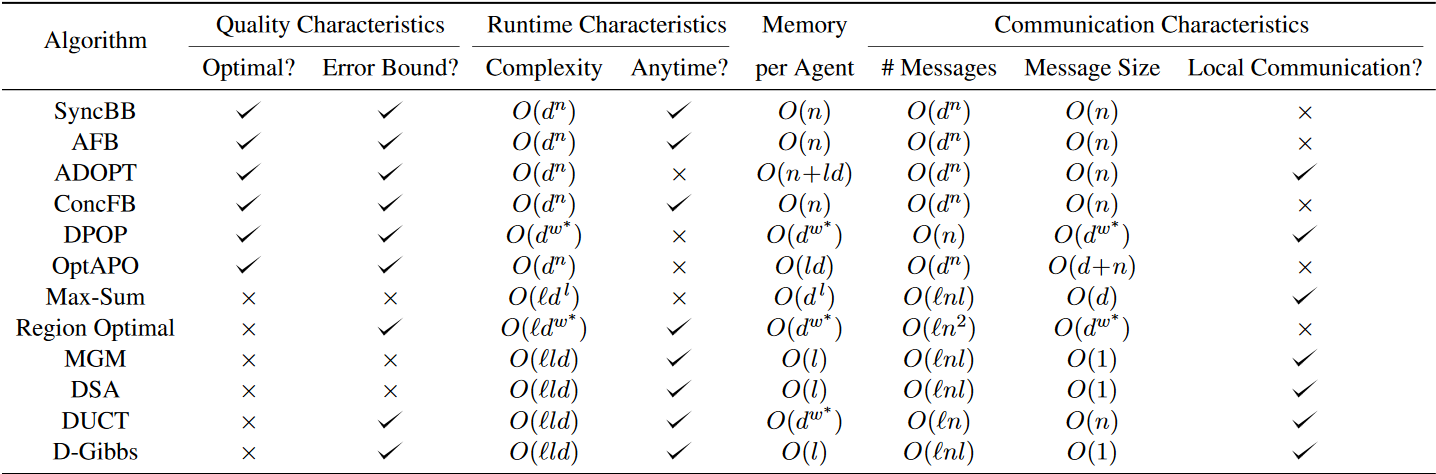
**Definition**:

* 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

### HDBSCAN\* (Campello, Moulavi, Zimek, & Sander, 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

### 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

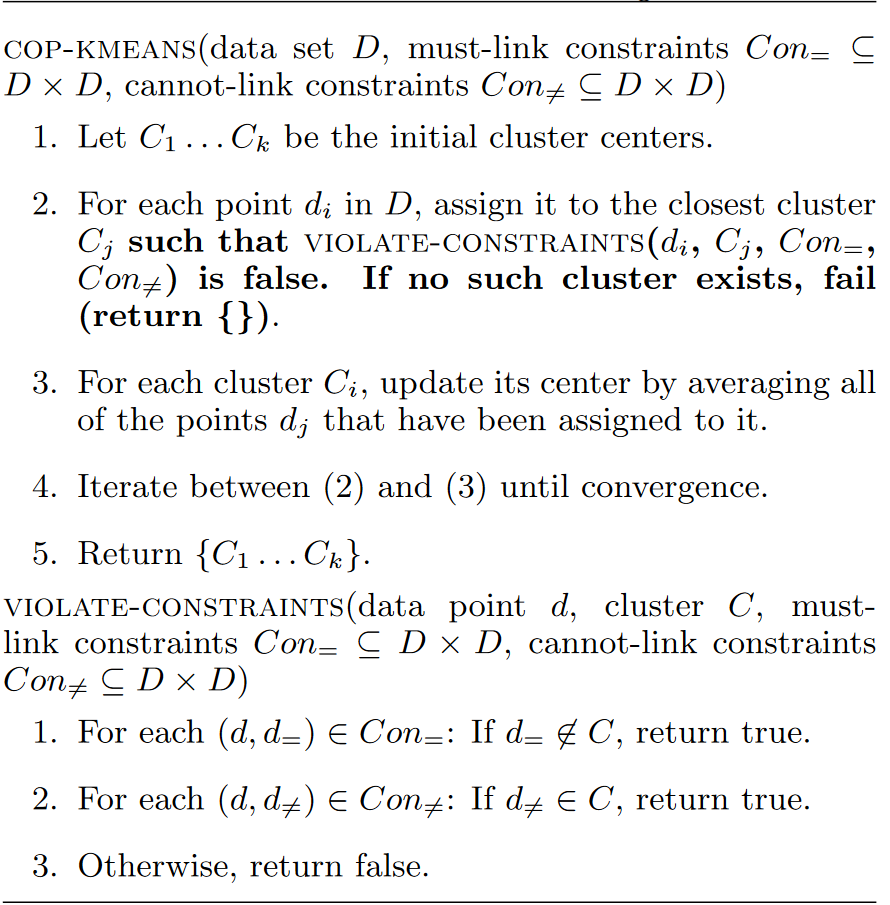
Clustering is an unsupervised method for data analysis. In clustering, the data points are grouped based on a similarity unit. The **objective** of clustering is to group a set of objects in such a way that the objects are similar intra-group and dissimilar inter-group as much as possible such that distinct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

### 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)

### CB-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

### 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.

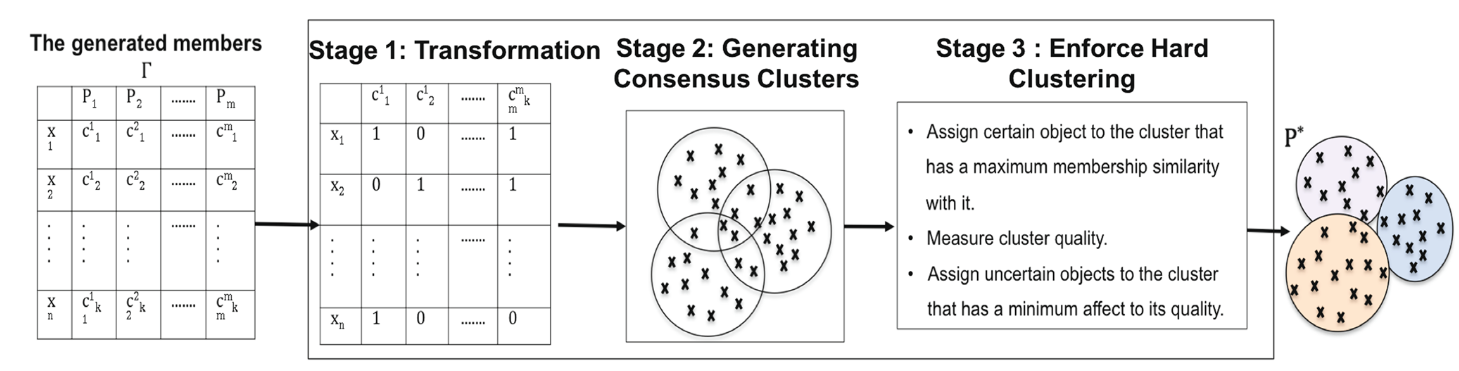
ACE requires a set of clustering models as the input. Therefore, multiple clustering models must be generated before running the algorithm. One simple method to generate a clustering is using k-means clustering. The input is a matrix, consisting of partitions as columns and objects as rows.

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