NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

POSTGRADUATE PROGRAMME
"INFORMATION AND KNOWLEDGE DATA MANAGEMENT"

MASTER THESIS

Clustering in Recommendation Systems Using Swarm Intelligence

Maria-Mirto G. Koliopoulou

Supervisor: **Panagiotis Stamatopoulos,** Assistant Professor

ATHENS

SEPTEMBER 2020

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
"ΔΙΑΧΕΙΡΙΣΗ ΔΕΔΟΜΕΝΩΝ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΓΝΩΣΗΣ"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ομαδοποίηση στα Συστήματα Συστάσεων μέσω Ευφυΐας Σμήνους

Μαρία-Μυρτώ Γ. Κολιοπούλου

Επιβλέπων: **Παναγιώτης Σταματόπουλος,** Επίκουρος Καθηγητής

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2020

MASTER THESIS

Clustering in Recommendation Systems Using Swarm Intelligence

Maria-Mirto G. Koliopoulou
S.N.: CS2180021

SUPERVISOR: **Panagiotis Stamatopoulos,** Assistant Professor

ADVISORY COMMITTEE: **Aphrodite Tsalgatidou,** Associate Professor
Isambo Karali, Assistant Professor

September 2020

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ομαδοποίηση στα Συστήματα Συστάσεων μέσω Ευφυΐας Σμήνους

Μαρία-Μυρτώ Γ. Κολιοπούλου
Α.Μ.: CS2180021

ΕΠΙΒΛΕΠΩΝ: **Παναγιώτης Σταματόπουλος,** Επίκουρος Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Αφροδίτη Τσαλγατίδου,** Αναπληρώτρια Καθηγήτρια
Ιζαμπώ Καράλη, Επίκουρη Καθηγήτρια

Σεπτέμβριος 2020

ABSTRACT

A recommender system (RS) is an application that exploits information to help users in decision making by suggesting items they might like. A collaborative recommender system generates recommendations to users based on their similar neighbor's preferences. However, this type of recommender system faces the data sparsity and scalability problems making the neighborhood selection a challenging task. This thesis proposes three hybrid collaborative recommender systems that each one combines the k-means algorithm with a different bio-inspired technique to enhance the clustering task, and therefore to improve the recommendation quality. The used bio-inspired techniques are artificial bee colony (ABC), cuckoo search optimization (CSO), and grey-wolf optimizer (GWO). The proposed approaches were evaluated over a MovieLens dataset. The evaluation shows that the proposed recommender systems perform better compared to already existing techniques in terms of mean absolute error (MAE), precision, sum of squared errors (SSE), and recall. Moreover, the experimental results indicate that the hybrid recommender system that uses the ABC method performs slightly better than the other two proposed hybrid algorithms.

SUBJECT AREA: Recommender Systems

KEYWORDS: clustering, swarm intelligence, collaborative filtering, k-means, artificial bee colony, recommender systems

ΠΕΡΙΛΗΨΗ

Ένα σύστημα συστάσεων είναι μία εφαρμογή που εκμεταλλεύεται πληροφορίες για να βοηθήσει τους χρήστες στη λήψη αποφάσεων προτείνοντας αντικείμενα που μπορεί να τους αρέσουν. Ένα σύστημα συστάσεων που βασίζεται στην τεχνική του συνεργατικού φιλτραρίσματος (collaborative filtering) δημιουργεί συστάσεις στους χρήστες με βάση τις προτιμήσεις παρόμοιων χρηστών. Ωστόσο, αυτός ο τύπος συστήματος συστάσεων δεν είναι τόσο αποτελεσματικός όταν τα δεδομένα αυξάνονται σε μεγάλο βαθμό (scalability) ή όταν δεν υπάρχει αρκετή πληροφορία (sparsity), καθώς δεν ομαδοποιούνται σωστά οι παρόμοιοι χρήστες. Αυτή η διπλωματική εργασία προτείνει τρείς υβριδικούς αλγορίθμους που ο καθένας συνδυάζει τον αλγόριθμο k-means με έναν αλγόριθμο ευφυΐας σμήνους για να βελτιώσει την ομαδοποίηση των χρηστών, και κατ' επέκταση την ποιότητα των συστάσεων. Οι αλγόριθμοι ευφυΐας σμήνους που χρησιμοποιούνται είναι ο αλγόριθμος τεχνητής κοινωνίας μελισσών (artificial bee colony), ο αλγόριθμος βελτιστοποίησης αναζήτησης κούκων (cuckoo search optimization) και ο αλγόριθμος βελτιστοποίησης γκρίζων λύκων (grey-wolf optimization). Οι προτεινόμενες μέθοδοι αξιολογήθηκαν χρησιμοποιώντας ένα σύνολο δεδομένων του MovieLens. Η αξιολόγηση δείχνει πως τα προτεινόμενα συστήματα συστάσεων αποδίδουν καλύτερα σε σύγκριση με τις ήδη υπάρχουσες τεχνικές όσον αφορά τις μετρικές του μέσου απόλυτου σφάλματος (mean absolute error - MAE), της ακρίβειας (precision), του αθροίσματος των τετραγωνικών σφαλμάτων (sum of squared errors - SSE) και της ανάκλησης (recall). Επιπλέον, τα αποτελέσματα της αξιολόγησης δείχνουν πως ο υβριδικός αλγόριθμος που χρησιμοποιεί την μέθοδο της τεχνητής κοινωνίας μελισσών αποδίδει ελαφρώς καλύτερα από τους άλλους δύο προτεινόμενους αλγορίθμους.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα συστάσεων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ομαδοποίηση, ευφυΐα σμήνους, συνεργατικό φιλτράρισμα, k-means, τεχνητή κοινωνία μελισσών, συστήματα συστάσεων

To my family

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Panagiotis Stamatopoulos, for the chance he gave me to work on this project, and for his continuous support and guidance in the preparation and completion of this thesis.

CONTENTS

LIST OF FIGURES

LIST OF IMAGES

LIST OF TABLES

PROLOGUE

This thesis has been conducted for the MSc degree in Computer Science, offered by the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens.

1. INTRODUCTION

In this age of information overload, people develop various strategies to make choices about what to buy, what to read, and even whom to date. Machine learning (ML) tries to help people in decision making by automating these strategies. The ML algorithms that accomplish the above are called Recommender systems (RSs). Recommender systems try to narrow down choices for people by finding and therefore suggesting relevant items to users. Such items could be movies to watch, text to read, products to buy, music to listen to, etc. Recommendation systems affect more and more our lives, considering that various daily applications use their recommendation engines to add value to users by helping them discover new content. Some examples of recommendations in our everyday lives are Amazon, LinkedIn, Netflix, and Facebook. Amazon uses data from its customers to identify which items are usually bought together and makes recommendations based on that. LinkedIn uses the experience, the job titles, and generally the user profile to suggest suitable jobs to the user. Netflix exploits its rating system to find similar subscribers, and therefore to recommend movies and shows to them. Finally, Facebook, as a social network application, does not directly recommend products but connections. Recommendation systems use different techniques to achieve their goal. Content-based (CB) and Collaborative filtering (CF) are the two most well-known recommendation techniques. CB based recommender systems look for similarities between the items or products that a person had bought or liked in the past to recommend options in the future. Collaborative filtering generates recommendations by using the information provided by many users, which is people’s collaborative behavior. CF faces two main challenges, data sparsity (huge number of missing values in the data set) and scalability (increasing number of users and number of items). In this thesis, three hybrid CF-based recommender systems are proposed to improve the sparsity and scalability issues. Each proposed system combines the k-means clustering algorithm with a different swarm intelligence (SI) optimization technique. These optimization techniques are artificial bee colony (ABC), cuckoo search optimization (CSO), and grey-wolf optimizer (GWO). The proposed systems are evaluated using a MovieLens dataset in terms of means absolute error (MAE), precision, sum of squared errors, and recall. The experimental results indicate that the proposed algorithms are more efficient compared with existing clustering-based collaborative recommendation systems. In fact, the recommender system that combines the k-means algorithm with the ABC optimization technique performs slightly better than the other two proposed RSs.

2. SWARM INTELLIGENCE

There are times in our lives where we face problems that we must solve quickly and efficiently. That is, to find the best solution among others within a reasonable time limit. Such a problem could be finding the shortest way to go to work. This kind of problems, where the objective is to find the best solution from all feasible solutions at a certain time, is called optimization problems. Optimization problems and their solution methods have been studied for years. These problems can be classified and described in different ways depending on whether they are continuous or discrete, constrained or unconstrained, static or dynamic, single or multi-objective, linear or non-linear.

Many optimization algorithms and methods have been developed to solve various optimization problems. An important class of such algorithms is metaheuristics. A metaheuristic algorithm can be defined as a higher-level algorithm that combines one or more heuristic procedures and guides them in an intelligent way to solve a wide variety of general classes of optimization problems [38]. Metaheuristics share three main characteristics: they are nature-inspired as they are based on some principles from physics and biology, they make use of stochastic components as they use random variables, and they have several parameters that need to be fitted to the problem to be solved. In developing a metaheuristic algorithm, the balance between two main components, exploration and exploitation, should be taken into consideration. Exploration is needed to explore the search space of the problem globally, and therefore to maintain the diversity of the solutions avoiding being trapped at a local optimum. On the other hand, exploitation is needed to find promising areas of the search space with high-quality solutions. Metaheuristics can be classified into two main classes, single-solution based metaheuristics and population-based metaheuristics. As their names suggest single-solution based metaheuristics deal with a single solution while population-based metaheuristics deal with a set of solutions (population). The present thesis makes use of a population-based method related to Swarm Intelligence (SI).

Swarm intelligence is a branch of metaheuristics that takes inspiration from the collective behavior of a group of social insect colonies and of other animal societies such as ants, fishes, bees, birds, and termites. This behavior is the result of the local interactions of the individuals with each other as well as their environment. The interaction between two individuals or the individual and the environment follows simple rules and the result from an interaction would either impel or restrain the behavior of a certain individual as positive feedback or negative feedback, respectively. The decision of a certain individual might be affected by random factors in addition to the result of interaction, which leads to fluctuations. The interaction occurs whenever a certain individual needs to make a decision. Collective behavior refers to a swarm, in which the individual behavior may be random, however, the aggregation of individual behavior turns out to be globally intelligent [35].

2.1 Properties of the swarm intelligence approach

The SI approach has various properties that are worth mentioning and describing. According to [36] such properties are:

- Autonomy: In a swarm intelligence approach, individuals are autonomous, controlling their own behavior in a self-organized way. No central control mechanism exists and the behavior of individuals is self-determined.

- Adaptability: Individuals can detect changes in the environment dynamically through communication forms. Thus, they can adapt their own behavior to the new changes.
- Scalability: The number of individuals used to solve an optimization problem can be easily increased without changing anything in the control architecture.
- Flexibility: No single individual of the swarm is essential, that is, any individual can be dynamically added, removed, or replaced.
- Robustness: As mentioned, no central control is needed, which means that there is no single point of failure.
- Self-organization: Every individual is a solution to the optimization problem to be solved and is constructed while the program runs. In other words, no individual is predefined or preprogrammed.
- Cost effectiveness: A SI approach consists of a finite collection of homogeneous agents that share the same capabilities and control algorithm. Thus, the implementation of an SI-based algorithm is a simple task.

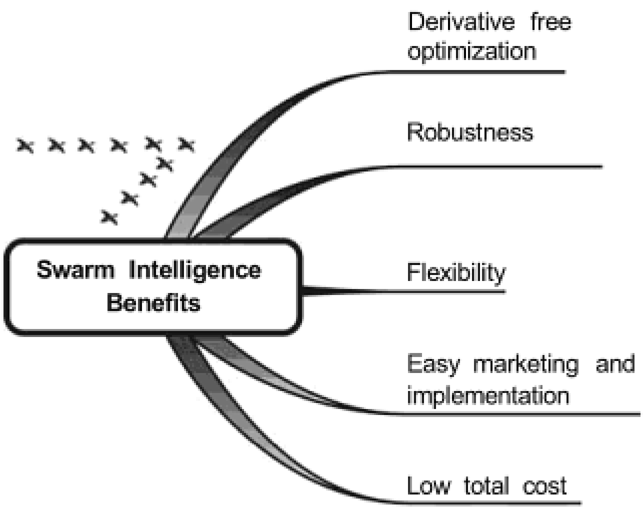

Figure 1: Key benefits from swarm intelligence [49]

2.2 Applications of swarm intelligence

Besides the applications to conventional optimization problems, SI can be employed in library materials acquisition, telecommunications, medical dataset classification, dynamic control, heating system planning, moving objects tracking, and prediction [32]. A well-known application of swarm intelligence is that of crowd simulation. Artists use SI for rendering, realistically depicting the movements of groups of fish and birds. The movie “Batman Returns” made use of SI for showing the movements of a group of bats. Another popular film trilogy named “Lord of the Rings” also made use of SI during battle scenes. It is clear that this metaheuristic has much to offer in a variety of fields of science.

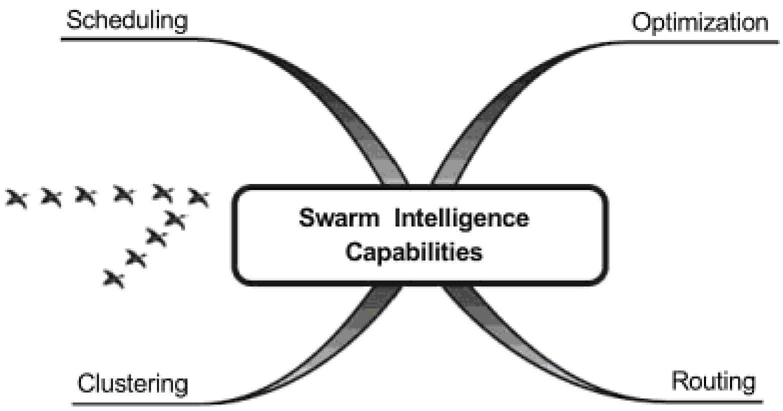

Figure 2: Key capabilities of swarm intelligence [49]

2.3 Paradigms of swarm intelligence algorithms

Below, various SI algorithms are described, such as Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Chicken Swarm Optimization (CHSO), Glowworm Swarm Optimization (GSO), Bat Algorithm (BA), and Artificial Fish Swarm Optimization (AFSO) for making the concept and the properties of swarm intelligence clear. A well-known and widely used SI algorithm called Artificial Bee Colony (ABC) that imitates the foraging behavior of honeybees, is described in detail in chapter 5. In addition, two more SI algorithms, the Cuckoo Search Optimization (CSO) that is inspired by the brood parasitism of some cuckoo species, and the Grey Wolf Optimizer (GWO) that utilizes the hierarchal order of a grey wolf pack, are described in chapter 6.

2.3.1 Ant Colony Optimization

The ant colony optimization was introduced by M. Dorigo and colleagues [33] as a population-based metaheuristic that can be used in difficult optimization problems. It is inspired by the foraging behavior of real ants which form and maintain a line to their food source by laying a trail of pheromone, i.e., a chemical compound which attracts other ants. Whenever an ant finds a food source, it returns to its nest following its own path and deposits a certain amount of pheromone. In this way, shorter paths have a bigger concentration of pheromone, as a result, other ants are more attracted to these paths leading to the equally valuable food sources and mark such paths with additional pheromone. More formally, in ACO, a set of software agents called "artificial ants" search for good solutions to a given optimization problem transformed into the problem of finding the minimum cost path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model, that is, a set of parameters associated with graph components (either nodes or edges) the values of which are modified at runtime by the ants. ACO is very efficient in solving difficult discrete combinatorial problems. Moreover, its extensions can be used in other categories of optimization problems such as continuous optimization problems.

2.3.2 Particle Swarm Optimization

The particle swarm optimization (PSO) was initially introduced in 1995 by James Kennedy and Russell Eberhart as a global optimization technique [34]. PSO is a population-based stochastic optimization technique for the solution of continuous optimization problems. It is inspired by social behaviors in flocks of birds and schools of fish. In particle swarm optimization (PSO), a set of artificial agents called particles search for good solutions to a given continuous optimization problem. Each particle is a solution of the considered problem and uses its own experience and the experience of neighbor particles to choose how to move in the search space. In practice, in the initialization phase each particle is given a random initial position and an initial velocity. The position of the particle represents a solution of the problem and has therefore a value, given by the objective function. While moving in the search space, particles memorize the position of the best solution they found. At each iteration of the algorithm, each particle moves with a velocity that is a weighted sum of three components: the old velocity, a velocity component that drives the particle towards the location in the search space where it previously found the best solution so far, and a velocity component that drives the particle towards the location in the search space where the neighbor particles found the best solution so far. PSO has been applied to many different problems and is another example of successful artificial/engineering swarm intelligence system [39].

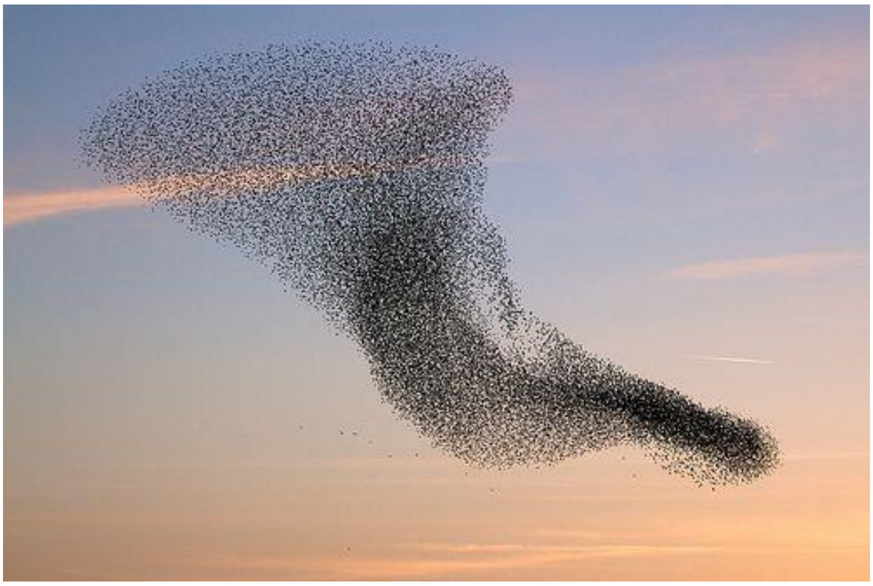

Image 1: Swarm of birds [50]

2.3.3 Chicken Swarm Optimization

The chicken swarm optimization (CHSO) algorithm is a swarm intelligence algorithm developed in 2014 by Meng et al. [56]. This algorithm utilizes the hierarchal order in the chicken swarm and the several behaviors of the chicken swarm. The chicken swarm is divided into three groups: rooster, hens, and chicks. This division depends on the food searching capability of the chickens. Chickens search for food together in a group, but each chicken has a different food searching capability and behavior. The chickens communicate among themselves by using over 30 distinct sounds. The sounds made by chickens for expressing pleasure, distress, panic, and danger are different [55]. The CHSO algorithm mimics the chickens' behavior by following the rules below:

- The chicken swarm is divided into several groups. Each group consists of a dominant rooster, a couple of hens, and chicks.
- The division of the chicken swarm depends on the fitness values of the chickens themselves. The chickens with the highest fitness value are designated as roosters, chickens with intermediate value are designated as hens, and so on.
- The mother-child relationship is established randomly. The hierarchical order and mother-child relationships are updated every several time steps.
- The hens go behind their group mate rooster and the chicks go behind their mother while searching for food. Chickens would also try to steal the food found by others, rising competition for food inside the group. The dominant individuals have an advantage in this competition.

2.3.4 Glowworm Swarm Optimization

The glowworm swarm optimization (GSO) algorithm is a swarm intelligence-based algorithm for optimizing multi-modal functions. GSO mimics the glow behavior of glowworms and was introduced in 2005 by Krishnanand and Ghose [42]. Glowworms glow at various intensities by modifying the intensity of a chemical called luciferin emission. Glowworms use their capability to attract mates during reproduction or prey for feeding. If a glowworm emits more luciferin, it glows more brightly, and it attracts more glowworms or prey. The brighter the glow intensity, the better the attraction is [43]. In GSO, a swarm of artificial glowworms is initially randomly distributed in the solution space. Each glowworm represents a potential solution to the optimization problem and carries a certain quantity of luciferin. The luciferin level of a solution (glowworm) indicates its quality. In other words, a brighter glowworm means a better solution. Each glowworm identifies other glowworms as neighbors within a certain radius using a probabilistic method. An artificial glowworm can be attracted by a neighbor only if the latter has higher luciferin intensity. The density of a glowworm's neighbors affects its direction of movement. More specifically, when the density of the neighbors is low, the glowworm's radius will increase to find more neighbors; otherwise will decrease, and smaller groups of artificial glowworms will be created. The above process is repeated until the algorithm satisfies the termination condition.

2.3.5 Bat Algorithm

The bat algorithm (BA) is a metaheuristic algorithm that was developed in 2010 by Xin-She Yang [44]. It was inspired by the echolocation behavior of microbats and other nocturnal animals. The echolocation is the process where bats emit echo signals in the space to navigate in the surroundings. Moreover, using this technique, bats can find the exact location of any object or prey, even in complete darkness. While hunting for prey, bats adjust their flight speed, the frequency, and intensity of their echo. BA is developed based on the hunting behavior of bats. In this optimization algorithm, a population of bats is initialized randomly. Each bat represents a solution and as in real life it has a velocity, a frequency, a position, and a loudness. New bats (solutions) are created by updating the velocities and positions of other bats enabling the exploration of the search space. The generation of new solutions, the evaluation of them, and therefore the search for the best one, continues until the best solution selected or a certain stop condition is met.

2.3.6 Artificial Fish Swarm Optimization

The AFSO (Artificial Fish Swarm Optimization) is a swarm intelligence algorithm that is inspired by the collective movement of fish and their social behavior [45]. The basic idea of this algorithm is to mimic the fish behavior such as searching for food, aim at finding a global optimum to an optimization problem. In AFSO, each artificial fish has a certain vision area and a position in the search space. If the artificial agent identifies a position, within its vision area, with bigger food concentration (objective function value), it moves towards it. Moreover, every artificial fish has a neighborhood. If the center of its neighborhood has a bigger concentration of food than its position, the fish will prefer to move to the neighborhood's center, provided that is not very crowded. An artificial fish can follow a neighborhood partner that has the best position too. Fishes swim randomly in the water, so in a phase of the AFSO, an artificial fish can choose a state at random in the vision area and move towards this state. This random selection accelerates the exploration of the artificial agent. AFSO executes the above fish behavior for several iterations until a termination condition is satisfied.

3. RECOMMENDER SYSTEMS

The amount of information in the world is increasing far more quickly than our ability to process it. On the internet, the number of choices is overwhelming, and the need to filter and prioritize all these choices is gradually getting bigger and bigger. Fortunately, some technologies have been developed which can help us alleviate the problem of information overload in order to find what is most valuable to us. Such technology is recommender systems.

The goal of a recommender system is to generate valuable recommendations to a group of users for various items or products that might interest them. This system is able to predict whether a user would be interested in an item or not, based on his profile. Many companies, such as YouTube, Amazon, and Netflix exploit recommender systems in order to increase their sales through personalized offers and enhanced customer experience. Recommendations improve the decision-making process and quality as they reduce the cost of searching and selecting items in an online shopping environment.

3.1 Recommendation filtering techniques

Recommender systems use two kinds of information to predict user needs and recommend related items to them. The first kind of information is about item characteristics such as categories and brand, and user characteristics such as preferences and profiles. The second kind of information is about user interactions such as ratings, number of purchases, and likes. Based on this information, various recommendation filtering techniques have been developed, aiming to create efficient and accurate recommendation systems that will provide good and useful recommendations to individual users. Fig. 1 presents the anatomy of these different recommendation filtering techniques, which are explained in detail below [1].

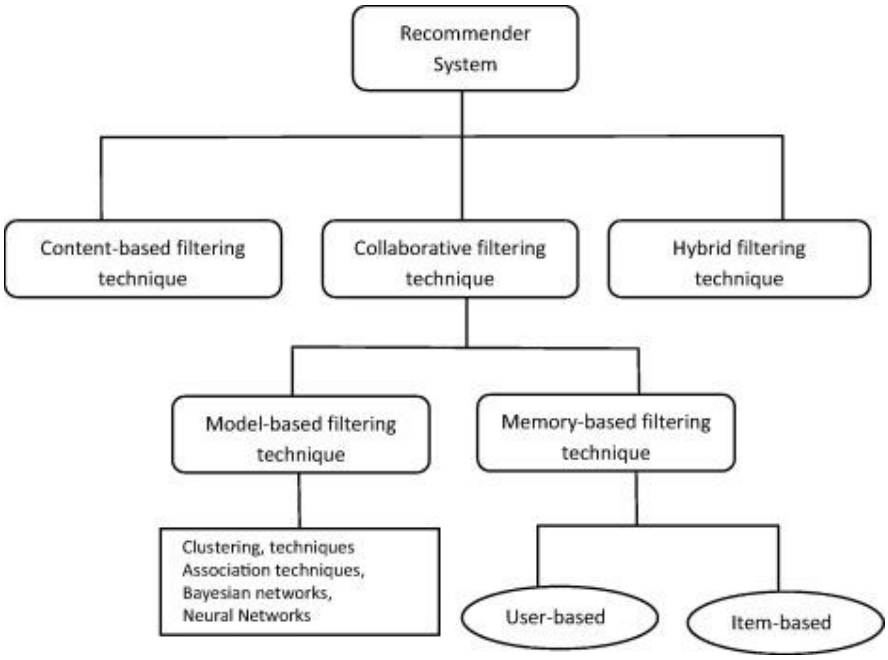

Figure 3: Recommendation filtering techniques [1]

3.1.1 Content-based filtering technique

Content-based technique generates predictions based on items' description and users' preferences. Items' features play a key role in content-based filtering approach. Such features, on a movie recommender system, could be the category, the actors, and the duration of movies. Similar items are usually grouped based on their features. Items that are more similar to the positively rated items are recommended to the user. In short, this approach makes the hypothesis that if a user was interested in an item in the past, he will be interested in a similar item in the future. Furthermore, content-based filtering technique uses different types of models to find similarity between documents and therefore to generate meaningful recommendations. It could use Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier, Decision Trees, or Neural Networks to model the relationship between different documents within a corpus. These techniques make recommendations by learning the underlying model with either statistical analysis or machine learning techniques [1].

3.1.1.1 Advantages and disadvantages of content-based filtering technique

Content-based technique is resistant to change. Even if the user profile changes, this technique can adjust its recommendations very quickly. Because content-based technique does not generally rely on user ratings or preferences, the recommendation accuracy remains unaffected even if the database does not contain any user preferences. This also means that the profile of other users is not needed on the generation of recommendations. However, a key issue that content-based technique must deal with is the lack of content. The effectiveness of such techniques is strongly connected with the quantity of the item's metadata.

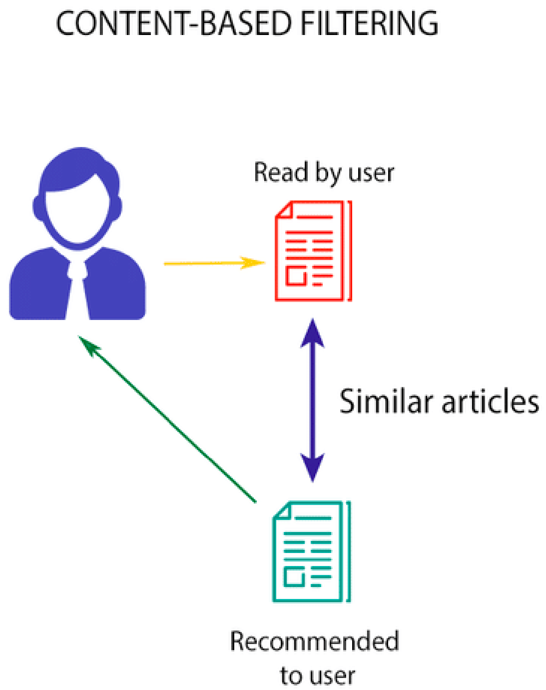

Figure 4: Content-based filtering technique in high-level [51]

3.1.2 Collaborative filtering technique

Collaborative filtering makes recommendations based on historic users' preferences for items. Users' preferences could be ratings, likes, clicks, purchases, etc. This technique matches users with relevant interests and preferences to make recommendations. Such users build a group called neighborhood. A user gets recommendations to those items that he has not rated before and that were already positively rated by users in his neighborhood [1]. For instance, if two users U1 and U2 belong in the same group and both like the item T1 and user U2 likes the item T2 too, then the first user may also be interested in the item T2. The technique of collaborative filtering can be divided into two categories: memory-based and model-based.

3.1.2.1 Memory-based technique

Memory-based technique uses user preferences to compute the similarity between users or items. Memory-based collaborative filtering can be achieved either through user-based or item-based technique. In user-based technique similar users who have similar ratings for similar items are found. The main goal is the prediction of a rating of an unrated item of a target user. The prediction gets its value from a weighted average of the ratings of the target item by users similar to the target user. Unlike user-based collaborative filtering, item-based collaborative filtering focuses on what items from all the options are more similar to what the user likes. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items. The two most popular similarity measures are Pearson correlation coefficient and Cosine similarity.

3.1.2.2 Model-based technique

Model-based collaborative technique relies on user-item interaction information. Predictions are made via models, thus machine learning and data mining techniques are needed. In this technique, the existing features are not given explicitly to the model as it could be done for content-based approach described above. Instead, the model can discover useful features by itself and make its representations of both users and items. This technique has large coverage as it can recommend a larger number of items to a larger number of users, compared to other techniques like memory-based [2].

3.1.2.3 Advantages and disadvantages of collaborative filtering technique

The main advantage of collaborative filtering technique is the ability to give accurate predictions even if there is not much content associated with items or the content is difficult to be analyzed. Collaborative filtering, as mentioned above, can also recommend items that are relevant to the target user even without the content being in the user's profile. However, this recommendation technique has to cope with a number of problems such as the cold-start problem that occurs when recommendations should be given to a new user with no ratings in any item, the data sparsity problem that occurs when only a few of the total number of items are rated by users, the scalability problem that occurs when recommendations should be provided while the number of users and items grows continually, and the synonymy problem that occurs when similar items have different names or entries and as a result recommender systems cannot make the distinction between them.

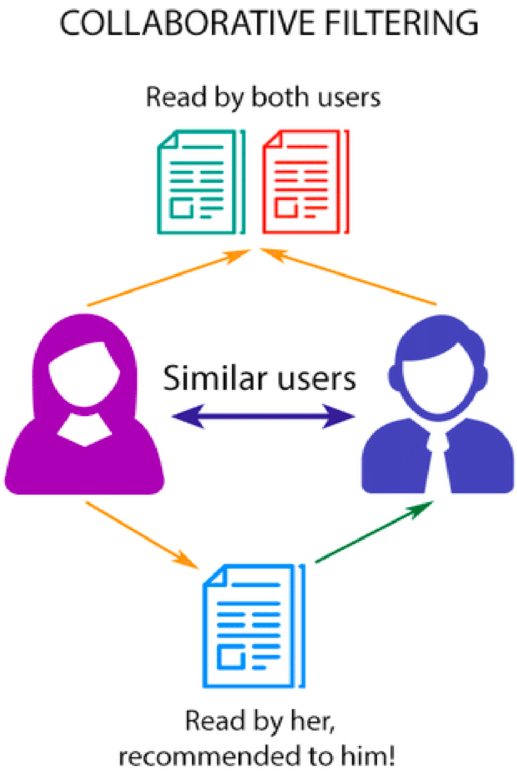

Figure 5: Collaborative filtering technique in high-level [51]

3.1.3 Hybrid filtering technique

As its name suggests, hybrid filtering technique combines content-based and collaborative filtering to gain better recommendations and overcome issues and challenges that rise by using only one of these pure recommendation techniques. The idea behind hybrid techniques is that a combination of algorithms will provide more accurate and effective recommendations than a single algorithm as the disadvantages of one algorithm can be overcome by another algorithm [1]. The combination of approaches can be done using any of the seven hybridization techniques below [3]:

- Weighted hybridization: Combines the score of different recommendation techniques to generate a recommendation list or prediction.
- Switching hybridization: Different recommendation techniques are available but only one is applied according to the user preference.
- Mixed hybridization: Recommendations from different recommenders are presented together.
- Feature-combination hybridization: The features produced by a specific recommendation technique are fed into another recommendation technique.
- Feature-augmentation hybridization: One recommendation technique is used to compute a feature or set of features, which is then part of the input to the next technique.

- Cascade hybridization: An order of preference among different items is constructed. One technique gives its recommendation to another one to be polished.
- Meta-level hybridization: The internal model generated by one recommendation technique is used as input to another.

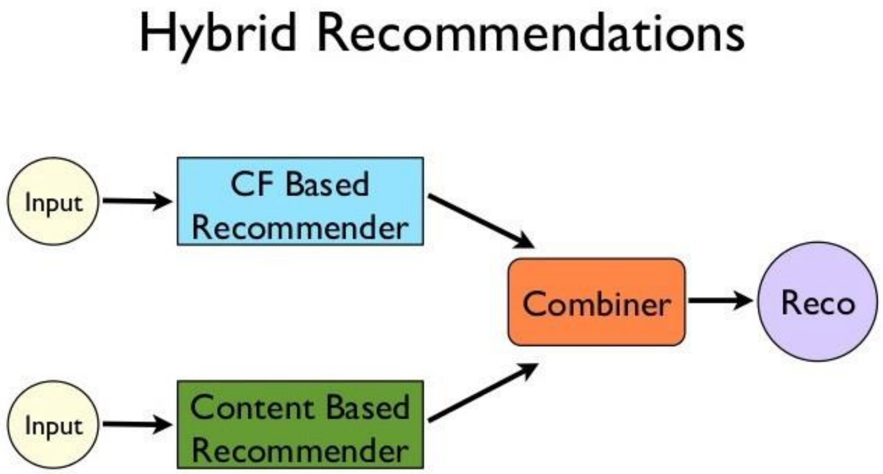

Figure 6: Hybrid filtering technique in high-level [52]

3.2 Evaluation of recommender systems

The goal of a recommender system is to provide accurate recommendations to its users that match their needs and their profiles. Thus, it is important to know whether a recommender system is effective or not. The evaluation of a recommender system can be done using various metrics or users themselves. The type of metrics used depends on the type of filtering technique. Some metrics widely used are accuracy, recall, precision, mean absolute error (MAE), and root mean square error (RMSE). Finally, another way of evaluation is to measure user reactions given the recommendations made. Such evaluation could be the measurement of user clicks on the recommended items.

4. SWARM INTELLIGENCE IN RECOMMENDER SYSTEMS

Swarm intelligence has done a great job in solving difficult and complex optimization problems. In the last few years, SI comes to try its strength on eliminating the challenges faced by traditional recommendation techniques. The use of SI for context-aware, social or multi-objective recommendation is an emerging trend in the RS research, with applications including web pages, movies, books, e-learning, and friends recommendation [4].

This chapter tries to cover the usage of swarm intelligence methods in recommender systems, reviewing a wide number of research articles. The publication date of the selected papers is within the last 3 to 5 years. The presented SI approaches in recommender systems are categorized as Ladislav Peška et al. [4] propose, which is based on the way of application of SI techniques and the recommending paradigms. These categories are:

- Feature weighting and feature selection approaches
- Clustering-based approaches
- Approaches in graph-based recommendation techniques
- Ensemble approaches and re-ranking
- Approaches in latent factor models

4.1 Feature weighting and feature selection approaches

- Choudhary, Kant and Dwivedi [5]: This paper focuses on the challenges faced by multi-criteria recommender systems (MCRSs). This class of recommender systems generates recommendations to users by considering various criteria ratings. Several aggregation functions are used in the computation of total similarity between users. However, these functions are unable to reveal the optimal weights of a user on various criteria. The authors try to deal with this challenge, that is learning the optimal weights of users on various criteria in the process of aggregation, by using particle swarm optimization (PSO). PSO contains a population of candidate solutions. Each candidate solution is a set of weights for each criterion. A fitness function is computed for each candidate solution and after several iterations and therefore searches near a candidate solution, the solution with the best fitness value, that is the best weights for each criterion, is chosen. To validate the effectiveness of their proposed approach, the authors examined the results on precision, recall, f-measure, and coverage against various similar approaches. The evaluation performed on a Yahoo Movie dataset and results indicate the superiority of the proposed approach.
- Choudhary, Mullick and Nagpal [6]: This paper explores the use of a bio-inspired meta-heuristic algorithm, named gravitational search algorithm (GSA), in recommendation systems. While generating recommendations for a user, a set of the user's closest neighbors is considered. To find these users the Euclidean Distance between users should be calculated on the feature vectors. The GSA technique has been used for developing appropriate weights for the features, which are then used for finding neighbors of the user. Gravitational search algorithm is similar to PSO. For this reason, the experiments performed are compared to that of PSO. More specifically, the experiments have been conducted on the Jester Dataset and mean absolute error, precision, recall, and

f1 score have been used as evaluation metrics. The results show that a recommender system based on GSA can produce comparable if not better results than a recommender system based on PSO.

- Gupta and Gusain [7]: The authors focus on the implementation of an improved context-aware recommendation system (CARS). This recommendation system uses users' specific data about the choice they made. Such data could be the time recommendation was made in, the day or even the weather. To overcome the two primary problems that CARS faces, the authors use the differential context weighting (DCW) methodology. DCW finds the ratings, that will be used to extract recommendations, based on weights that all contexts have depending upon their importance or relevance in the data. Thus, the more accurate the weights are, the more accurate recommendations will be produced. For this reason, the authors use the swarm-based metaheuristic PSO, to optimize the weights required for DCW. Since the objective of this paper was to find the best similarity function that can be employed in DCW to obtain recommendations, there is no comparison between the proposed approach with another swarm-based approach.
- Hamada and Hassan [8]: This paper focuses on the implementation of an effective multi-criteria recommender system. As mentioned above, how to use multiple ratings for various attributes of items during the recommendation process is one of the problems of these systems. The article presents a neural-network-based multi-criteria recommender system integrated with k-nearest neighborhood collaborative filtering for unknown criteria ratings prediction. As the authors say, the efficiency of an artificial neural network in solving prediction problems, is strongly connected with the algorithm used to train the network. The authors use the particle swarm optimization algorithm to train the artificial neural network and investigate the significance of this swarm intelligence method in improving the prediction and accuracy of the recommender systems. The proposed approach has been tested with a Yahoo! Movie dataset for recommending movies to users. Some evaluation metrics used are the mean average error, root mean square error, precision, recall, f-measure, and Gini coefficient. The experimental results of the proposed approach are compared with two KNN-based traditional RSs, one user-based and one item-based, and three ANN-based MCRSs. The first ANN-based is integrated with a user-based KNN, the second is integrated with an item-based KNN and the third one is the combination of the previous two ANN-based systems. The results show that all the ANN-based systems are much better than systems that use traditional techniques with the hybrid one to be the best one. In general, the experiments indicate that a neural network trained with the PSO algorithm could be used in MCRSs to improve their accuracy.
- Katarya and Verma [9]: This research work proposes a collaborative recommender system that utilizes the meta-heuristic gray wolf optimizer algorithm (GWO) and fuzzy c-mean technique and predicts a movie rating for a particular user based on his preferences and similarity of users. With the help of gray wolf optimizer, features of movies in which users have rated are initialized. In other words, GOW is used in features reduction of the selected movie dataset. After the features reduction, FCM is used to classify the users in the dataset by similarity of user ratings. For the evaluation of the proposed RS, various evaluation metrics were used such as mean absolute error, standard deviation, precision, and recall. The authors compare their recommender system with

several already established systems such as K-means, FCM, and PCA. The results indicate that the proposed recommender system offers slightly better recommendations when compared with the other RSs.

- Tomer et al. [10]: This paper introduces a hybrid user data model, as it is a combination of different data filtering techniques, and uses GSA for feature weight optimization. The proposed model exploits user ratings, demographic data as well as data derived from item description for creating a fuzzified user profile. GSA is used to give the appropriate weights to different features of each user aiming at an effective and accurate computation of similarity of users. The proposed technique has shown better results than Pearson correlation based Collaborative Filtering (PCF), Fuzzy Collaborative Filtering (FCF), Fuzzy Genetic Algorithm based Collaborative Filtering (FG-CF) and Fuzzy Particle Swarm Optimization based Collaborative Filtering (FPSO-CF). The results were analyzed using MAE and coverage.
- Wang et al. [11]: The authors propose a RS model based on the support vector machine (SVM). Recommendations are generated using items' content information and users' demographic and behavior information. The performance of SVM algorithm is fully connected with the used parameters. To achieve optimal performance, the authors use an improved PSO algorithm for the parameter optimization of SVM. For the evaluation process, the MovieLens 1M data set was selected. The proposed algorithm is compared to traditional techniques and two other approaches that use different algorithms for the parameter optimization of SVM. The results show that the proposed RS outperforms the other methods overall.
- Yang et al. [12]: This work proposes a different context-aware recommender system. More specifically, the authors introduce a new Item Splitting approach called Complex Splitting. Item Splitting is based on the idea that one item can be considered as two different items under different contextual conditions as a result to improve the accuracy of the recommendations. The difference with Complex Splitting is that the latter takes into consideration multiple contextual conditions to split a user or an item. The use of Complex Splitting arises two challenges, the selection of the optimal contextual condition combination and the determination of the number of contextual conditions to be selected for each user. To deal with these challenges the authors use the discrete binary particle swarm optimization transforming these challenges into a contextual conditions combinatorial optimization problem. Based on the evaluation metric RMSE, the authors show that the proposed approach can further improve the accuracy of recommendations.
- Yadav et al. [13]: This paper tries to enhance the recommendation accuracy of the traditional collaborative-filtering based recommender system. This work uses a heuristic swarm intelligence technique called Bat Algorithm (BA) to compute the items' weights, aiming at a better neighborhood for the active user and therefore at better personalized recommendations. The proposed approach was compared to that of ABC based approach. As evaluation metrics were used the MAE and the f1 score and as experimental dataset the Jester datset-1. The results indicate that BA performs much better than ABC.

4.2 Clustering-based approaches

- Bastami et al. [14]: This work presents a new unsupervised approach that tries to improve the performance of the similarity-based link prediction problem. The goal of the aforementioned problem is to predict unobservable links, missing links or links that will be formed in the future in a social network. The proposed approach consists of three steps, community detection, optimization subgraph and local link prediction. Swarm intelligence appears with GSA algorithm in the second step. GSA tries to find optimized subgraph(s) by merging the selected strong communities derived from the first step. These optimized subgraphs are used in the last step for concurrent predictions ending up in a set of link predictions. Various datasets (Blogs, Netflix, Cora, etc.) and metrics (accuracy, precision, recall and AUC) were used in order to evaluated the presented approach. The experimental results show high accuracy and speed of the proposed approach as well as significant reduction of CPU time and memory space usage.
- Ganesan and Selvaraju [15]: This paper proposes an effective way for web user grouping in web search applications. The user's clustering aiming at a formation of groups of users with similar web travel and is achieved by a PSO algorithm in association with Open Directory Project (ODP) dataset. The search data were collected by using a crawler, so a data cleaning process was needed. Having clean data and the categories for clustering the users from the ODP dataset, the authors applied PSO technique to identify the best cluster of each user. The authors compared the proposed approach to the K-means and DB-Scan clustering methods in terms of purity and entropy measures. From the results, it seems that the user clustering with PSO approach performs better.
- Katarya [16]: In this research work, a hybrid recommender system is proposed which combines k-means clustering algorithm with a bio-inspired artificial bee colony (ABC) optimization technique. The authors present a unified model that is applied to a MovieLens dataset for improved efficient predictions. As any SI method, ABC tries to find the best solution among various candidate solutions, which are randomly initialized, by computing the fitness value for each one of them. In this model, k-means algorithm is used in the computation of the fitness value of each candidate solution. The authors compute the precision, recall, and MAE measures for different approaches, showing that the presented model has slightly better results. A similar approach that uses cuckoo search instead of ABC algorithm is proposed in Katarya and Verma [30]. From the experiments conducted for the two approaches, the cuckoo search based approach and the ABC based approach, it seems that cuckoo search gives slightly better results than ABC.
- Katarya and Verma [17]: The authors present a collaborative recommender system enhanced with particle swarm optimization technique that improves the concerns of both high dimensionality and data sparsity. The proposed approach is a hybrid model that uses k-means, PSO and fuzzy c-means, and focuses on movie recommendations. Along with this hybrid-model, the authors use a method named 'type division'. Type division converts the initial dataset, that contains movies and ratings for these movies by a user, to a new form in which users are divided based on types of movies they watched. A combination of k-means and PSO is applied to the new dataset finding initial centers that are optimized and ready to be fed to fuzzy c-means for additional optimization. In other words, at first, k-means algorithm calculates the desired number of cluster centers to give them as input to PSO as an alternate to their random allocation. The final output

of k-means and PSO is a set of optimized centers that are used by FCM to form the final clusters. When compared to already existing methods in terms of MAE, the proposed approach gives improved results.

- Katarya and Verma [18]: In this research work, an improved CF-based recommender system regarding accuracy and efficiency is proposed. The authors present a fuzzy c-mean and a bio-inspired approach called artificial algae algorithm (AAA). In this approach initially, a rating matrix is provided to FCM to produce a cluster's number of each user. PCC is applied to FCM clusters to produce intermediate similarities of users. Then AAA optimization approach is implemented. In AAA initially, all users (algae) are distributed randomly in different clusters (colonies) and the final cluster for each user is computed. Combining the intermediate similarities resulting from AAA and FCM, the authors get the final user similarities that eventually use to produce item recommendations to the users. In the experimental evaluation, the authors used four real datasets concluding that the proposed RS delivered better recommendations for all four datasets when compared to other alternatives. The efficiency of the system was estimated by evaluation metrics such as MAE, precision, recall. Finally, the authors compared the previously described approach (Katarya and Verma [17]) with the one described here showing that the latter performs better.
- Vellaichamy and Kalimuthu [19]: In this work, a hybrid collaborative movie recommender system is presented that combines fuzzy c-means clustering with bat optimization SI technique to reduce the scalability problem and enhance the clustering. The authors use fuzzy c-means clustering technique to partition the users into groups. The accuracy of clustering is strongly connected with the initial cluster center points that are given as input in FCM. The random initialization of the cluster center points can result in a local optimum solution. The authors address this issue using bat algorithm. Bat algorithm is used to obtain the initial position of clusters to be fed to FCM. The proposed system was evaluated over a MovieLens dataset. The experimental results show that the proposed RS can perform better compared to other techniques in terms of MAE, precision, and recall.
- Logesh et al. [20]: In this paper, a novel user clustering approach based on quantum-behaved particle swarm optimization (QPSO) is proposed and is applied in a travel recommendation system. The proposed approach comprises of three steps. In the first step, the users in the dataset are clustered through QPSO algorithm. In the second step, the cluster of the active user is created to be used in the third and last step where the ratings are predicted, and the top-n most relevant items are generated to the user. Moreover, the authors enhance the just described QPSO-based approach by using in the first step an ensemble model which includes the bio-inspired clustering methods QPSO, K-PSO, and K-MWO. This model aggregates the generated user clusters of the three algorithms to produce a final clustering result. Both approaches were evaluated on two real-time datasets of Yelp and TripAdvisor. The authors used four evaluation metrics precision, recall, f-measure, accuracy, and hit rate to evaluate the performance of recommendation approaches. The experimental results show that the enhanced QPSO approach performs better when compared to various alternatives such as c-means, k-means, and PSO.

4.3 Approaches in graph-based recommendation techniques

- Alathel [21]: This research applies a hybrid of the ant colony system (ACS) model and the ant system (AS) algorithm to trust-based recommender systems (TBRS), which are systems that use explicit trust values among the users. In other words, the author presents a model that can provide a user with rating prediction for an unrated item by utilizing the rating information provided by other users in the network that are not necessarily directly trusted by the user. Ant colony inspired algorithms are applied to problems that can be represented as a connected graph. For this reason, the author modeled the problem as a graph where the nodes represent users and the edges represent the trust between the users. Having the graph, several ants start from the active user to find as many ‘good users’ as possible. ‘Good users’ are users that can be reached through the active user’s web of trust. The ant that creates the path with the most ‘good users’ is the best solution. The Epinions.com dataset is used for the evaluation of the proposed RS and the results show that this approach outperforms the basic CF algorithm that uses Pearson similarity and Massa’s MoleTrust (MT).

- Beldjoudi et al. [22]: This main purpose of this research is to present a new approach to recommend educational resources in folksonomies by leveraging the structured content that is accessible via linked open data (LOD) and using ant colony optimization (ACO). Folksonomy is a classification system in which users apply social tags to online items, to make those items easier for themselves or others to find later [31]. The authors iteratively explore the RDF data graph to produce various recommendations. Using ant colony optimization, the proposed approach performs a search for the appropriate paths in the linked open data graph and selects the best neighbors of an active user to provide improved recommendations. To evaluate the quality of the proposed recommender system, the authors used the following metrics: recall, precision, and f1 metric. The presented results show that all three metrics achieved good values. However, no comparison with another approach is presented.

- Gohari et al. [23]: In this paper, a novel trust-based approach, called Semantic-enhanced Trust based Ant Recommender System (STARS) is presented. This approach satisfies the following trust properties: asymmetry, transitivity, dynamicity, and context-dependency. As its name suggests this method uses ant colony optimization to perform a depth-first search for the optimal trust paths in the trusted network and selects the best neighbors of an active user to provide better recommendations. In detail, artificial ants start from the node that represents the active user and search for valuable food sources (i.e. trusted neighbors) in the trusted network. Experimental results on real-world datasets show that the proposed RS gives good results in terms of prediction accuracy and recommendation quality and can overcome the data sparsity and cold-start problems.

- Rehman et al. [24]: This article presents a cloud-based food recommendation system that assists patients suffering from various diseases to select a proper diet that fulfills patients’ nutrition requirements. The proposed RS uses ant colony optimization technique to generate optimal food lists and recommends suitable foods according to the values of users’ pathological test results. Ant colony approach takes as input a graph of foods to generate the optimal food set for the users. Each ant constructs a solution by visiting nodes (food items) that provide the best cost in terms of low error compared to targets. A target represents the number of food ingredients required against the disease that the active user has.

The experimental results show that sufficient accuracy can be achieved by increasing the number of ants.

- Sherkat et al. [25]: This paper introduces a new link prediction algorithm in a social network based on ant colony optimization approach. A social network consists of nodes and links between them. To predict a new link between two nodes, there should be at least one relation between these two nodes. The authors transform the link prediction problem into the problem of predicting links between a source node and a target node in a community. Thus, communities that can provide valuable information for link prediction should be found. ACO undertakes to find such communities which are triangle subgraphs and are used later for link prediction. The authors apply the proposed algorithm to various kinds of networks. In some of them, the proposed approach gives the best result in comparison to other link prediction algorithms.
- Xing et al. [26]: In this work, a user recommendation strategy based on particle swarm for Microblog network is proposed. The authors developed a PSO-based algorithm capable to recommend users using their influence, their interactions among other users, and the coherence between them. In other words, PSO uses the aforementioned three social factors to form clusters, from which the top N users are chosen and recommended to the target user. Experimental results show that, compared to PageRank-based algorithm, the proposed approach delivers much better results in terms of precision and recall.

4.4 Ensemble approaches and re-ranking

- Katarya and Verma [27]: In this article, a hybrid music recommender system, which uses context and collaborative approaches, is proposed. The authors build a multi-layer context graph for their music recommendation system. This graph consists of user-context layer, item-context layer, and decision-context layer. User context describes a user’s personal information such as gender, age, and country. Item context describes several attributes of the item. Decision context describes attributes that determine the decision itself such as time, location, and mood. The authors use four different methods to build an efficient recommendation system. These methods are collaborative filtering, depth-first search algorithm, Bellman-ford algorithm, and particle swarm optimization. Depth-first search and Bellman-ford algorithms use the multi-layer graph to find each user’s favorite artist. PSO is used to address the task of learning to rank. Having the songs of the favorite artist of the user and the similarity score of all items from collaborative filtering, PSO produces an optimized ranked list of items to recommend. The authors compare the proposed approach with existing ones and show that the presented recommender system delivers the best recommendations regarding recall results.
- Chifu et al. [28]: This paper focuses on the implementation of a recommender system that produces recommendations for healthy meals. The authors propose a hybrid model that consists of an invasive weed optimization algorithm, a PSO algorithm, and a tabu search algorithm. This recommendation system is essentially based on the invasive weed optimization algorithm. Tabu search and PSO algorithms are the hybridization components that improve the search capabilities of the core component and produce an optimized solution. Moreover, tabu search uses concepts from the reinforcement learning technique and PSO uses the path relinking technique.

4.5 Approaches in latent factor models

- Laishram et al. [29]: In this work, the authors prove that the combination of a swarm intelligence technique or evolutionary technique with a matrix factorization technique can lead to an efficient recommender system that produces accurate recommendations. In matrix factorization, a user-rating matrix Y, with a small portion of known ratings, is factorized into two latent factors U and V such that the product U x V is similar to Y. The resulting two latent factors are used to estimate the unknown ratings. The authors propose four methods that combine either a variant of PSO technique or a genetic algorithm (GA) with the Maximum Margin Matrix Factorization (MMMF), that is a variant of MF, to find the optimal factor matrices U and V. For the evaluation of the proposed methods, the authors used a MovieLens dataset. The results indicate that their approaches provide better solutions than MF alone.

5. SWARM INTELLIGENCE IN THE CLUSTERING PROBLEM

In the previous chapter, various hybrid algorithms that use SI methods, along with other algorithms in the clustering problem, are described. One of these algorithms is the one proposed in [16]. This hybrid algorithm combines k-means and ABC to group users more efficiently, and therefore to enhance the developed recommendation system. One of the algorithms that this thesis presents, which is described in chapter 6 in detail, is inspired by the article [16] as it proposes a different combination of the two algorithms (k-means and ABC). Before the implementation of the proposed hybrid models, a primary investigation of how an SI method deals with the clustering problem was conducted. More specifically, a study of how ABC works to find optimal clusters, and a comparison between ABC and k-means, on a movie dataset, is presented along with the related conclusions. Both algorithms (k-means and ABC) are explained in detail below. In addition, the ABC algorithm is implemented by following the steps described below (section 5.3).

5.1 The clustering problem

The clustering problem is a NP-hard problem where the basic idea is to form groups of objects (clusters) in such a way that objects in the same cluster are more similar to each other than to those in other clusters. The clustering task can be achieved by various algorithms that differ in their understanding of what constitutes a cluster and how to efficiently find them. A common way to cluster some data is to transform the clustering problem into a mathematical one where a k-partition of the original data should be found. K-partition means that k clusters will be formed in such a way that no instance will belong to more than one cluster. Each group is represented by a centroid. So, the output of the clustering procedure is a set of k centroids. In other words, the centroids are points on the search space defined by the examined data, and since each centroid defines a group, each data point will be assigned to its closest centroid.

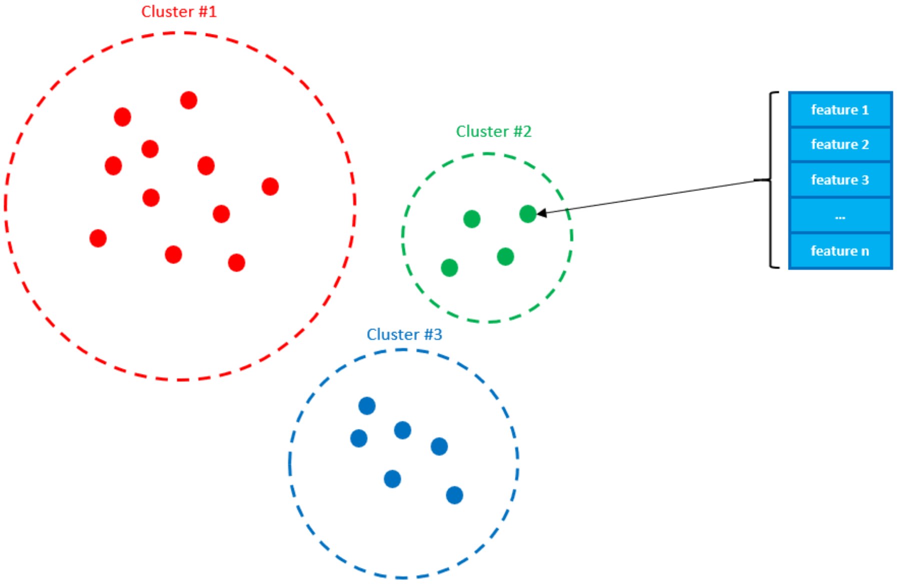

Figure 7: Clustering example [53]

5.2 K-means algorithm

The K-means algorithm is an iterative distance-based algorithm that tries to partition a dataset into distinct k groups (clusters), where each data point belongs to only one group. The main objective of this algorithm is to form clusters that are as different as possible, while keeping the points within a cluster as similar as possible. In other words, the k-means algorithm tries to minimize the sum of squared distance between the data points and the cluster's centroid. The K-means algorithm works as follows:

- Step 1: Specify the number of clusters k.
- Step 2: Select k random points from the data as centroids.
- Step 3: Assign all the points to the closest cluster centroid.
- Step 4: Recompute the centroids of newly formed clusters.
- Step 5: Repeat steps 3 and 4 until one of the following stopping criteria is satisfied:
    - Centroids of newly formed clusters do not change.
    - Data points remain in the same cluster.
    - Maximum number of iterations is reached.

5.3 Artificial Bee Colony algorithm

The ABC algorithm is a swarm based meta-heuristic algorithm which simulates the foraging behavior of honeybees and was introduced by Karaboga in 2005 for real-parameter optimization [37]. The model consists of three essential components: The employed bees, which work on the collection of food to the hive at a specific food source, the onlooker bees, which indicate when a specific food source is not worth it anymore, and the scout bees which are the ones looking for new food sources. Initially, all food source positions are discovered by scout bees. Thereafter, the nectar of food sources is exploited by employed bees and onlooker bees, and this continual exploitation will ultimately lead to their exhaustion. Then, the employed bee which was exploiting the exhausted food source becomes a scout bee in search of further food sources once again. In other words, the employed bee whose food source has been exhausted becomes a scout bee. In ABC, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of employed bees is equal to the number of food sources (solutions) since each employed bee is associated with one and only one food source. The main steps of the algorithm can be described as follows [46]:

- Step 1: Generate the initial population of solutions randomly. Let X_i^j represent the i^{th} solution (food source). Each solution is generated as follows:

  $$X_i^j = X_{min}^j + rand(0,1)\left(X_{max}^j - X_{min}^j\right)$$

  where X_{max}^j and X_{min}^j are the upper and lower bounds for the dimension j, respectively.
- Step 2: Each employed bee X_i generates a new candidate solution V_i as equation below:

$$V_i^j = X_i^j + rand[-1,1](X_i^j - X_k^j)$$

where X_k is a randomly selected candidate solution ($i \neq k$) and j is a random dimension index selected from the set $\{1, 2, ..., d\}$. Once the new candidate solution V_i is generated, the greedy selection process is used. If the fitness value of V_i is better than that of its parent X_i, X_i is replaced by V_i; otherwise X_i remains unchanged.

- Step 3: Assign onlooker bees to employed bees according to probabilities, produce new solutions, and apply the greedy selection process. The probability selection is really a roulette wheel selection mechanism which is described as equation below:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j}$$

where fit_i is the fitness value of the i^{th} solution in the swarm.

- Step 4: If a solution cannot be improved over a predefined number (called threshold limit) of cycles, then the food source is abandoned, and the employed bee becomes a scout bee for discovering new food sources randomly.
- Step 5: Memorize the best food source found so far.
- Step 6: If the termination condition is not satisfied, go to Step 2; otherwise, stop the algorithm.

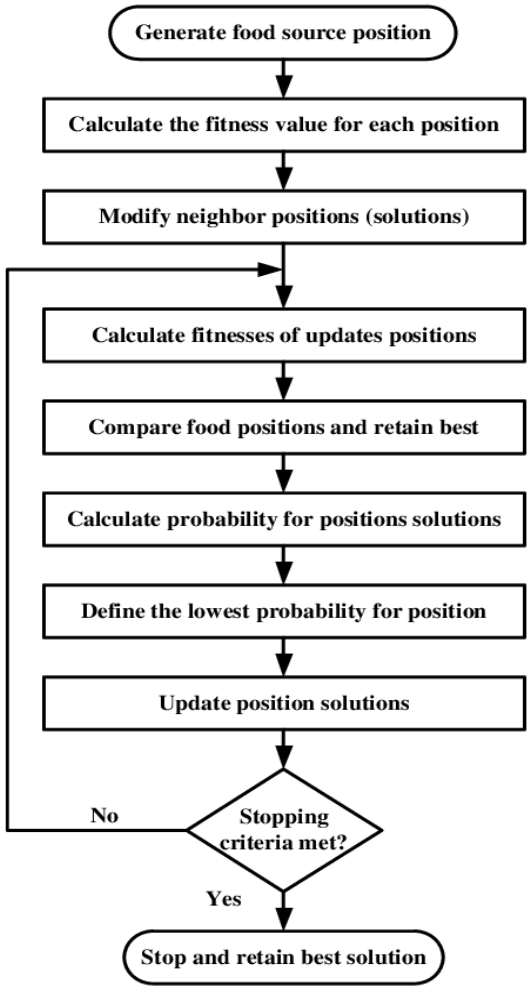

Figure 8: Steps of ABC algorithm [54]

5.4 Clustering as an optimization problem

In order to apply the ABC algorithm to the clustering problem, the transformation of the problem into an optimization one is needed. A well-defined optimization problem needs a search space, a set of d-dimensional input decision variables and an objective function. In ABC, each bee represents a whole solution, that is each bee can represent a complete set of candidate centroids. In case of a d-dimensional space, each bee will be a $\mathrm{k} \times \mathrm{d}$-dimensional vector where k is the number of the desired clusters. The boundaries of the search space indicate the upper and lower values that a centroid can have. Finally, in the clustering task, the objective is to maximize the distance between two distinct groups and minimize the inner distance within a group. Thus, as an objective function the well-known Sum of Squared Errors (SSE) could be chosen, which is a metric that computes the squared distance of each instance in the data to its closest centroid. The goal of this optimization task is to minimize this function.

$$SSE = \sum_{k=1}^{K} \sum_{\forall x_i \epsilon C_k} ||x_i - \mu_\kappa||^2 \quad (1)$$

Having transformed the clustering problem to an optimization one, the ABC algorithm can be applied to find the best k-partition of the given data. The steps of ABC algorithm for solving a clustering optimization problem, based on the described steps above, are the following:

- Step 1: Randomly generate the initial population of solutions and set the maximum number of iterations and the number of cycles that a solution can remain unchanged (threshold limit). Each solution represents a vector with k centroids. Let cost function (1) be the objective function.
- Step 2: Produce new solutions for the employed bees, evaluate them, and apply the greedy selection process.
- Step 3: Assign onlooker bees to employed bees according to probabilities, produce new solutions, and apply the greedy selection process.
- Step 4: If the search time of an employed bee is more than threshold limit, stop the exploitation process of the sources abandoned by bees and send the scouts in the search area for discovering new food sources, randomly.
- Step 5: Memorize the best food source found so far.
- Step 6: If the termination condition is not satisfied, go to Step 2; otherwise, stop the algorithm.
- Step 7: Determine the optimal centroids. Assign each data point to the closest centroid and return the final k clusters.

5.5 ABC clustering VS k-means clustering

This section tries to answer a main question: Can the artificial bee colony algorithm give comparable results, if not better, than the k-means? To answer this, various clustering results, with different number of clusters and dimensions of center points, of both algorithms are compared in terms of SSE. The clustering task is conducted on a publicly available MovieLens dataset [47] that has 100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users.

To apply ABC in the movies clustering problem, the definition of the boundaries of the search space, the decision variables and the objective function are needed. This definition is described as follows:

- Boundaries of the search space: The boundaries of the search space range from 1 to 5 as the chosen dataset contains user ratings of scale 1 – 5.
- Input variables: Each solution (bee) is a vector of candidate centroids with dimension equal to the multiplication of the number of clusters with the number of the chosen genre movies.
- Objective function: Sum of Squared Errors.

To form clusters, the algorithm exploits the user preferences, that is ratings. Specifically, not each user rating for each movie is used but, the average rating of each user of each movie genre. In other words, each user is represented in the search space as a vector of average ratings of every movie genre.

Figure 9 and figure 10 show how k-means and ABC grouped a set of users into two clusters, respectively. The users were grouped based on their average rating of romance and average rating of sci-fi movies. That is, each point (user) in the search space is a 2-dimensional vector where x is the average rating of sci-fi movies and y is the average rating of romance movies. It seems that both algorithms grouped the users in the same way. The groups are mostly based on how each user rated romance movies. If their average rating of romance movies is close or over 3, then they belong to one group (purple points). Otherwise, they belong to the other group (red points). In terms of SSE, k-means offers a slightly better solution than ABC. However, there is a case where ABC can form clusters better than k-means. This case is illustrated in Fig. 11. This figure shows, the complete clustering task of the k-means with a value of SSE bigger than the one of ABC. This happens because the efficiency of k-means is strongly connected with the random initial center points (initial points of each cluster) that are given to the algorithm. In other words, the randomly selected initial center points can lead to a local optimum as illustrated in Fig. 11.

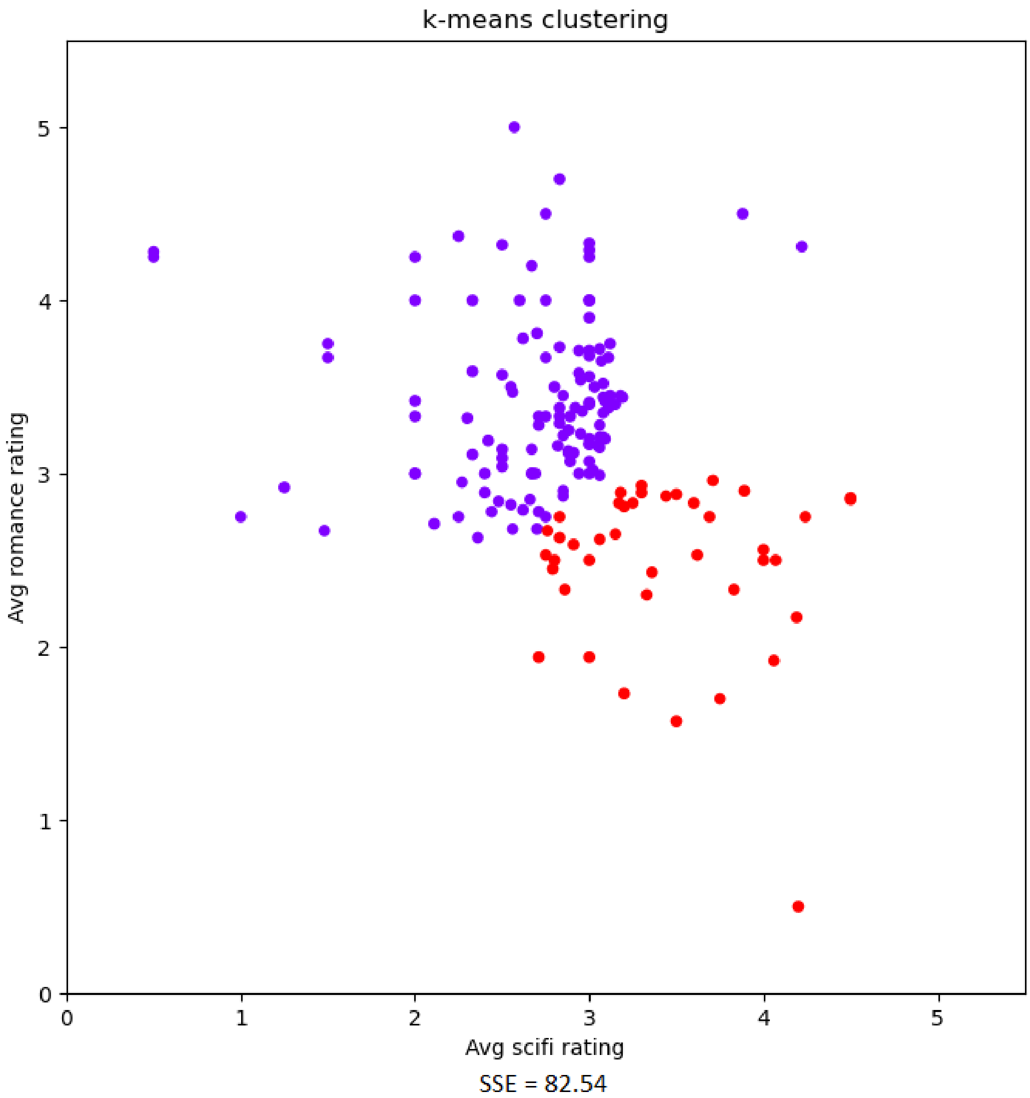

Figure 9: Data partitioning into 2 clusters with k-means

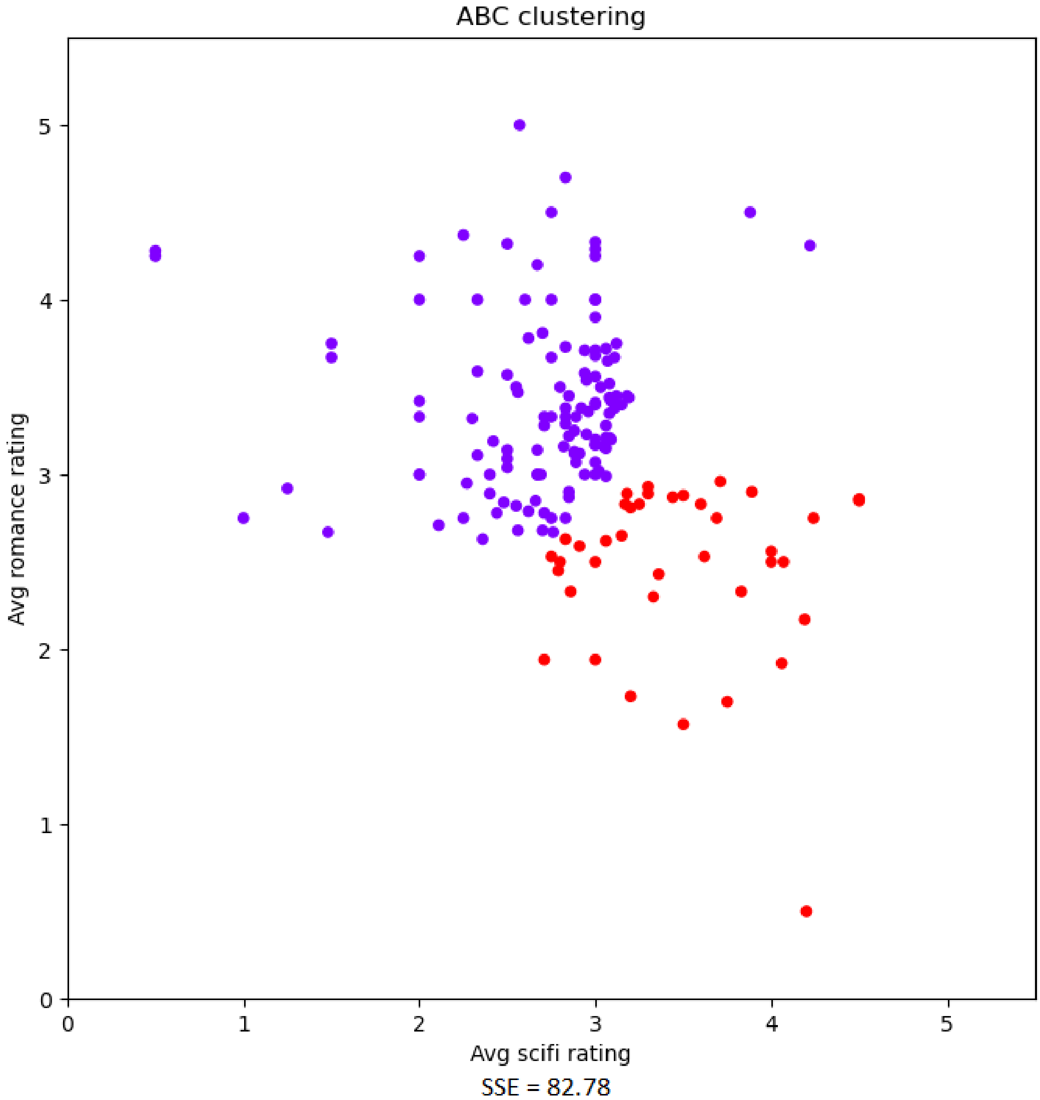

Figure 10: Data partitioning into 2 clusters with ABC

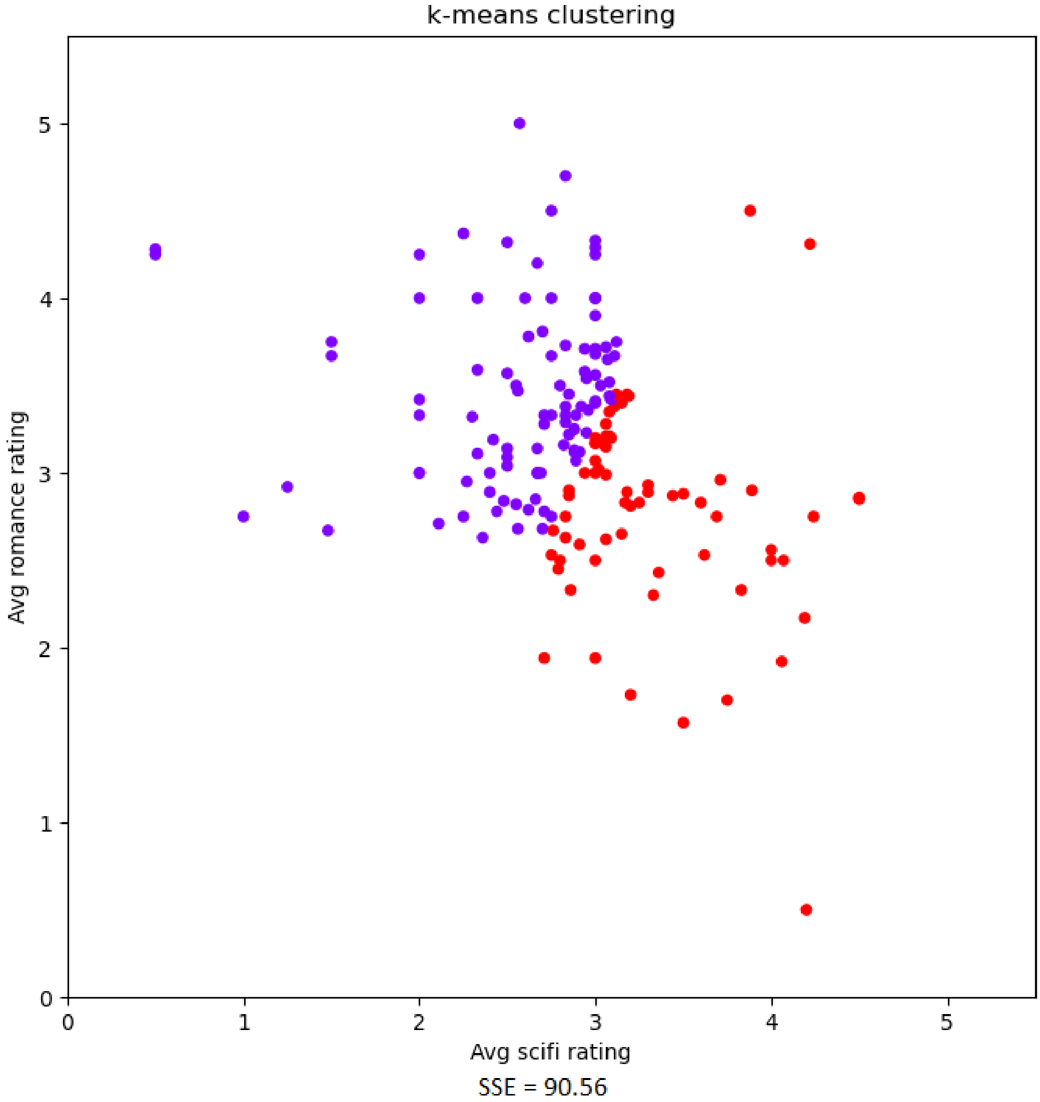

Figure 11: Data partitioning into 2 clusters with k-means – local optimum

The following two figures, figure 12 and figure 13, illustrate a different partition of the users of the previous clustering example, this into three clusters. The groups that the two algorithms have formed are quite similar and the values of SSE are too close. The users were grouped into users who like romance but not sci-fi movies (green points), users who like sci-fi but not romance movies (red points), and users who like both sci-fi and romance movies (purple points). ABC is very efficient in this example as well and can be better than k-means when the latter is trapped in a local optimum (Fig. 14).

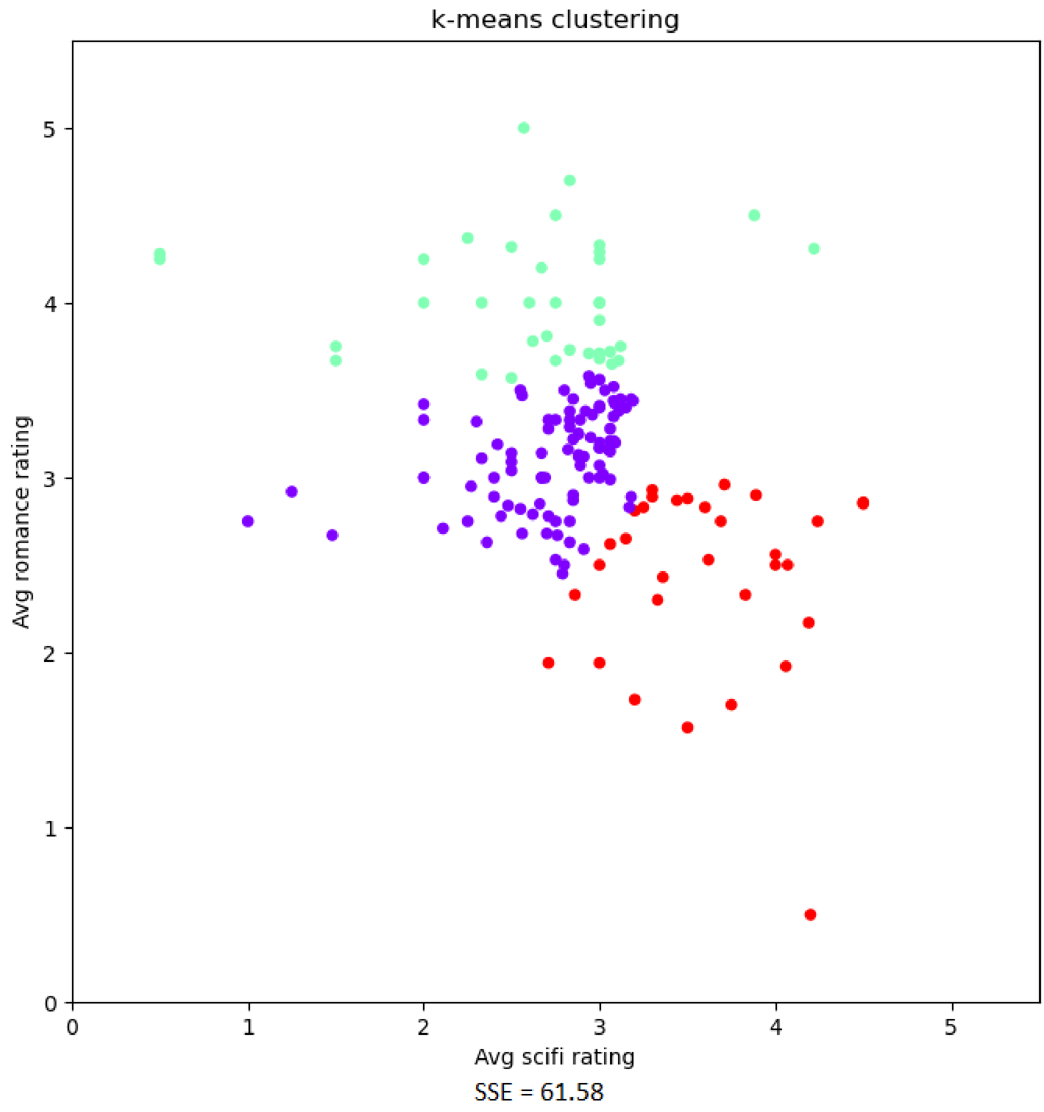

Figure 12: Data partitioning into 3 clusters with k-means

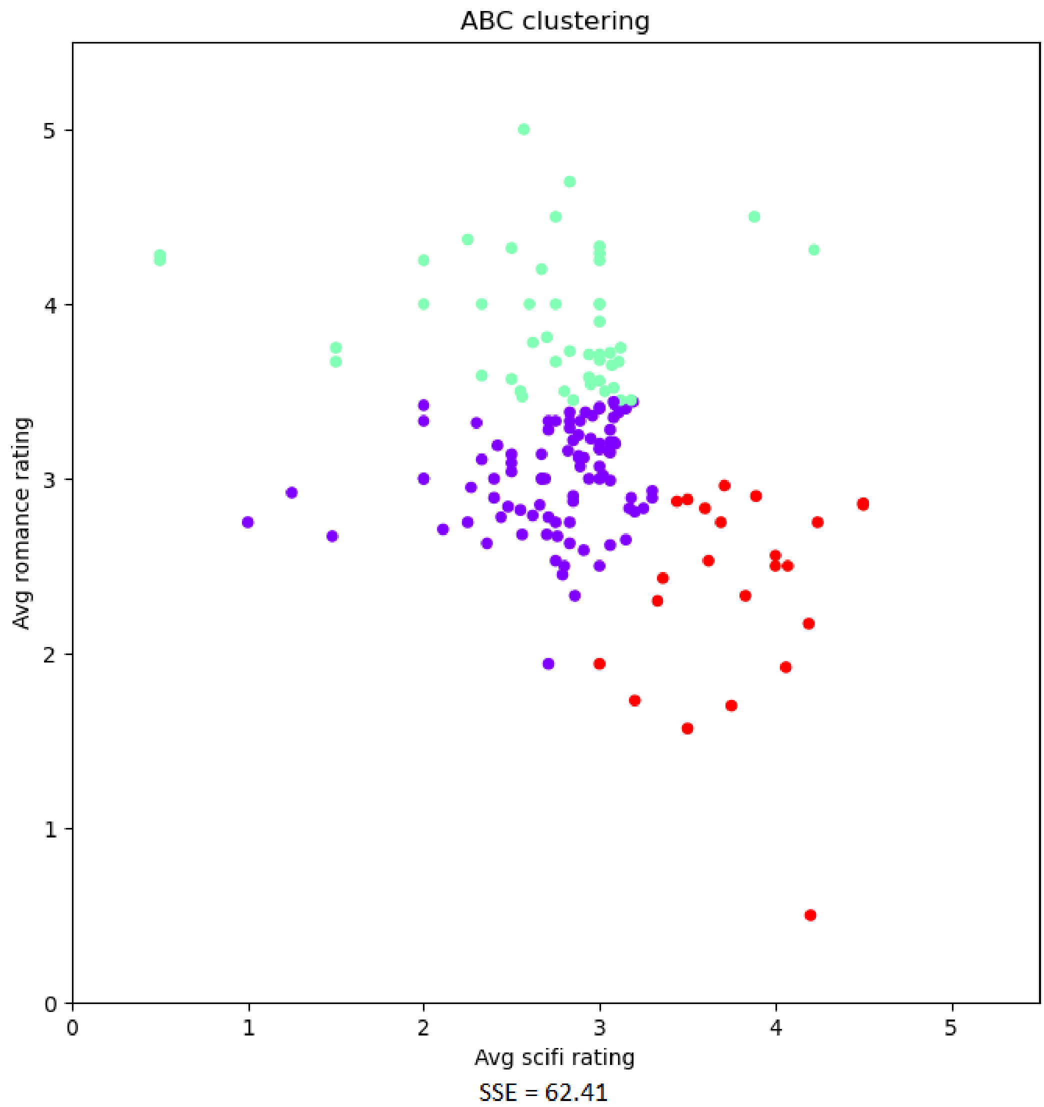

Figure 13: Data partitioning into 3 clusters with ABC

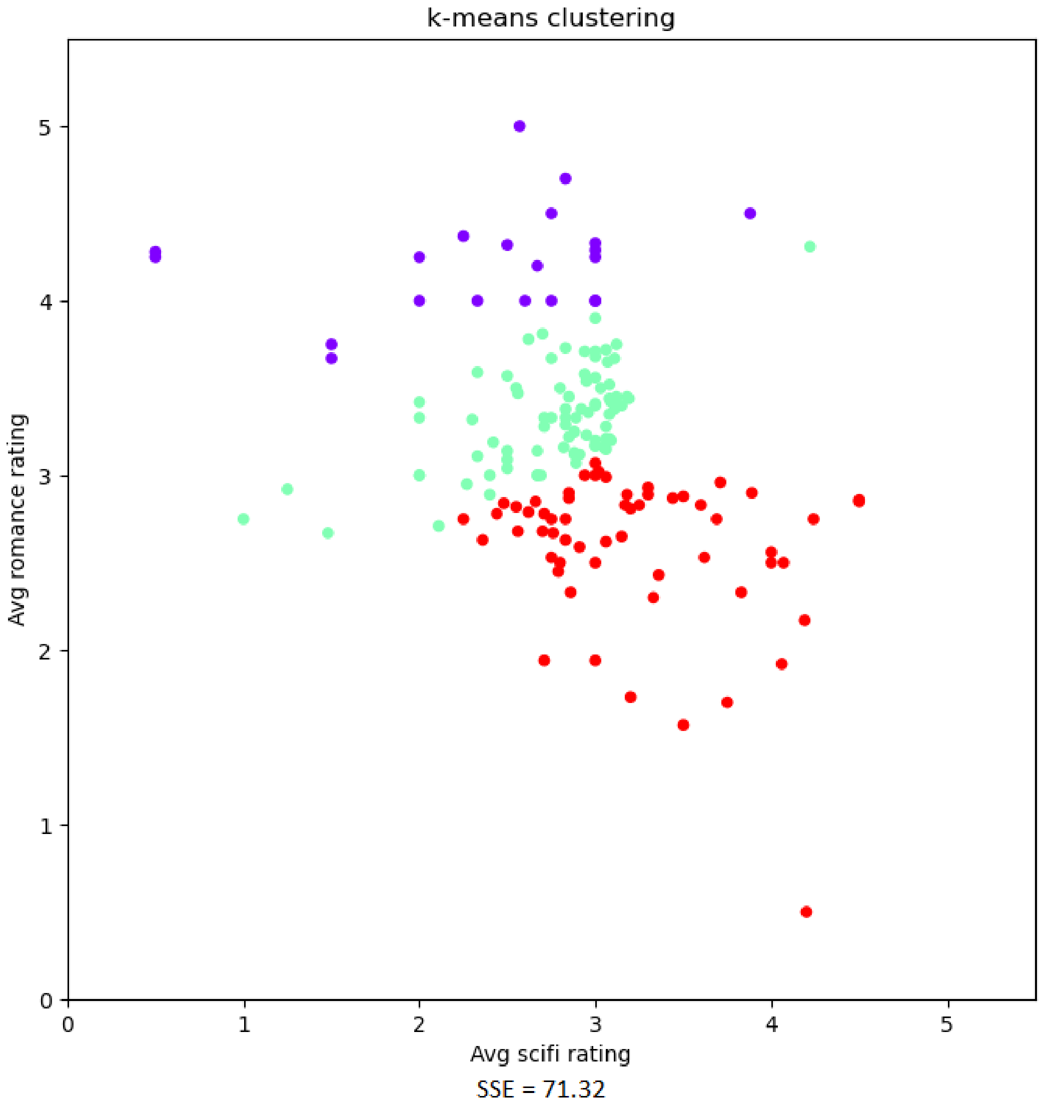

Figure 14: Data partitioning into 3 clusters with k-means – local optimum

Another clustering example, and the last one based on 2-dimensional user preferences (the average rating of romance and the average rating of sci-fi movies), is the partition of users into seven clusters. Fig. 15 and Fig. 16 show this partition formed by k-means and ABC, respectively. As in the previous examples, the clustering of both algorithms is close enough and ABC can overcome k-means algorithm in case bad initial center points are given as input to k-means.

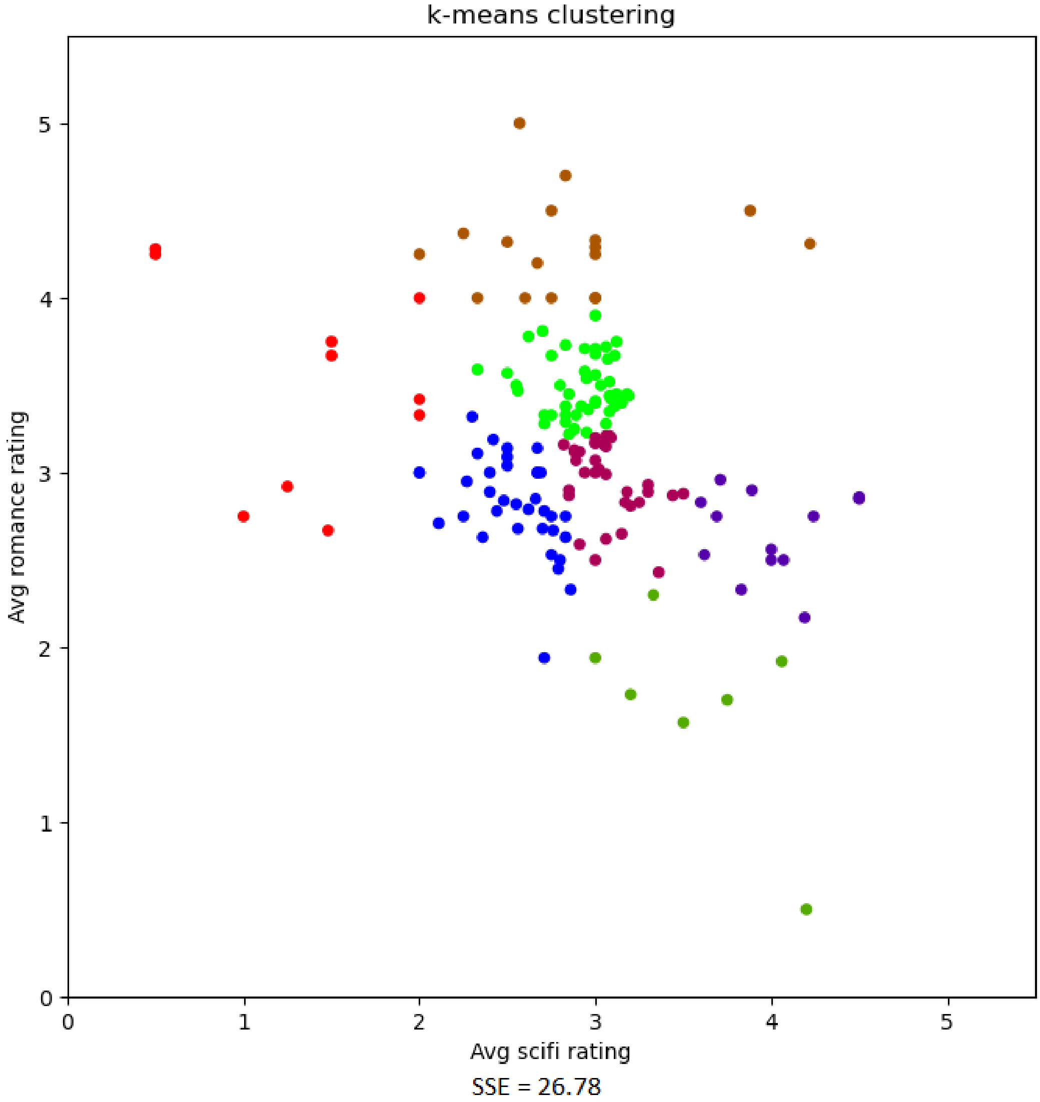

Figure 15: Data partitioning into 7 clusters with k-means

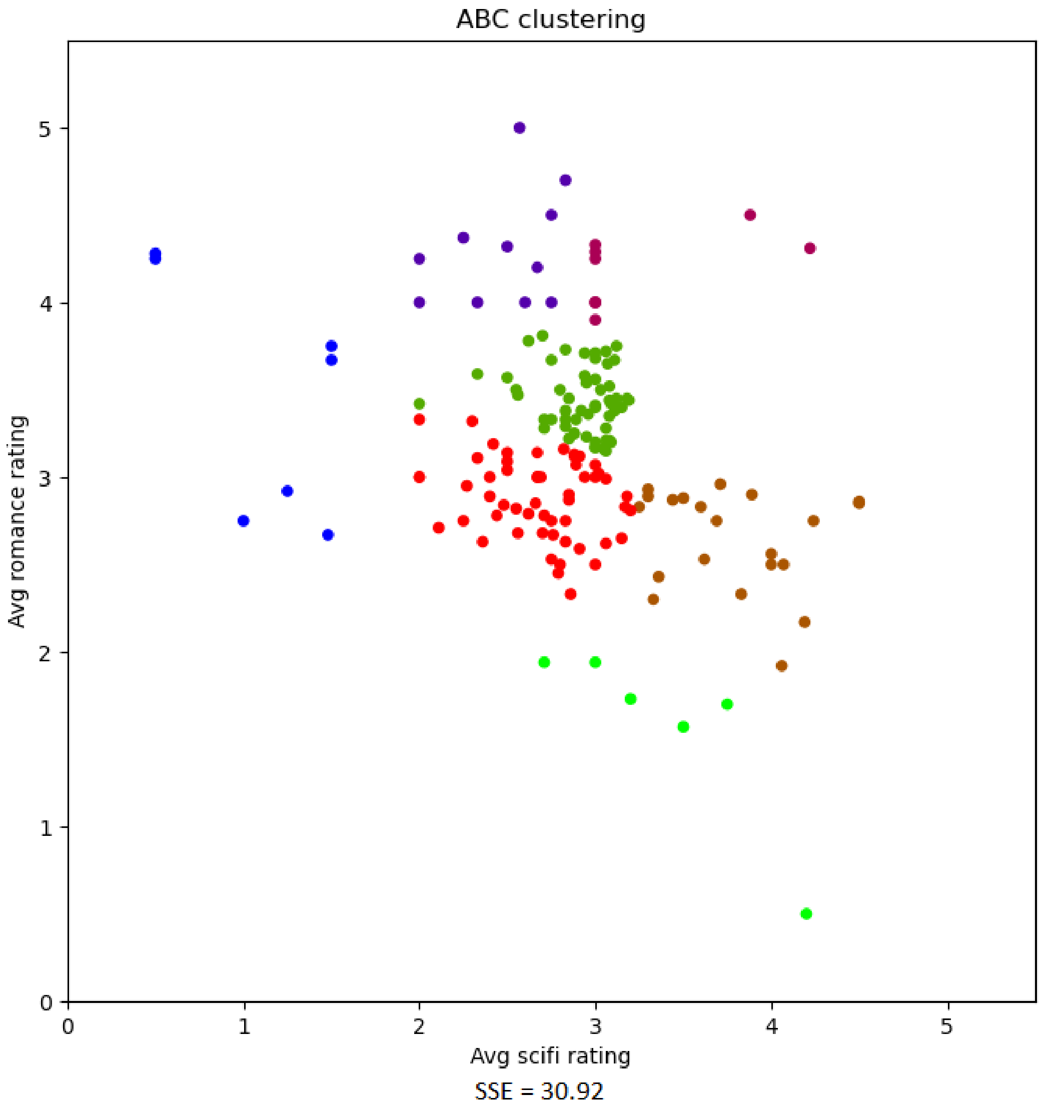

Figure 16: Data partitioning into 7 clusters with ABC

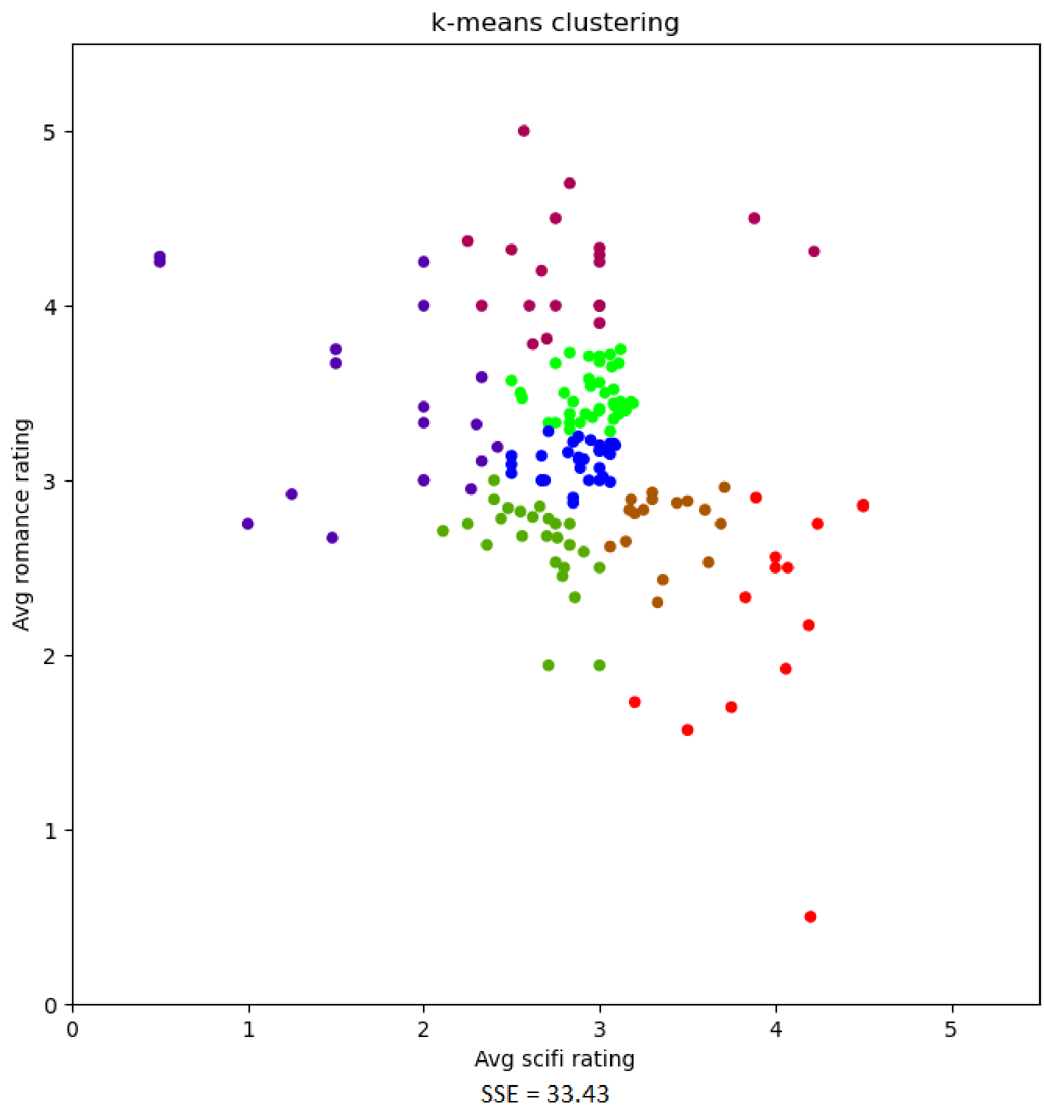

Figure 17: Data partitioning into 7 clusters with k-means – local optimum

Previous clustering examples illustrated how users rated romance and sci-fi movies. The two following figures show how the users can be grouped based on their average ratings of three movie genres, sci-fi, romance and action. Fig. 18 presents the clusters that were formed by the k-means while Fig. 19 presents the clusters that were formed by the ABC. The size of each point (user) in the search space suggests how much the user likes the action films. In other words, the large points indicate average ratings over 3 while small points the opposite. Even in a 3-dimensional space the resulting clusters of the ABC are quite similar with those of k-means and can be better when random initial centers negatively affect k-means (Fig. 20).

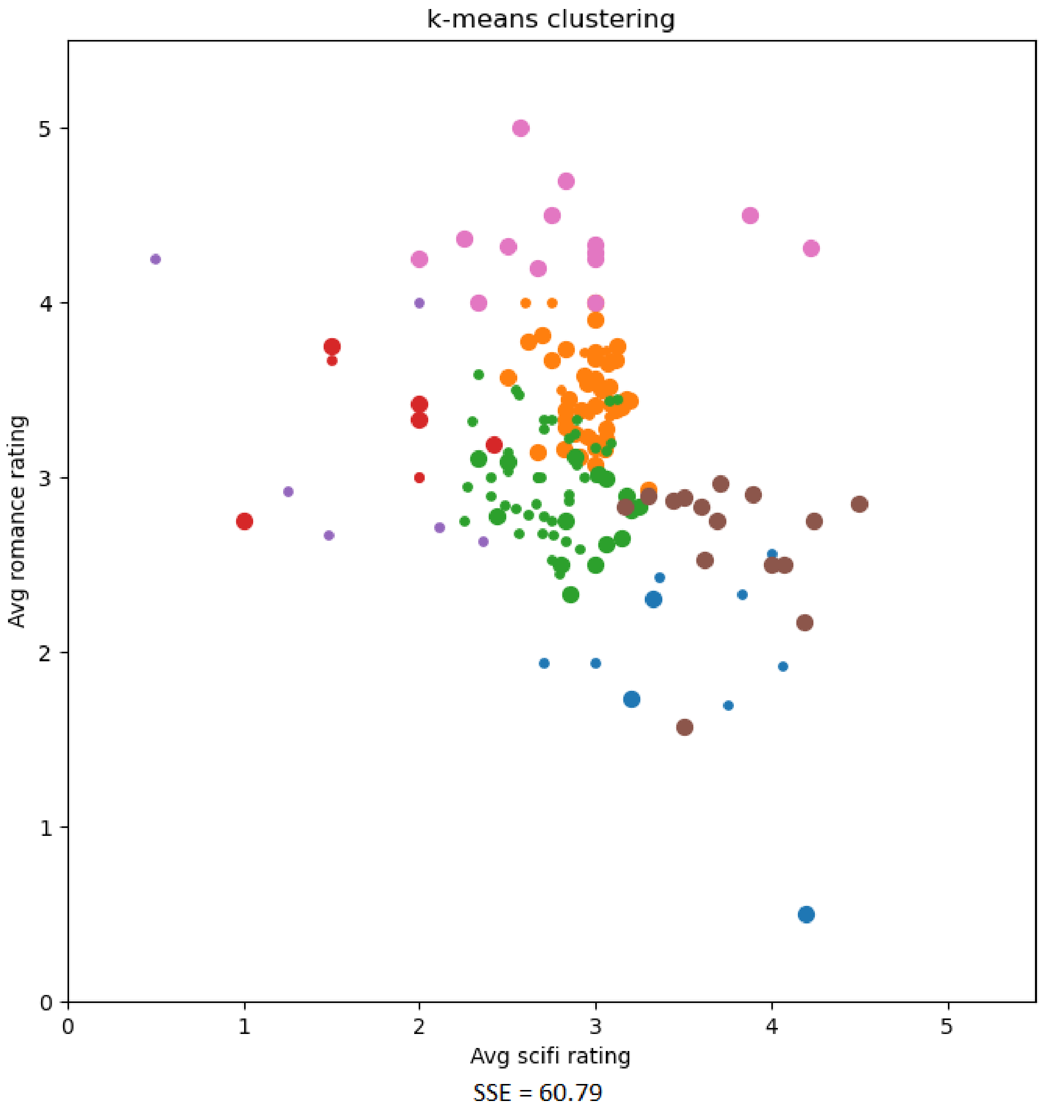

Figure 18: Data partitioning into 7 clusters with k-means in 3-dimensional space

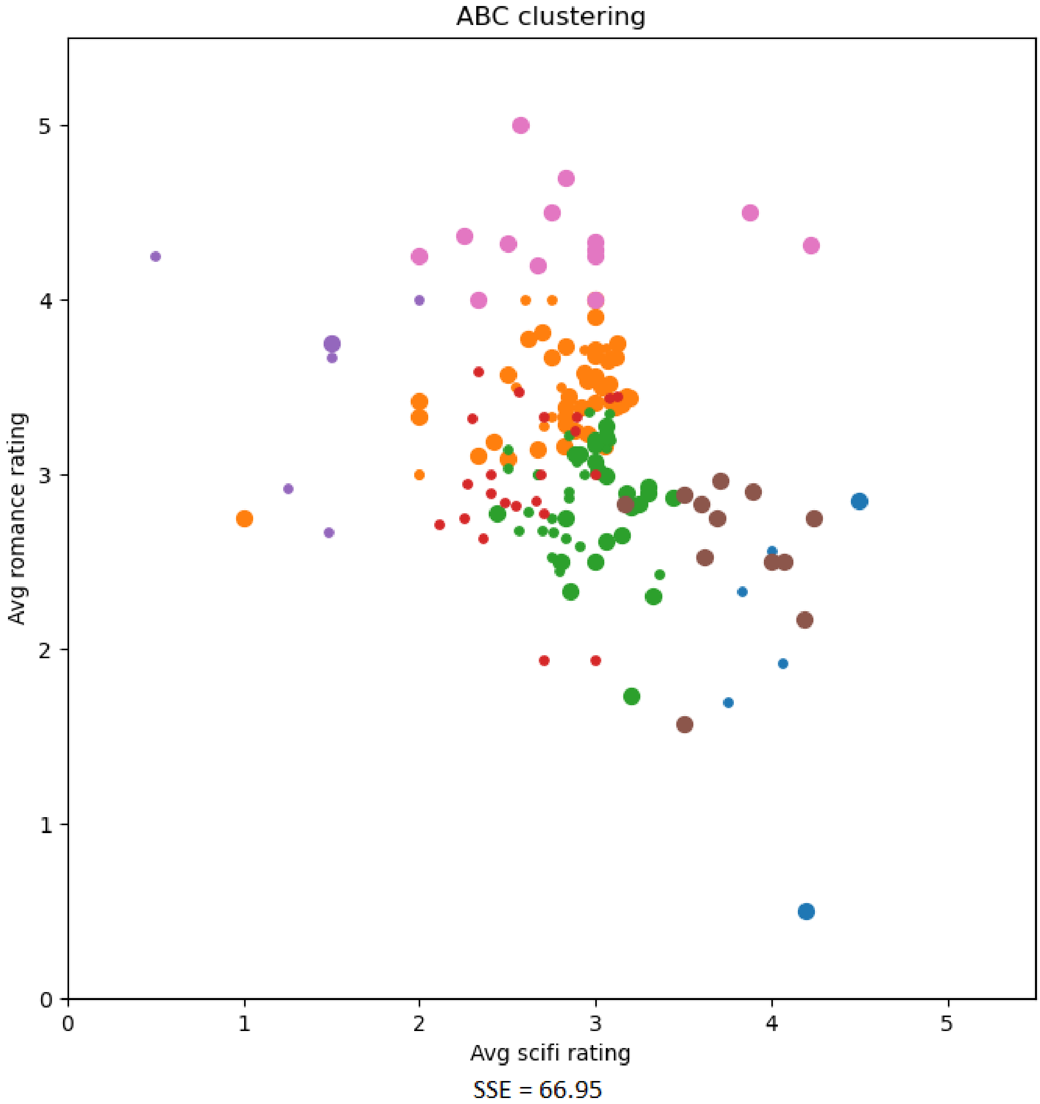

Figure 19: Data partitioning into 7 clusters with ABC in 3-dimensional space

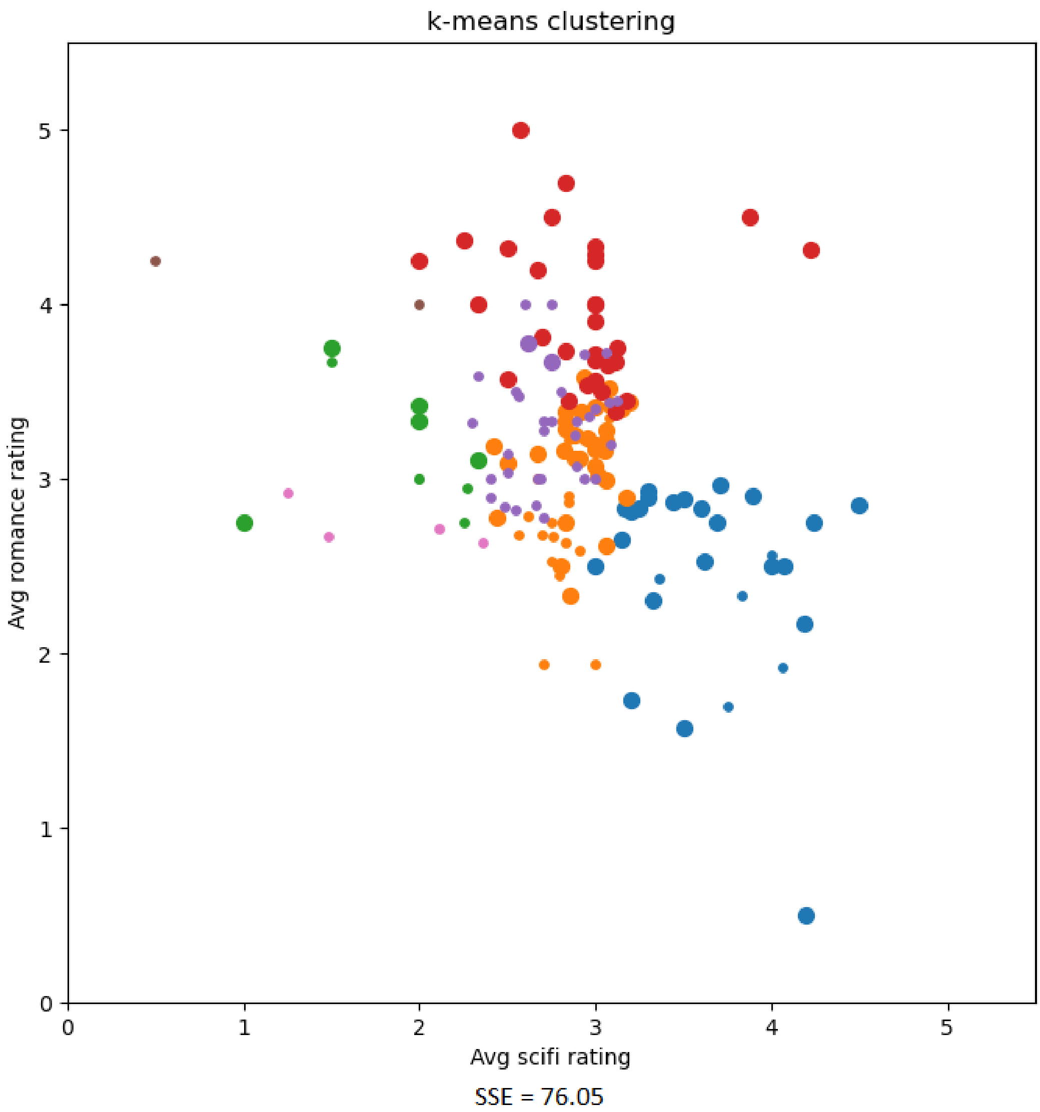

Figure 20: Data partitioning into 7 clusters with k-means in 3-dimensional space – local optimum

So far, it seems that ABC can form clusters just as well as k-means. But, can ABC form clusters based on more than 3 movie genres efficiently? Fig. 21 answers this question as it shows the value of the SSE metric for ABC and k-means for a different number of movie genres (bigger than 3). That is, for different dimensions of the vector that represents a user. As shown in the figure, ABC achieves quite the same results with k-means until 7 movie genres. As the number of movie genres increases, it is observed that k-means overcome the ABC in terms of SSE. So, the number of movie genres can significantly affect the performance of the algorithm, especially when the number of users to be grouped is relatively big, and therefore the number of clusters.

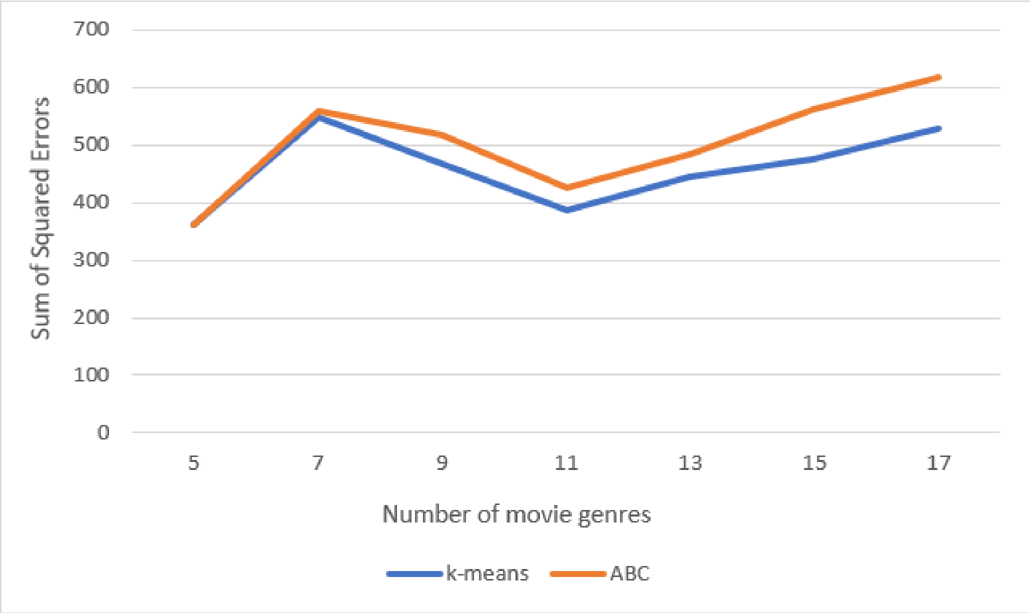

Figure 21: ABC and k-means clustering for different number of movies genres

From the results of the previous clustering tasks, it seems that ABC can achieve better results than k-means only in small clustering problems. That is, problems with a small number of users and user preferences. As an optimization algorithm, ABC explores the search space and evaluates several solutions to find the best one among them. In case there are many users to be clustered by many user preferences, ABC needs to find and therefore evaluate a huge number of solutions within a reasonable time limit. So, the amount of the solutions to be evaluated along with the time limitation can lead ABC to find a local optimum.

6. THE PROPOSED ALGORITHMS

As shown in the previous chapter, there are cases that k-means and SI techniques cannot form user groups efficiently on their own. This thesis proposes three collaborative filtering recommendation approaches that combine k-means with a different bio-inspired algorithm to form optimal clusters. Every hybrid approach aims at better recommendation quality by overcoming challenges that rise by using only one of its two algorithms. Each approach is described in detail in the following subsections.

The process of the proposed collaborative recommender systems is described in Fig. 22. Initially, a user-rating matrix is given to the proposed hybrid optimization algorithm, which finds the optimal initial cluster centers. These centers are given to k-means to produce the final clusters. Having the clusters and a target user, the nearest cluster to the target user can be determined. Then, the most suitable items for the user are found by calculating the weighted average of the neighbors' ratings of items. Finally, the top N items are recommended to the user.

6.1 ABC – k-means

The first algorithm proposed is an improved version of the algorithm presented in [16]. It uses K-means and ABC to optimize the initial clusters that eventually will be given to k-means to form the final clusters. The steps involved in the proposed hybrid algorithm are the following:

- Initialization phase
  - Initialize the population size, the number of clusters, the maximum number of iterations, the max number of cycles that a solution can remain unchanged (threshold limit), and the solution boundaries.
  - Generate the initial population (solutions). Each solution (a set with k centroids) is a randomly chosen user of the users to be clustered.
  - Evaluate the initial solutions and find the best one. For the evaluation of each solution, use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution. Replace the given solution with the produced final clusters.
- Employed bees' phase
  - Produce new solutions for the employed bees using different solutions of the population.
  - Evaluate the new solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution. Replace the given solution with the produced final clusters.
  - If a new solution is better than a current solution, replace the current solution with the new one, otherwise increase its trials (cycles that the solution remains unchanged) by one.
- Onlooker bees' phase
  - Assign onlooker bees to employed bees according to probabilities and therefore produce new solutions using different solutions of the population.

  - Evaluate the new solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution. Do not replace the given solution with the produced final clusters.
  - If a new solution is better than a current solution, replace the current solution with the new one, otherwise increase its trials by one.
- Scout bees' phase
  - Find the solutions that their trials reached the threshold limit. Replace each solution with a new solution and set its trials to zero.
  - Evaluate the new solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution. Do not replace the given solution with the produced final clusters.
  - If a new solution is better than a current solution, replace the current solution with the new one.
- Final phase
  - If the max number of iterations is reached, find the solution with the lowest cost, that is the best solution. This solution will be given to k-means to produce the final clusters.

The four improvements made to produce better recommendations are presented below. The first three improvements concern the proposed hybrid algorithm, while the last one the way in which the predictions are calculated.

1. In the initialization phase the population is not created randomly using the given boundaries. Instead, the initial solutions are generated by using the users to be clustered. That is, a solution is a set of randomly chosen users. The number of the users is equal to the number of clusters. In other words, each user in the solution is a cluster centroid.
2. A new solution is created by changing the entire old solution and not just one of its dimensions. This enhances the exploration process
3. A solution is evaluated by calculating the SSE of the clusters that are produced when the solution is fed to the k-means. In the initialization and in the employed bees' phase the produced clusters replace the initial solution. Thus, it is investigated whether a solution can be further improved. In the onlooker and scout bees' phases the solution to be evaluated is not changed to avoid being trapped in a local optimum.
4. In the process of recommendation, various prediction ratings for various items are calculated to find the most suitable items for the target user. The prediction rating of an unrated item is calculated based on the weighted average of the neighbors' ratings of items. The weights are produced by calculating the Pearson correlation of the target user with all its neighbors.

6.2 CSO – k-means

The second hybrid algorithm that this thesis proposes combines the k-means algorithm with the cuckoo search optimization (CSO) technique. CSO is one of the many nature-inspired algorithms developed based on the reproduction of cuckoo birds. This algorithm was introduced in 2009 by Yang and Deb [40]. It is inspired by the brood parasitism of cuckoo species and the characteristics of Lévy flights. Cuckoos lay their eggs in the nests of other host birds, hoping that their eggs will be raised by birds of other species. Each egg in a nest represents a solution, and the aim is to find better solutions to replace the not so good solutions in the nests. Cuckoo search optimization algorithm is based on three basic rules:

- Each cuckoo lays one egg at each iteration and selects a nest randomly to lay its egg in it.
- The best nests with high quality of eggs are carried forward to the next generation.
- For a fixed number of nests, a host cuckoo can discover that the egg in its nest does not belong to it and with a probability $p_a \epsilon \ (0, 1)$, the host cuckoo can either throw the egg away or abandon the nest and build a new one somewhere else.

The algorithm can be extended to more complicated cases where each nest can have multiple eggs that is, multiple solutions. However, in this thesis the simple version of the algorithm is used.

The steps of the proposed algorithm that combines CSO and the k-means algorithm are the following:

- Initialization phase
    - Initialize the population size (number of nests), the solution boundaries, the max number of iterations and the probability of a cuckoo's egg detection.
    - Generate an initial population of the host nests (solutions) randomly. Each solution (a set with k centroids) is a randomly chosen user of the users to be clustered.
    - Evaluate the initial solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution.
- Iteration phase
    - Generate new solutions by performing Lévy flights but keep the current best.
    - Evaluate the new solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution.
    - Replace the bad solutions in the nests with better solutions and find the best solution so far.
    - Using the probability of a cuckoo's egg detection, empty some nests and fill them with new solutions.
    - Evaluate the new solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution.
    - Replace the bad solutions in the nests with better solutions and find the best solution so far.
- Final phase

- If the max number of iterations is reached, find the solution with the lowest cost, that is the best solution. This solution will be given to k-means to produce the final clusters.

The first and the fourth improvements applied to the proposed ABC – k-means algorithm were applied to this algorithm as well.

6.3 GWO – k-means

The final hybrid recommender system that this thesis proposes combines the k-means algorithm with the grey wolf optimizer. The GWO mimics the hierarchical order and hunting mechanism of grey wolves in nature. Grey wolves mostly prefer to live in a pack. Each pack consists of four groups, alphas, betas, deltas, and omegas. The alphas are the leaders of the pack and are responsible for making decisions about various issues such as hunting. The betas are subordinate wolves that help the alpha in decision-making. The lowest level in the hierarchy of grey wolves is omega. The omegas are accountable to all the other groups and they are the last wolves that are allowed to eat. Between the omegas and the betas, there are the deltas. In this group belong the scouts, the sentinels, the elders, the caretakers, and the hunters.

The GWO algorithm is based on the hunting behavior of alphas, betas, deltas, and omegas. Each solution represents a wolf, and therefore each solution belongs to a hierarchical category. The best solution is considered as the alpha. The second and the third best solutions are the beta and the delta, respectively. The rest of the candidate solutions belong to the omega category. Based on the above, the steps of the algorithm can be described as follows:

- Initialization phase
  - Initialize the number of wolves (solutions), the solution boundaries, and the max number of iterations.
  - Generate the initial solutions randomly. Each solution (a set with k centroids) is a randomly chosen user of the users to be clustered.
  - Evaluate the initial solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution.
  - Initialize the alpha, beta, and delta solutions. That is, the three best solutions, respectively.
- Iteration phase
  - Update each solution by using the alpha, beta, and delta solutions as they have better knowledge about the potential location of the best solution.
  - Evaluate the new solutions. For the evaluation of each solution use k-means algorithm. Give each solution as the initial clusters to k-means and set the SSE of the final clusters as the cost of the solution.
  - Update the alpha, beta, and delta solutions. That is, the three best solutions, respectively.
- Final phase

- If the max number of iterations is reached, find the solution with the lowest cost (alpha), that is the best solution. This solution will be given to k-means to produce the final clusters.

The improvements in the initialization phase and in the recommendation process that were applied to the proposed ABC – k-means hybrid algorithm, were applied to this algorithm as well.

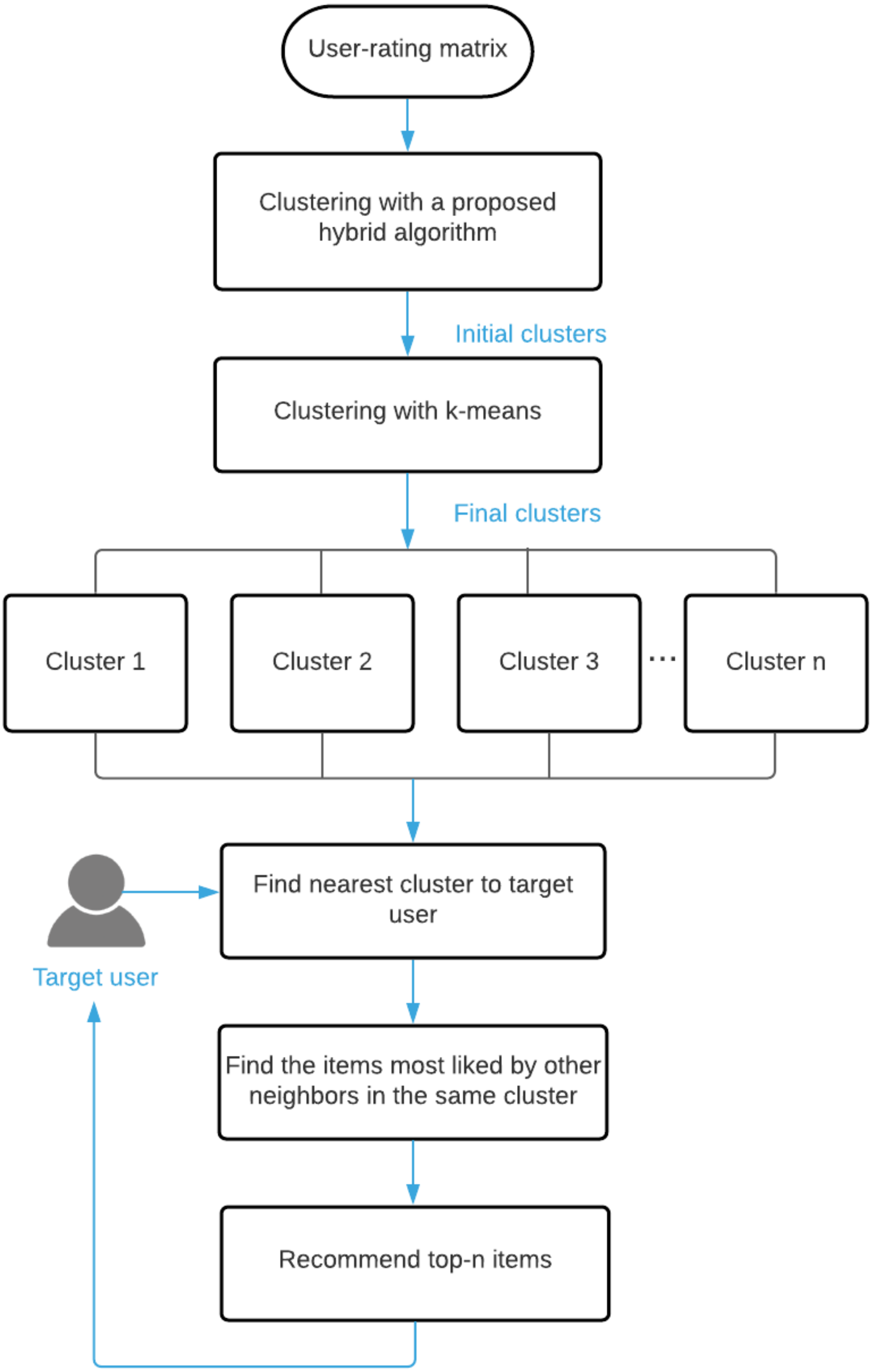

Figure 22: Overview of the proposed collaborative recommender system

7. EXPERIMENTS AND RESUTLS

7.1 Datasets

The MovieLens100k dataset was selected for the evaluation of the proposed recommender systems. This dataset was collected at the University of Minnesota by the Group lens Research project team. It includes 100,000 ratings from 943 users on 1682 movies. Ratings are made on a 5-star scale, with one-star increments (1 star – 5 stars). In this dataset, each user has rated at least 20 movies, and each movie has been rated at least once. The dataset was divided into 80% as the training set and 20% as the testing set. The training data were used to cluster the users, while the testing data to make predictions, and therefore recommendations to the users.

7.2 Evaluation criteria

The quality of the proposed recommendation methods was measured using Mean Absolute Error (MAE), precision, SSE, and recall.

- Mean Absolute Error: This statistical measure calculates the difference between the predicted ratings and actual ratings of users as shown in (4). The lower the value of the metric, the better the predictions made by the recommender system.

$$MAE = \frac{\sum |p_{ij} - a_{ij}|}{N}$$

  Where p_{ij} is the predicted rating value for user i on item j, N is the total number of predicted items, and a_{ij} is the real rating of user i on item j.

- Precision: Precision is defined as the fraction of number of items thar are good recommendations for the user (true positives) to the total number of recommendations (true positives + false positives).

$$precision = \frac{true\ positives}{true\ positives + false\ positves}$$

- Recall: Recall is defined as the fraction of number of items that are relevant (true positives) to the total number of items that is actually considered as relevant (true positives + false negatives).

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

7.3 Performance results

This section presents the evaluation of the proposed algorithms in terms of MAE, precision, recall, and SSE. The proposed methods are compared to the one proposed in [16] and to existing clustering techniques such as k-means, PCA-GAKM, PCA-KM, UPCC, and SOM.

The evaluation metrics were calculated for a different number of clusters. In Fig. 23 the comparison of the proposed algorithms with the existing clustering-based CF methods [26] regarding MAE (see Table 1) is presented. The MAE values of PCA-GAKM, PCA-KM, UPCC, SOM, PCA-SOM, and GAKM, on the MovieLens 100k dataset, are retrieved from the literature. As seen in the figure, the proposed hybrid algorithms, ABC-KM, CSO-KM, and GWO-KM have a Mean Absolute Error equal to 0.767, 0.769, 0.768, respectively when the number of clusters is 25. On the other hand, existing methods have MAE above 0.78. It seems that as the number of clusters increases, the proposed algorithms perform better compared to the other algorithms. Considering that the dataset is extremely sparse, only 6.3% of the user-item ratings have a value, even a seemingly small improvement is significant. In addition, it seems that among the three proposed methods, the ABC-KM approach gives slightly better or the same MAE.

Fig. 24 shows that the proposed ABC-KM algorithm provides a slightly higher precision rate than the k-means and the hybrid algorithm of [16]. The proposed approach achieves maximum precision value equal to 0.723 when the number of clusters is 35. Moreover, the other two proposed approaches have quite similar precision rates with the proposed ABC-KM, and therefore higher values form the other already existing algorithms.

The Recall of the proposed methods is presented in Fig. 25. This figure shows that the proposed algorithms provide greater recall value according to a different number of clusters. The maximum Recall is achieved by the proposed ABC-KM approach for 35 clusters and is equal to 0.778. The Recall achieved by the GWO-KM and the CSO-KM algorithms, is very close to that of ABC-KM. The other two methods (k-means and ABC-KM[16]) achieve less precision than the proposed methods.

Precision and recall are binary measures used to evaluate models with binary output. To compute these two measures, for the evaluation of the proposed algorithms, the conversion of the numerical ratings (1 to 5) into binary (relevant and irrelevant items) was needed. The conversion was made by assuming that any rating above 3.5 corresponds to a relevant item and any rating below 3.5 is irrelevant. This means that if the predicted rating and the actual rating are above 3.5, the prediction is a true positive. If the predicted rating and the actual rating are below 3.5, the prediction is a true negative. If the predicted rating is above 3.5 and the actual rating is below 3.5, the prediction is a false positive. In another case, the prediction is a false negative. The high value of the two metrics is due to the way they are calculated. Normally, the high MAE indicates low precision and recall. However, as already mentioned, the actual and the predicted ratings are considered equal if they fall in the same range (below or above 3.5) and not if they have the same value.

Fig. 26 illustrates the strong connection between the effectiveness of a recommender system and the quality of the formed clusters. The previous figures show that the proposed algorithms perform better than k-means in terms of MAE, precision, and recall. This happens because the proposed algorithms form better clusters than k-means. Looking at Fig. 26, the proposed algorithms attain lower SSE values than the traditional k-means across a different number of clusters.

Table 1: MAE for different approaches

System/cluster	5	10	15	20	25	30	35	40
PCA-GAKM	0.790	0.770	0.770	0.780	0.781	0.785	0.786	0.788
PCA-SOM	0.820	0.790	0.790	0.790	0.800	0.805	0.806	0.807
SOM	0.819	0.810	0.810	0.810	0.810	0.805	0.810	0.810
UPCC	0.825	0.825	0.824	0.828	0.824	0.824	0.825	0.825

k-means	0.818	0.816	0.815	0.813	0.812	0.812	0.810	0.800
PCA-KM	0.850	0.845	0.841	0.840	0.840	0.840	0.840	0.840
GAKM	0.815	0.805	0.804	0.804	0.804	0.804	0.803	0.803
ABC-KM [16]	0.773	0.764	0.764	0.771	0.780	0.784	0.782	0.787
ABC-KM Proposed	0.773	0.768	0.765	0.767	0.767	0.766	0.765	0.767
CSO-KM	0.779	0.769	0.769	0.771	0.769	0.773	0.770	0.772
GWO-KM	0.779	0.768	0.768	0.772	0.768	0.772	0.767	0.767

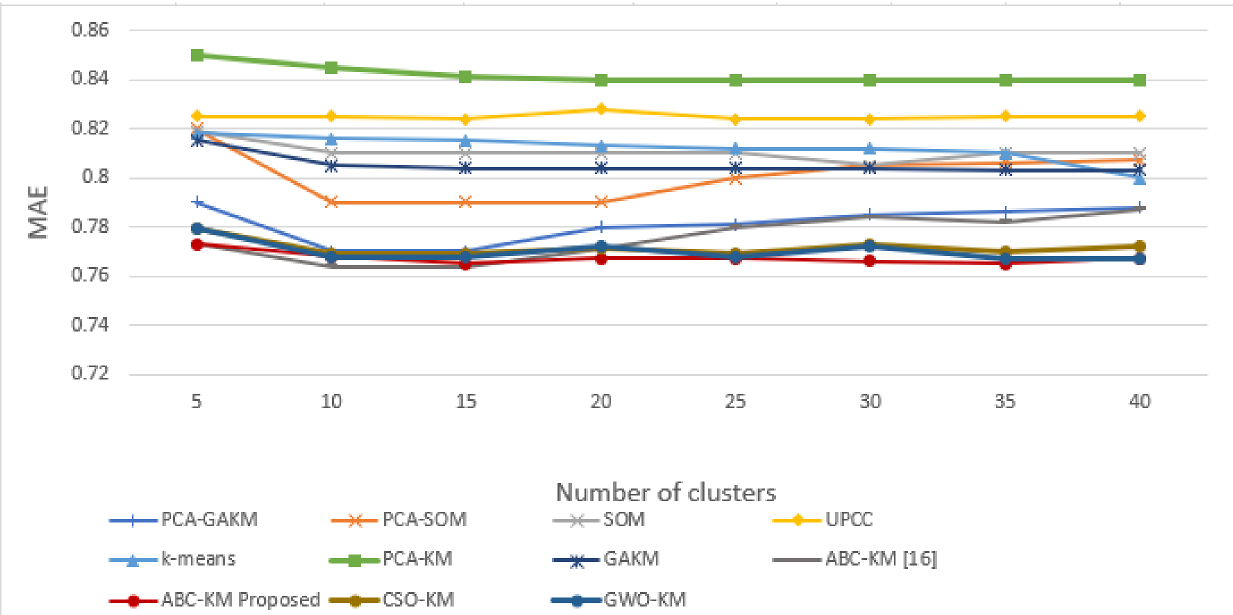

Figure 23: MAE for different approaches

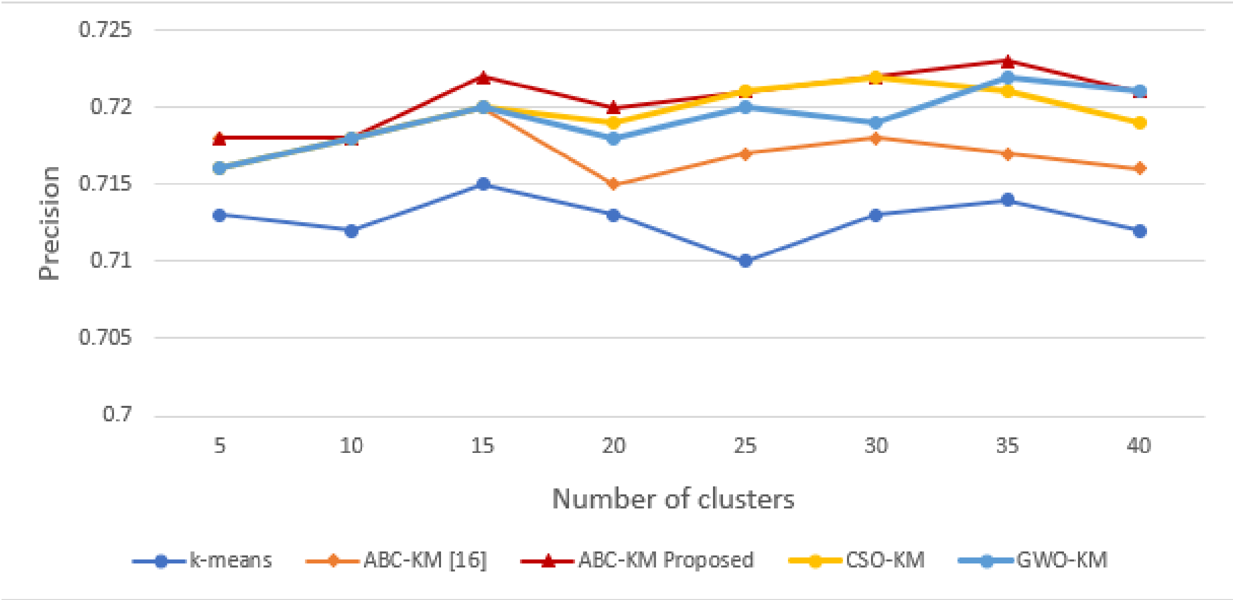

Figure 24: Precision for different approaches

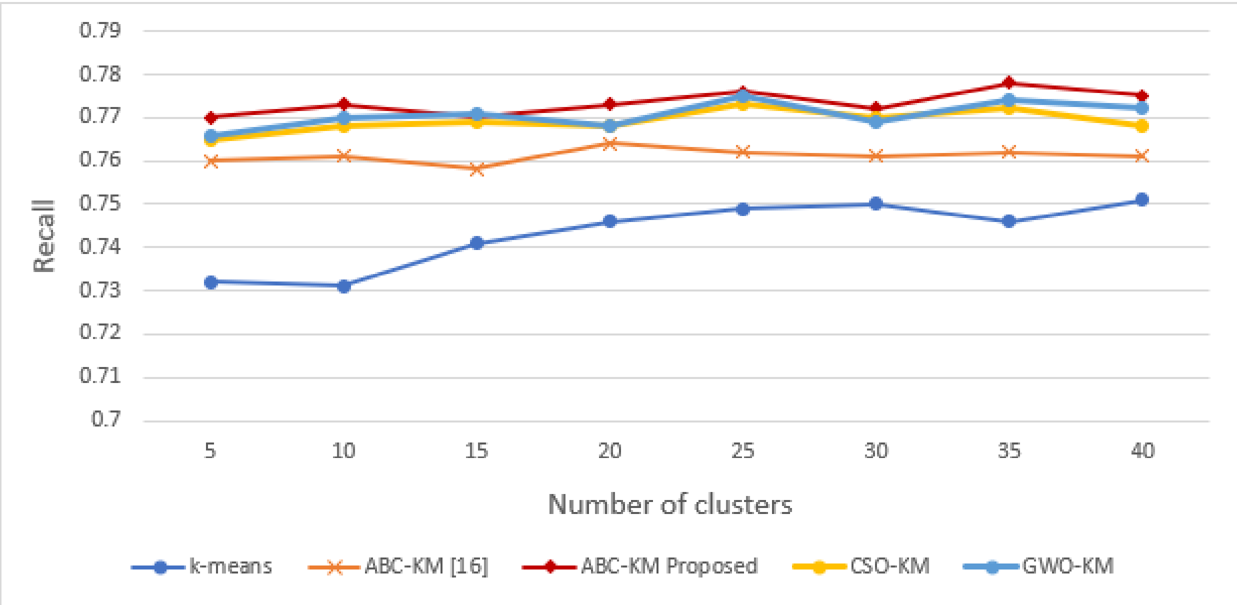

Figure 25: Recall for different approaches

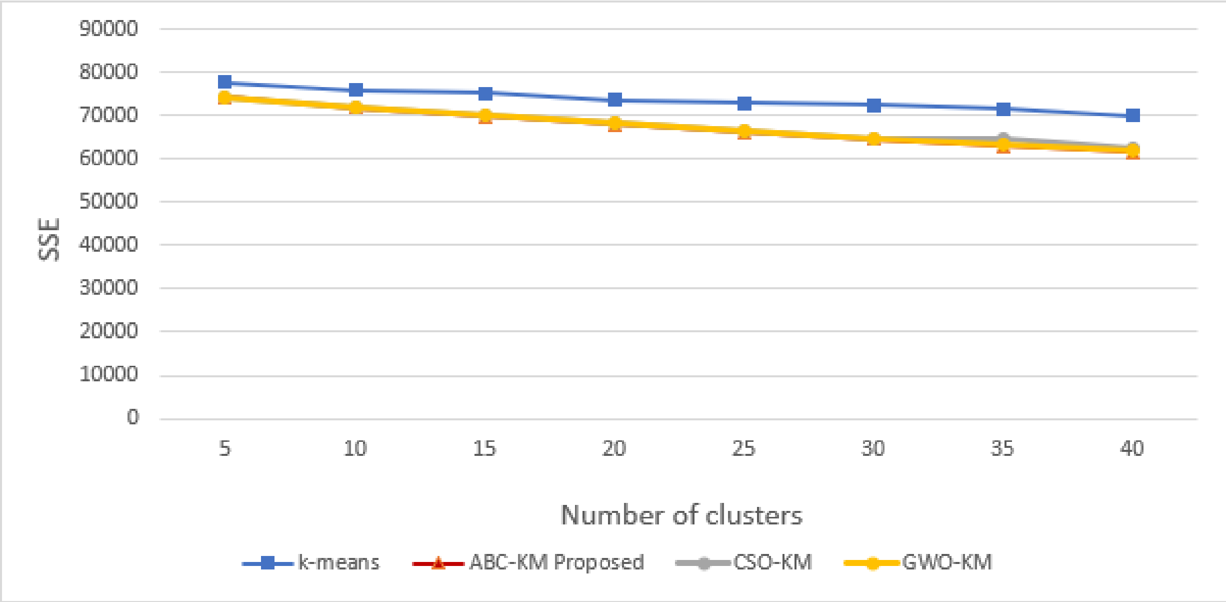

Figure 26: SSE for different approaches

8. CONCLUSIONS AND FUTURE WORK

This thesis presents three swarm-based hybrid collaborative movie recommender systems. Each recommender system combines the k-means algorithm with a different swarm intelligence technique (ABC, CSO, GWO). The proposed algorithms are hybrid models that form clusters efficiently by reducing the scalability and data sparsity problems. The approach that combines the k-means algorithm and the bio-inspired artificial bee colony method is an improved version of the approach described in [16]. Each proposed method produces the optimal initial clusters' centers that will be fed again to k-means to produce the final clusters. Moreover, during the recommendation process, the prediction ratings for a user are calculated based on the Pearson similarity that the user has with his neighbors. The performance of the proposed techniques is measured on a MovieLens dataset using various metrics such as MAE, SSE, recall, and precision. The evaluation results indicate that the proposed recommender systems offer better recommendations than the existing techniques.

For future work, a new hybrid recommender system that combines the k-means algorithm with the new population-based algorithm, named Sonar Inspired Optimization (SIO), could be developed. The SIO algorithm, proposed by Tzanetos A., Dounias G. (2017), is a new swarm intelligence algorithm that has never been used to support recommender systems in the clustering problem. Moreover, the proposed algorithms could be changed in a way to use the fuzzy c-means algorithm to find the optimal final clusters. Finally, another challenge is the use of other important characteristics of users, along with ratings, for more accurate and reliable predictions.

ABBREVIATIONS - ACRONYMS

RS	Recommender System
RSs	Recommender Systems
ABC	Artificial Bee Colony
MAE	Mean Absolute Error
SSE	Sum of Squared Errors
ML	Machine Learning
CB	Content-Based
CF	Collaborative Filtering
SI	Swam Intelligence
ACO	Ant Colony Optimization
ACS	Ant Colony System
AS	Ant System
STARS	Semantic-enhanced Trust based Ant Recommender System
BA	Bat Algorithm
PSO	Particle Swarm Optimization
CSO	Cuckoo Search Optimization
GSO	Glowworm Swarm Optimization
AFSO	Artificial Fish Swarm Optimization
RMSE	Root Mean Squared Error
MCRSs	Multi-Criteria Recommender Systems
GSA	Gravitational Search Algorithm
CARS	Context-Aware Recommendation System
DCW	Differential Context Weighting
GWO	Gray Wolf Optimizer
FCM	Fuzzy C-Means
PCF	Pearson correlation based Collaborative Filtering
FCF	Fuzzy Collaborative Filtering
FG-CF	Fuzzy Genetic Algorithm based Collaborative Filtering
FPSO-CF	Fuzzy Particle Swarm Optimization based Collaborative Filtering
SVM	Support Vector Machine
ODP	Open Directory Project
AAA	Artificial Algae Algorithm

QPSO	Quantum-behaved Particle Swarm Optimization
LOD	Linked Open Data
GA	Genetic Algorithm
MMMF	Maximum Margin Matrix Factorization
MF	Matrix Factorization
SIO	Sonar Inspired Optimization
CHSO	Chicken Swarm Optimization

REFERENCES

[1] F.O.Isinkayea, .O.Folajimi, B.A.Ojokoh, Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal Volume 16, Issue 3, pp. 261-273, November 2015, https://doi.org/10.1016/j.eij.2015.06.005.

[2] Gastón Rodríguez, “Introduction to Recommender Systems in 2019”, https://tryolabs.com/blog/introduction-to-recommender-systems.

[3] Yagnesh G. patel, Vishal P.Patel, A Survey on Various Techniques of Recommendation System in Web Mining, IJEDR Volume 3, Issue 4, 2015, https://www.ijedr.org/papers/IJEDR1504119.pdf.

[4] Ladislav Peška, Tsegaye Misikir Tashu, Tomáš Horváth, Swarm intelligence techniques in recommender systems - A review of recent research, Swarm an Evolutionary Computation 48, pp. 201-219, August 2019, https://doi.org/10.1016/j.swevo.2019.04.003.

[5] P. Choudhary, V. Kant, P. Dwivedi, Handling natural noise in multi criteria recommender system utilizing effective similarity measure and particle swarm optimization, Procedia Comput. Sci. 115 (2017) 853–862, https://doi.org/10.1016/j.procs.2017.09.168.

[6] V. Choudhary, D. Mullick, S. Nagpal, Gravitational search algorithm in recommendation systems, in: Advances in Swarm Intelligence, Springer, 2017, pp. 597–607, https://doi.org/10.1007/978-3-319-61833-3_63.

[7] A. Gupta, K. Gusain, Selection of Similarity Function for Context-Aware Recommendation Systems, Springer, Singapore, 2017, pp. 803–811, https://doi.org/10.1007/978-981-10-3874-7_76.

[8] M. Hamada, M. Hassan, Artificial neural networks and particle swarm optimization algorithms for preference prediction in multi-criteria recommender systems, Informatics 5 (2) (2018), https://doi.org/10.3390/informatics5020025.

[9] R. Katarya, O.P. Verma, Recommender system with grey wolf optimizer and fcm, Neural Comput. Appl. 30 (5) (2018) 1679–1687, https://doi.org/10.1007/s00521-016-2817-3.

[10] S. Tomer, S. Nagpal, S.K. Bindra, V. Goel, Fuzzy gravitational search approach to a hybrid data model based recommender system, in: Knowledge Science, Engineering and Management, Springer International Publishing, 2018, pp.337–348, https://doi.org/10.1007/978-3-319-99365-2_30.

[11] X.Wang, J.Wen, F. Luo, W. Zhou, H. Ren, Personalized Recommendation System Based on Support Vector Machine and Particle Swarm Optimization, Springer, 2015, pp. 489–495, https://doi.org/10.1007/978-3-319-25159-2_44.

[12] S. Yang, Q. Peng, L. Chen, The BPSO Based Complex Splitting of Context-Aware Recommendation, Springer, 2016, pp. 435–444, https://doi.org/10.1007/978- 981-10-0356-1_46.

[13] S. Yadav, Vikesh, Shreyam, S. Nagpal, An improved collaborative filtering based recommender system using bat algorithm, Procedia Comput. Sci. 132 (2018) 1795–1803, https://doi.org/10.1016/j.procs.2018.05.155.

[14] E. Bastami, A. Mahabadi, E. Taghizadeh, A gravitation-based link prediction approach in social networks, Swarm Evol. Comput. (2018), https://doi.org/10.1016/j.swevo.2018.03.001.

[15] S. Ganesan, S. Selvaraju, Application of Particle Swarm Optimization and User Clustering in Web Search, Springer, 2015, pp. 427–437, https://doi.org/10.1007/978-81-322-2208-8_39.

[16] R. Katarya, Movie recommender system with metaheuristic artificial bee, Neural Comput. Appl. 30 (6) (2018) 1983–1990, https://doi.org/10.1007/s00521-017-3338-4.

[17] R. Katarya, O.P. Verma, A collaborative recommender system enhanced with particle swarm optimization technique, Multimed. Tool. Appl. 75 (15) (2016) 9225–9239, https://doi.org/10.1007/s11042-016-3481-4.

[18] R. Katarya, O.P. Verma, Effectual recommendations using artificial algae algorithm and fuzzy c-mean, Swarm Evol. Comput. 36 (2017) 52–61. https://doi.org/10.1016/j.swevo.2017.04.004.

[19] V. Vellaichamy, V. Kalimuthu, Hybrid collaborative movie recommender system using clustering and bat optimization, Int. J. Intell. Eng. Syst. 10 (5) (2017), https://doi.org/10.22266/ijies2017.1031.05.

[20] R. Logesh, V. Subramaniyaswamy, V. Vijayakumar, X.Z. Gao, V. Indragandhi, A hybrid quantum-induced swarm intelligence clustering for the urban trip recommendation in smart city, Future Gener. Comput. Syst. 83 (2018) 653–673, https://doi.org/10.1016/j.future.2017.08.060.

[21] D. Alathel, Ant Colony Inspired Models for Trust-Based Recommendations, Ph.D. thesis, School of Engineering and Applied Sciences of The George Washington University, 2015, https://doi.org/10.1016/j.eswa.2011.07.124.

[22] S. Beldjoudi, H. Seridi, N.E.I. Karabadji, Recommendation in collaborative e-learning by using linked open data and ant colony optimization, in: Intelligent Tutoring Systems, Springer International Publishing, Cham, 2018, pp. 23–32, https://doi.org/10.1007/978-3-319-91464-0_3.

[23] F.S. Gohari, H. Haghighi, F.S. Aliee, A semantic-enhanced trust based recommender system using ant colony optimization, Appl. Intell. 46 (2) (2017) 328–364, https://doi.org/10.1007/s10489-016-0830-y.

[24] F. Rehman, O. Khalid, N.U. Haq, A.U.R. Khan, K. Bilal, S.A. Madani, Diet-right: a smart food recommendation system, KSII Trans. Int. Inf. Syst. (2017), https://doi.org/10.3837/tiis.2017.06.006.
[25] E. Sherkat, M. Rahgozar, M. Asadpour, Structural link prediction based on ant colony approach in social networks, Phys. A 419 (2015) 80–94, https://doi.org/10.1016/j.physa.2014.10.011.
[26] L. Xing, Q. Ma, L. Jiang, Microblog user recommendation based on particle swarm optimization, China Commun. 14 (5) (2017) 134–144, https://doi.org/10.1109/CC.2017.7942321.
[27] R. Katarya, O.P. Verma, Efficient music recommender system using context graph and particle swarm, Multimed. Tool. Appl. 77 (2) (2018) 2673–2687, https://doi.org/10.1007/s11042-017-4447-x.
[28] V.R. Chifu, I. Salomie, E.S¸ . Chifu, C.B. Pop, D. Valea, M. Lupu, et al., Hybrid Invasive Weed Optimization Method for Generating Healthy Meals, Springer, 2016, pp. 335–351, https://doi.org/10.1007/978-3-319-18296-4_28.
[29] A. Laishram, S.P. Sahu, V. Padmanabhan, S.K. Udgata, Collaborative Filtering, Matrix Factorization and Population Based Search: the Nexus Unveiled, Springer, 2016, pp. 352–361, https://doi.org/10.1007/978-3-319-46675-0_39.
[30] R. Katarya, O.P. Verma, An effective collaborative movie recommender system with cuckoo search, Egypt. Inf. J. 18 (2) (2017) 105–112, https://doi.org/10.1016/j.eij.2016.10.002.
[31] https://en.wikipedia.org/wiki/Folksonomy.
[32] https://www.hindawi.com/journals/tswj/2013/528069/
[33] Dorigo M., Stützle T. (2003) The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: Glover F., Kochenberger G.A. (eds) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol 57. Springer, Boston, MA. https://doi.org/10.1007/0-306-48056-5_9.
[34] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.
[35] Zhang, S., Lee, C.K.M., Yu, K.M. et al. Design and development of a unified framework towards swarm intelligence. Artif Intell Rev 47, 253–277 (2017). https://doi.org/10.1007/s10462-016-9481-y.
[36] Ding C., Lu L., Liu Y., Peng W. (2011) Swarm Intelligence Optimization and Its Applications. In: Shen G., Huang X. (eds) Advanced Research on Electronic Commerce, Web Application, and Communication. ECWAC 2011. Communications in Computer and Information Science, vol 143. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20367-1_74.
[37] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005, https://doi:10.4249/scholarpedia.6915.
[38] Weise, Thomas. (2009). Global Optimization Algorithm: Theory and Application.
[39] http://www.scholarpedia.org/article/Swarm_intelligence.
[40] X. Yang and Suash Deb, "Cuckoo Search via Lévy flights," 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, 2009, pp. 210-214, doi: 10.1109/NABIC.2009.5393690.
[41] Alsattar, H.A., Zaidan, A.A. & Zaidan, B.B. Novel meta-heuristic bald eagle search optimisation algorithm. Artif Intell Rev 53, 2237–2264 (2020). https://doi.org/10.1007/s10462-019-09732-5
[42] Krishnanand, K.N., Ghose, D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intell 3, 87–124 (2009). https://doi.org/10.1007/s11721-008-0021-5.
[43] [Chakraborty, Amrita & Kar, Arpan. (2017). Swarm Intelligence: A Review of Algorithms. 10.1007/978-3-319-50920-4_19.
[44] Yang XS. (2010) A New Metaheuristic Bat-Inspired Algorithm. In: González J.R., Pelta D.A., Cruz C., Terrazas G., Krasnogor N. (eds) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence, vol 284. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12538-6_6.
[45] Neshat, M., Sepidnam, G., Sargolzaei, M. et al. Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. Artif Intell Rev 42, 965–997 (2014). https://doi.org/10.1007/s10462-012-9342-2.
[46] Chunhua Ju, Chonghuan Xu, "A New Collaborative Recommendation Approach Based on Users Clustering Using Artificial Bee Colony Algorithm", The Scientific World Journal, vol. 2013, Article ID 869658, 9 pages, 2013. https://doi.org/10.1155/2013/869658.
[47] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872.
[48] Zan Wang, Xue Yu, Nan Feng, Zhenhua Wang, An improved collaborative movie recommendation system using computational intelligence, Journal of Visual Languages & Computing, Volume 25, Issue 6, 2014, Pages 667-675, ISSN 1045-926X, https://doi.org/10.1016/j.jvlc.2014.09.011.

[49] Kordon A.K. (2010) Swarm Intelligence: The Benefits of Swarms. In: Applying Computational Intelligence. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69913-2_6.
[50] https://www.youtube.com/watch?v=08v446aEj-0.
[51] Anchit Jain (2019), Movie Recommendation System – Content Filtering https://medium.com/data-science-101/movie-recommendation-system-content-filtering-7ba425ca0920.
[52] Karthik Bhaskar (2019), A New Approach to Recommendation Systems https://medium.com/datadriveninvestor/a-new-approach-to-recommendation-systems-5cd06c5e84db.
[53] https://datascience.stackexchange.com/questions/33937/how-to-solve-online-clustering-problem.
[54] Hashim, Hashim & Ayinde, Babajide & Abido, Mohammed. (2016). Optimal Placement of Relay Nodes in Wireless Sensor Network Using Artificial Bees Colony Algorithm. Journal of Network and Computer Applications. 64. 10.1016/j.jnca.2015.09.013.
[55] Deb, S., Gao, X., Tammi, K. et al. Recent Studies on Chicken Swarm Optimization algorithm: a review (2014–2018). Artif Intell Rev 53, 1737–1765 (2020). https://doi.org/10.1007/s10462-019-09718-3.
[56] Meng XB, Liu Y, Gao X, Zhang H (2014) A new bio-inspired algorithm: Chicken Swarm Optimization. In:International conference in swarm intelligence. Springer, Cham, pp 86–94.