Journal of Scientific & Industrial Research
Vol. 83, March 2024, pp. 274-281
DOI: 10.56042/jsir.v83i3.4358

# Leveraging Modified Social Group Optimization for Enhanced E-Commerce Recommendation Systems

Sai Shaktimayee Sahu* & Suresh Chandra Satapathy
School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar 751 024, Odisha, India

*Received 25 July 2023; revised 26 January 2024; accepted 12 February 2024*

Intelligent recommendation systems have gained significant popularity in recent times due to their ability to ease item or service selection for users and enhance profit-making opportunities for businesses. E-commerce recommender systems are in high demand across online platforms. There is a pressing need for continuous innovation to improve the performance of these e-commerce recommendation systems in terms of accuracy in suggesting preferences. However, many existing recommendation systems are not able to perform well when there is a data sparsity or incomplete data. To address above challenges, this study introduces a novel approach that combines collaborative filtering with Modified Social Group Optimization (MSGO), a type of evolutionary optimization methods. The main objective is to improve the precision of the recommendation system specifically for movie recommendations. The collaborative filtering technique is leveraged to analyse user-item interactions and find patterns to predict user preferences. To evaluate the proposed system, a simulation is conducted using movie recommendation data. The results demonstrate that the integration of MSGO into the collaborative filtering framework yields improved performance compared to the original SGO algorithm. These findings provide promising evidence for the effectiveness of MSGO in enhancing the accuracy of movie recommendations within the e-commerce context.

**Keywords:** Collaborative filtering, e-Commerce, SGO, Evolutionary optimization, Recommendation system

## Introduction

The rapid growth of e-commerce has revolutionized online shopping and product identification worldwide.[1] Especially during the recent pandemic, more people relied on various online shopping sites to select and purchase their merchandise. However, the vast numbers of available products and diverse user preferences have created a challenge in making relevant selections. E-commerce recommendation systems play a vital role in this scenario.[2] These systems analyse user data, such as browsing history, buying behaviour, choices, tastes, and demographic profiles, to generate personalized recommendations. These personalized suggestions not only facilitate an easier shopping experience for users but also save them from the inconvenience of physically visiting multiple stores. Moreover, they boost profits for sellers. The primary goal of e-commerce recommender systems is to provide accurate, relevant, and diverse recommendations that align with the user's preferences. These systems are designed to learn from user data, understand their needs, likes, and dislikes, and improve their overall buying experience, benefiting both customers and sellers.

In the literature, various techniques like collaborative filtering[3], content-based filtering[4], and hybrid approaches have been widely adopted for building intelligent recommendation systems. These techniques have significantly impacted the performance of recommender systems, reducing efforts in selecting relevant products. However, they also face challenges, including the scarcity of user data during initial searches, data sparsity, scalability, privacy concerns, accuracy, and the need to adapt to changing user preferences. As a result, researchers and practitioners have sought alternate, innovative, and cost-effective approaches. Recent trends have shown increased usage of strategies like Deep Learning and Natural Language Processing. In this work, the authors explore the potential of evolutionary optimization algorithms, which have shown effectiveness in handling large search spaces with uncertainty to find optimal solutions.

To enhance the accuracy of the e-commerce recommendation system, the authors employ Modified Social Group Optimization (MSGO)[5,6],

*Author for Correspondence
E-mail: saishakti1997@gmail.com

which leverages the diversity and exploration capabilities of the technique. The MSGO is a modified version of Social Group Optimization (SGO), initially developed by Satapathy and Naik. The SGO has demonstrated excellent performance in numerous optimization functions and has been extensively applied in various engineering applications.[7–11] The adoption of MSGO aims to provide more precise results regarding users' preferences and dislikes, ultimately leading to improved recommendations in the e-commerce applications such as retail and wholesale; auctioning; e-banking; booking, and so on.

**Literature Review**

Ujjin *et al.*[12] proposed a new recommender system that's designed to help us find personalized recommendations that perfectly suit our interests and preference. Here this paper showed that the Particle Swarm Optimization (PSO) system performed better than a non-adaptive approach. It was able to make more accurate predictions compared to the Genetic Algorithm system and the Pearson algorithm in most situations. In simple terms, the PSO system was really good at finding the right answers and outperformed the other methods it was compared to.

Kulkarni *et al.*[13] described a review of the state-of-the-art techniques named as the paper titled "Context Aware Recommendation Systems" provides an in-depth examination of the current state-of-the-art methods used in context-aware recommendation systems. It offers a comprehensive overview of various techniques and approaches that consider the contextual components, such as time, location, and user preferences, to deliver more personalized and relevant recommendations. The review aims to highlight the strengths, limitations, and potential future directions in this rapidly evolving field of recommendation systems. But in this survey may not cover the very latest techniques in context-aware recommendation systems, potentially leaving out some promising approaches.

Kumar *et al.*[14] proposed a hybrid Recommendation System (RS) for movies that combines Collaborative Filtering (CF), Content-Based Filtering (CBF), and sentiment analysis of tweets from microblogging sites. By leveraging movie-related tweets, the system aims to understand current trends, public sentiment, and user responses to improve movie recommendations. Experiments show that the proposed model outperforms other approaches in precision. But the two disadvantages of this paper are as follows Limited Language Scope: here the RS focuses on sentiment analysis of tweets in English, which might restrict its effectiveness for non-English speaking users or users who express their opinions in different languages and Bias from Twitter Data: The RS heavily relies on sentiments expressed on Twitter, which may not represent the overall movie audience's preferences, leading to potential bias in recommendations based on Twitter users' opinions.

Schafer *et al.*[2] discussed that how recommender systems have evolved from being novelties to essential business tools in E-commerce. It explores how these systems learn from customers' preferences and recommend products to enhance sales. The paper presents taxonomy of recommender systems, detailing their interfaces, technologies, and customer inputs. It also highlights the potential for recommender systems to automate mass customization for E-commerce sites, enabling personalized services and predicting demand in the supply chain. But the limitations of this paper are Limited Empirical Evidence: lack in-depth empirical evidence or case studies to support the claims and findings regarding the effectiveness of recommender systems in increasing sales and achieving mass customization and Ethical Considerations: it may not thoroughly address the potential risks and biases associated with recommender systems, especially regarding customer privacy and the balance between site benefits and customer preferences.

From the short literature review, the gap is clearly evident. Though there are many advances in developing recommender system, none of those have taken care the sparsity of data which affect the overall prediction performance. In this work, we have mainly concentrated the focus on how to develop an improved recommendation system to handle sparsity and incomplete data for gaining prediction accuracy.

## Recommendation System Fundamentals

Recommendation systems belong to the category of intelligent systems. They are designed to gather users' search data, understand their preferences, behaviours, etc., and suggest products or services that align with their likes and dislikes. In recent times, many online firms, entertainment platforms, social media, and other platforms have extensively employed such recommendation systems to enhance their customer base and increase profits. There are generic system designs and requirements applicable to any recommender system. The fundamental components of each such system, as illustrated in Fig. 1, include data collection, user profiling, item representation, recommendation algorithms, evaluation metrics, the cold-start problem, and real-time adaptation.

**User Data and Profiling**

User profiling is primarily built upon the collection of user data online. Typically, users create their logins on many online platforms (although some platforms collect user data even without explicit user login creation). Alongside demographic information, explicit feedback provided by users while browsing the platform/services, as well as implicit data such as browsing history (including items searched, days of the month, seasons, occasions, and time spent on the platform), are recorded. These vital data serve as the backbone for achieving high-performance in a recommendation system. By capturing and analysing this demographic data, diverse preferences, and other relevant properties, the intelligent system creates relevant and useful user profiling.

**Item Representation**

In this phase, vectorization is performed for each item. Feature extraction and selection are the main tasks during this stage. Attributes like price, colour, brand, genre, or other keywords are extracted and represented in vectors using suitable coding techniques for processing. This stage greatly supports pattern generation and matching, facilitating grouping of items into several baskets based on similarities or patterns. The efficiency of the recommendation system predominantly depends on the quality of patterns generated and matched. Therefore, heavy emphasis is placed on this stage to provide real-time recommendations of relevant results. Various encoding schemes, distance metrics, and clustering algorithms are employed to develop a robust component in this system.

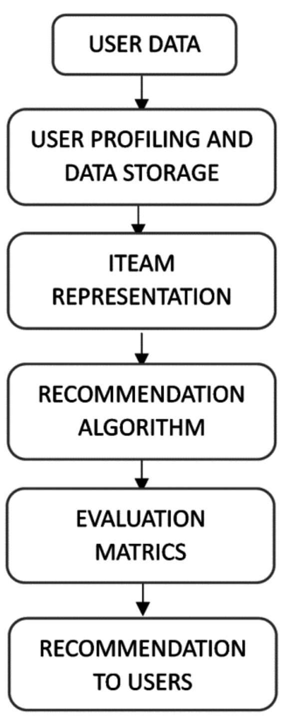


Fig. 1 — Components of recommendation system

**Recommendation Algorithms**

This component encompasses the techniques and algorithms that form the central part of an intelligent recommendation system's functioning. The data collected in phase one must be analyzed to derive patterns for generating suitable recommendations. Users' profiling, purchase behaviour, and other relevant data serve as input ingredients to this phase. Various well-researched algorithms such as collaborative filtering, content-based filtering, hybrid filtering, knowledge-based filtering, NLP approaches, and Deep learning models are commonly used. In this work, MSGO has been employed in this phase.

**Evaluation Metrics**

Performance measurement is a crucial aspect of any development, particularly software development. The performance of the designed recommendation system must be evaluated from both user and seller perspectives for various requirements. Among others, precision, accuracy, recall, mean average precision, and diversity are the most sought-after metrics. These metrics gauge the quality and effectiveness of the recommendation system.

**Recommendations to Users**

This is the final part of the system where the generated recommendations are delivered to the user through a suitable channel or interface.

## Social Group Optimization: Preliminaries

The SGO algorithm is introduced by Satapathy & Naik in the year of 2016. The SGO algorithm is for dealing with the continuous optimization problem. The SGO algorithm is based on social behaviour of human to solve complex problem. Each human is empowered with knowledge that decides the capacity level for solving a problem. The SGO algorithm is a meta-heuristic population-based algorithm, whereas the population is considered as a group of persons.

Each person represents as candidate solution for the problem. The behavioural traits in humans represent the design variables of problems which correspond to dimension of the problem. Each person acquires knowledge that is corresponding to the 'fitness'. For proper understanding of SGO algorithm, one must follow the paper. The algorithm given below details the flow of SGO.

**SGO Algorithm**

Start

Initialise population size N, dimension D, self-introspection parameter C, maximum number of iteration max_iter

Create initial population $P_i$ (i = 1, 2, ………, N) in D-dimensional search space,

During the initialization phase, randomly distribute all of the group members throughout the search space.

Determine the fitness value $f_i$in every individual $P_i$According to the issue at hand.

Stage 1: Finding the best member (gbest) of the group

[min value, index no.] = $\min\{f_i, i = 1,2,3, \ldots, N\}$

Gbest = P (index no, :) in the case of a minimization problem.

Stage 2: Launch improving phase to update people's knowledge using gbest

Stage 3: Start the acquiring phase by selecting a person at random from the group and adhering to the gbest to further update their knowledge.

Stage 4: if

Everyone possesses about comparable fitness values or meets the termination requirement.

then

finish the search and show the best solution for the selected issue

else

go to Stage 2.

endif

Stop

## MSGO basics

In this section, we present the ideas behind the improvements made to MSGO. When comparing an individual's current fitness level with their prior fitness level, just the absolute fitness value is examined in conventional SGO and MSGO, disregarding the values of the corresponding dimensions. Nonetheless, there can be circumstances in which the dimensions vary but the fitness stays the same for both the prior and current values. Under such circumstances, it may be advantageous to take into account the person who offers the same fitness values but in a different configuration. This makes it possible for someone fresh to take part in the procedure. We provide a straightforward illustration of this concept below.

Assume that Xi is the present person with D dimensions. And after values are obtained using either Algorithm 1 or 2, Xinew remains the same person.

F (Xi) and f (Xinew) are Xi and Xinew's respective equivalent fitness values.

If f (Xinew) < f (Xi) then Xinewis selected for the next step. (for minimization problem)

Else if

F (Xinew) == f (Xi)

then the respective dimensions of both Xi and Xinew are checked. If dimensions of Xinew are differing from Xi then Xinew is chosen.

Let Xi = [ 20 40 60 80] is the total of each person's maximal level of fitness = 200

Xinew has been discovered to be [40 20 80 60] following the execution of Algorithm 1 or 2. Here, the fitness level is likewise 200. But to create an improved variant, we'll disregard Xi and move Xinew to the following stage for analysis. In keeping with the identical concept, our enhancement suggestion likewise chooses the gbest.

## Proposed MSGO Recommendation System

The work presented in this paper focuses on creating a recommendation system for movie recommendations using the MovieLens dataset.[15] The central engine for the recommendation algorithm revolves around the collaborative filtering technique.

Collaborative filtering is one of the most preferred techniques in recommendation systems. Its central idea is based on the assumption that users who share similar buying patterns or preferences in the past are likely to have the same patterns matched in the future. This collaborative approach relies on user preferences as the principal metadata for developing a recommendation system. As shown in Fig. 1, the input to the collaborative filter algorithm comes from users' profiling.

There are two types of collaborative filtering: User-based collaborative filtering and Item-based collaborative filtering. The main difference between these two filtering techniques lies in the entities used for developing recommendations. In user-based collaborative filtering, recommendations are made to the target user by identifying the preferences of other users that match with the target user's preferences. On the other hand, in item-based collaborative filtering, recommendations are made based on the ratings or interactions of other users that match with the ratings and patterns of the target user.

The implementation of the user-based approach is quite intuitive and straightforward. In contrast, the item-based approach involves the use of various similarity distances such as Euclidean distance, cosine similarity, or Pearson correlation to closely match items. The item-based approach tends to work well for large datasets compared to the user-based approach since similarity measures are calculated offline. Although collaborative filtering approaches have many advantages, they also face challenges such as data sparsity, scalability, and the cold start problem.

In this work, we have chosen the user-based collaborative filtering for further improvement. One of the constraints in user profiling is that the collected data are often sparse or incomplete. Sparsity occurs when most users interact with only a small portion of the data items, or when a small number of users interact with most of the data items, or both. This sparsity can lead to inaccurate or limited recommendations as it becomes challenging to identify meaningful patterns between users or items. As a result, the recommendation system might be trapped in a local optima solution, only recommending popular items and leaving out other obscure/unexplored items.

To address the issue of sparsity, we have combined an evolutionary approach known as MSGO with the collaborative filtering approach. We have used the MovieLens dataset for all our simulations. This dataset contains a total of 943 users, and each user has attributes such as age, gender, and occupation as profile parameters. Whenever a new user is added, the demographic information is recorded with a timestamp. The items in this dataset refer to movies, and their attributes, such as genres, theatre, and dates of release, are recorded.

The representation of items/users is done using a vector consisting of the attributes/features/dimensions considered for processing. In our work, we have used 22 features, including movie ratings, age, occupation, and 18 movie genre types such as action, adventure, animation, comedy, and more. These 22 attributes are used to represent the profile of a user. User (j, i) – profile for user with its movie rating of 3 is shown in Table 1. The first four attributes include binary gender (0 for male, 1 for female), integer ratings on a scale from 1 to 5, and truncated age to represent years only. Occupation is encoded as an integer value from the provided list in the dataset. The remaining 18 attributes are encoded in binary form, where 0 represents the class is not present, and 1 indicates the class is present.

The profile selection is the crucial aspect of the collaborative filtering system. Recommendations are made based on matching the profile of the active user with the profiles in the database. A similarity matching algorithm is utilized to find the best matches, followed by voting to recommend the most suitable item. Two cases arise in this situation. Case 1: considering the entire profiles in the database for similarity calculation, and Case 2: selecting a subset of the dataset for profile selection. The second case further leads to two sub-cases. Case 1: selecting a neighbourhood of an active user, where the neighbourhood is defined by a random or fixed integer value that should be less than the number of profiles in the database. Case 2: selecting a random number of profiles from the database.

If the dataset is very large, selecting the entire profiles for similarity calculation can be time-consuming and may not be preferred. However, it often guarantees providing the most suitable recommendation. Therefore, using a subset of the entire database is a wise approach, even though the recommendation may not be as accurate as in the previous case. However, it is faster and can be used as an online option. In our work, we have simulated the subset approach. In the first simulation, a fixed number of profiles are taken from the database for similarity calculation. A random integer 'p' (which must be less than the number of profiles in the database) is generated, and we choose the first 'p' profiles from the database. In the second simulation, random 'p' profiles are chosen from the database. A modified Euclidean Distance is used to calculate the similarity measure.

Modified Euclidean distance is calculated with the bellow formula:

$$E(A,B) = \sqrt{\sum\nolimits_{m=1}^{n} \sum\nolimits_{F=1}^{22} w_{a^*} D_{m,a}(A,B)^2} \qquad \ldots(1)$$

where,

Table 1 — User (j, i) – profile for user with its movie rating of 3

| Rating | Age | Gender | Occupation | 18 genres |
|---|---|---|---|---|
| 3 | 23 | 1 | 23 | 010001000000000000 |

A is the active user B is the user obtained after the profile selection (A ≠ B).

n: number of common movie A and B rated

$W_a$: active users' preference for feature a

m: common movie between user (A, n) and between (B, n) that appears.

$D_{m,a}$ : difference in profile value foe feature A and B on movie item n.

From the literature, it is evident that most Recommender systems use pooling when comparing between two profiles. While having the same opinion on a movie might indicate similarity between two users, it's essential to consider that movie preferences alone may not be the sole deciding factor. Personal details and background of users could also influence their choices. Hence, the inclusion of demographic information such as age, gender, etc., is necessary to identify similarities between users. To achieve this, a new variable called 'feature preference' is introduced in Eq. 1. To implement a realistic recommender system, these feature preferences must be carefully fine-tuned to adequately reflect each user's preferences in the process.

In our work, we utilize the Modified Social Group Optimization (MSGO) algorithm to optimally fine-tune these feature preferences. The profile of a user is represented as a vector of size 22, with each cell assigned unique features/attributes mentioned earlier in the paper. Each attribute is assigned specific values, which we refer to as "feature preferences." These feature values are adjusted to find the best candidate/profile with a set of weights that matches the active user's profile. In our work, this adjustment is accomplished using the MSGO algorithm.

Before delving into the details of how this is achieved, let's provide a brief description of the MSGO algorithm.

## Simulations and Analysis of Results

All our simulations are performed for the purpose of recommending movies, and we chose a dataset from MovieLens containing a total of 943 user profiles. Before conducting any simulations, an initialization is performed on user profile values. In many evolutionary-based methods, values are usually assigned randomly between 0 and 1. Similarly, in our simulations, we set the values of the 22 attributes by calling a random function between 0 and 1. For attributes that must be either 0 or 1, we apply a threshold mechanism. If the value is greater than or equal to 0.5, it is set to 1; otherwise, it is set to 0. During random assignments, if the value of any attribute/feature is 0, it is considered that the respective attribute is absent.

At the beginning of the MSGO implementation, a population of fixed size is chosen. The input to MSGO is a matrix of size M × N where, M represents the number of candidate/user profiles, and N represents the number of features in each user (which is 22 in our experiments). Suitable parameters are set for MSGO, and the introspection factor (c) is empirically set to 2 to yield the best results. To evaluate the performance of MSGO, a fitness function must be defined, guiding MSGO towards the global optima to set the best "feature preferences" for each individual/user. The fitness metric considers the sum of weighted values. Once MSGO is executed for a stipulated number of user-defined iterations, the output is used for the recommendation purpose.

For training and testing, the dataset needs to be split. We followed the classical 2/3rd and 1/3rd approach, where two-thirds of the data is used for training, and one-third is used for testing. Before that, we set up the simulation. We used the neighbourhood approach for the selection of user profiles, which is efficient. We carefully split a subset of the dataset (943 users) into training, validation, and testing. We followed two experimental setups as given below:

#### Experiment A

The first 100 users are chosen from the 943 and split into three parts: training, validation, and testing. For training, out of the first 100, we take the first 80, the next 10 for validation, and the last 10 as active users for testing. This sequence is repeated for the first 900 data in the original set, ignoring the last 43 to ensure even numbers for the simulation. This allows for overall performance comparisons of our proposed recommender.

#### Experiment B

In this experiment, the first 100 users are taken as active users for testing, and another 100 are randomly chosen from the remaining users for training. Additionally, 50 random users are taken for validation, and they are not part of the training set or the active user set.

For comparative analysis, we simulated the original SGO with the same algorithmic and user parameters, and the dataset split was the same as in Exp A and Exp B. The accuracy of the system is defined and

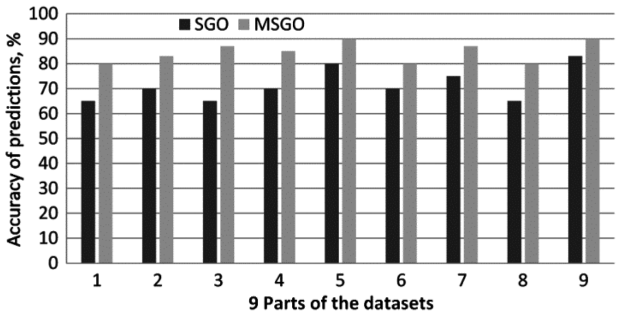


Fig. 2 — Comparison of performance of SGO Vs MSGO for the Experiment A

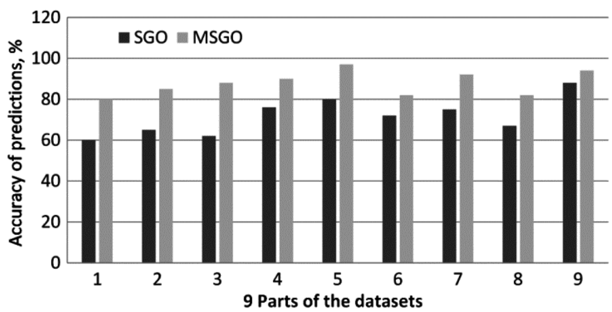


Fig. 3 — Comparison of performance of SGO Vs MSGO for the Experiment B

computed with the total number of correctly predicted recommendations. The accuracy charts for both experiments is presented in Fig. 2 and Fig. 3 respectively.

From the figures above, it is evident that Modified SGO outperforms SGO significantly. We conducted 10 simulations for each sample phase, and the average accuracy is reported in the figures for both experiments. The population size was kept at 30 for all simulations and experiments, while the introspection factor for both SGO and MSGO was set to 2.

Regarding the selection of user profiles for predictions in both experiments, we observed that the method of selection does not have a significant impact. This indicates the ability of MSGO to converge to optimal values of feature preferences, ensuring that the preferences of active users align with those in the validation and testing sets. This is a crucial finding in our work and is valuable for developing collaborative filtering recommendation systems for various applications. However, it should be noted that there are some challenges as well. In our MSGO simulations, we used a fixed value for the introspection parameter, which was chosen empirically after several simulations to find the best value. Further research is needed to explore methods for making this parameter adaptable during the algorithm's run.

## Conclusions

This work focused on improving the e-commerce recommendation system for movie recommendations using an evolutionary optimization technique. The base recommender system considered in this paper is collaborative filtering. The selection of user profiles and fine-tuning of feature preferences for each profile are crucial aspects in collaborative filtering. Unless optimal feature weights are chosen, the system might be trapped in a local optimal path and fail to provide accurate preferences for active users. To address this, we applied a modified version of Social Group Optimization (SGO) to fine-tune the feature values of each selected profile, which are then analysed to predict preferences for active users. This is especially crucial when dealing with datasets having sparsity. The comparative analysis with the original SGO clearly shows the effectiveness of MSGO over SGO for the movie dataset. However, further research is envisioned in the future to compare MSGO with other state-of-the-art algorithms. This will help in gaining a deeper understanding of the algorithm's performance and its potential applications in various recommendation systems.

## References


1 Bawack R E, Wamba S F, Carllio K D A & Akter S, Artificial intelligence in e-commerce: A bibliometric study and literature review, *Electron Mark*, **32** (2022) 297–338, doi: 10.1007/s12525-022-00537-z.

2 Schafer J B, Konstan J & Riedl J, Recommender systems in e-commerce, *ACM Int Conf Proceeding Ser*, 1999,158–166, doi: 10.1145/336992.337035.

3 Parvin H, Moradi P & Esmaeili S, TCFACO: Trust-aware collaborative filtering method based on ant colony optimization, *Expert Syst Appl*, **118** (2019) 152–168, doi: 10.1016/j.eswa.2018.09.045.

4 Reddy S R S, Nalluri S, Kunisetti S, Ashok S & Venkatesh B, Content-based movie recommendation system using genre correlation, in *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018,* (Springer Singapore), **2** (2019) 391–397, doi: 10.1007/978-981-13-1927-3_42.

5 Satapathy S & Naik A, Social group optimization (SGO): A new population evolutionary optimization technique, *Complex Intell Syst*, **2**(**3**) (2016) 173–203, doi: 10.1007/s40747-016-0022-8.

6 Naik A, Satapathy S C & Abraham A, Modified social group optimization—a meta-heuristic algorithm to solve short-term hydrothermal scheduling, *Appl Soft Comput J*, **95** (2020) 106524, doi: 10.1016/j.asoc.2020.106524.

7 Kalla N & Parwekar P, Social group optimization (SGO) for clustering in wireless sensor networks, *Intell Eng Inform*, **695** 2018 119–128, doi: 10.1007/978-981-10-7566-7_13.

8 Rajinikanth V & Satapathy S C, Segmentation of ischemic stroke lesion in brain mri based on social group optimization

and fuzzy-tsallis entropy, *Arab J Sci Eng*, **43**(**8**) (2018) 4365–4378, doi: 10.1007/s13369-017-3053-6.

9 Jena J J & Stapathy S C, A new adaptive tuned social group optimization (SGO) algorithm with sigmoid-adaptive inertia weight for solving engineering design problems, *Multimed Tools Appl*, **83(1)** (2021) 3021–3055, doi: 10.1007/s11042-021-11266-4.

10 Praveen S P, Rao K T & Janakiramaiah B, Effective allocation of resources and task scheduling in cloud environment using social group optimization, *Arab J Sci Eng*, **43**(**8**) (2018) 4265–4272, doi: 10.1007/s13369-017-2926-z.

11 Jena J J, Dash S C B & Satapathy S C, Stability analysis based parameter tuning of social group optimization, *Complex Intell Syst*, **8**(**4**) (2022) 3409–3435, doi: 10.1007/s40747-022-00684-y.

12 Ujjin S & Bentley P J, Particle swarm optimization recommender system, *2003 IEEE Swarm Intell Symp SIS 2003 – Proc* (IEEE) 2003, 124–131, doi: 10.1109/SIS.2003.1202257.

13 Kulkarni S & Rodd S F, Context aware recommendation systems: A review of the state of the art techniques, *Comput Sci Rev*, **37** (2020) 100255, doi: 10.1016/j.cosrev.2020.100255.

14 Kumar S, De K & Roy P P, Movie recommendation system using sentiment analysis from microblogging data, *IEEE Trans Comput Soc Syst*, **7**(**4**) (2020) 915–923, doi: 10.1109/TCSS.2020.2993585.

15 https://grouplens.org/datasets/movielens/100k [Accessed on 15th April 2023]