Reaching Agreement in an Interactive Group Recommender System

Dai Yodogawa[1][a] and Kazuhiro Kuwabara[2][b]

[1]*Graduate School of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577 Japan*

[2]*College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577 Japan*

Keywords: Group Recommender System, User Model, Agent, Conversation Strategy.

Abstract: For a group recommender system, it is important to recommend an item that can be accepted by all group members. This paper proposes a group recommender system where preferences elicited from group members are used to select an item that is agreeable to all of them. In this system, an agent that corresponds to each group member manages estimation of the corresponding user's preferences. Virtual negotiation is conducted among these agents to find an appropriate item to recommend, and the selected item is presented to group members. If it is not accepted, the system asks members to relax their requirements and accordingly updates its recommendation. We report and discuss the results of simulation experiments with different personality types of conflict resolution and different conversation strategies.

1 INTRODUCTION

With ever-increasing information available, recommender systems have become part of our everyday lives. Many recommender systems target an individual user, but much research also has focused on systems that target a group of people (Ricci et al., 2015). For a group recommender system, a recommendation can be generated by (1) aggregating users' profiles to make a profile as a group and applying a recommender algorithm for an individual user, or (2) aggregating items' rankings or ratings for each user to produce a recommendation for a group (Felfernig et al., 2018).

For certain application domains, such as finding a group travel destination, it is important to recommend an item that all the group members can accept. For such a case, the concept of negotiation is a promising approach that makes use of users' rankings or ratings for each item (Bekkerman et al., 2006). For each user, an agent is placed that has the preference information of the corresponding user and acts on behalf of the user. Negotiation is often conducted among these agents to find an agreed item.

We develop an interactive group recommender system that asks for users' requirements and feedback on a recommended item (Yodogawa and Kuwabara, 2019). By asking users to relax their requirements, the system attempts to find an item that all the group members can accept.

In this paper, we extend our system to include user agents. Here, an agent is not meant to act on behalf of the corresponding user, but rather, the agent is placed inside the recommender system and it manages the estimated values of a user's preferences. By introducing these agents, a recommender system can simulate negotiations among users inside the system and produce an item that might be acceptable to all the users.

When the produced item is actually accepted by the users, the recommendation process ends. Otherwise, the system asks the users to relax their requirements. Based on their responses, the system updates its estimates of users' profiles, selects a new item for recommendation, and presents it to the users. This process continues until the selected item satisfies all the users or no further items can be recommended.

In this paper, we consider a conversation strategy for the system to effectively reach an agreement. To evaluate the proposed system, we conduct simulation experiments with a user model that is based on personality types of conflict resolution.

The remainder of this paper is organized as follows: Section 2 describes related work, Section 3 describes our proposed agent-based mechanism for a group recommendation system, Section 4 presents and discusses the results of simulation experiments to examine the characteristics of the proposed system, and Section 5 concludes the paper and discusses future work.

[a] https://orcid.org/0000-0002-0319-9225
[b] https://orcid.org/0000-0003-3493-1076

Yodogawa, D. and Kuwabara, K.
Reaching Agreement in an Interactive Group Recommender System.
DOI: 10.5220/0009160502950302
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 1*, pages 295-302
ISBN: 978-989-758-395-7; ISSN: 2184-433X

2 RELATED WORK

Several studies have applied a multi-agent system to a group. In the PUMAS-GR system, a user agent is set up for each user, and agents negotiate with each other using a monotonous concession protocol (Villavicencio et al., 2016). It was reported that the PUMAS-GR system found a better solution compared with the approach of aggregating the users' ranking of items. Similarly, agents have been introduced to elicit group preferences (Garcia et al., 2011), where a user agent models a human user's preferences. To aggregate preferences, voting procedures and negotiation among agents are utilized.

In contrast to these systems, in the proposed system, user agents do not act for a human user. Rather, they mimic the negotiation between users with their estimated preferences.

3 RECOMMENDATION MECHANISM

3.1 Data Model

In the proposed recommender system, we assume there are m items, from which an item is selected for recommendation. Each item t_i ($1 \leq i \leq m$) has n attributes that describe the item's features. We also assume that there are K users in the group. User u_k ($1 \leq k \leq K$) will input their requirements $r_j^{(u_k)}$ for each attribute j. Item t_i is assumed to have an evaluation function $eval_j^{(t_i)}$ for each attribute j, which takes each user's requirement value for attribute j, $r_j^{(u_k)}$, as its parameter and returns an evaluation score about the user's satisfaction of item t_j regarding attribute j. The range of the evaluation function is assumed to be between 0 and 1, where 0 means that the user does not like the item at all regarding the particular attribute. Note that the domain of the evaluation function is set differently for each attribute. For example, attributes describing price and distance may have different value ranges.

User u_k's utility value for item t_i, $U^{(u_k)}(t_i)$, is calculated as follows:

$$U^{(u_k)}(t_i) = \frac{\sum_{j=1}^{n}(w_j^{(u_k)} \cdot eval_j^{(t_i)}(r_j^{(u_k)}))}{\sum_{j=1}^{n} w_j^{(u_k)}} ,$$

where $w_j^{(u_k)}$ represents a weight to model the importance of attribute j is to user u_k's utility value. In addition, we assume that user u_k has a threshold $T^{(u_k)}$ that needs to be satisfied when a recommended item is accepted by the user. That is, user u_k is assumed to accept item t_i whose utility value for user u_k equals or exceeds the threshold ($U^{(u_k)}(t_i) \geq T^{(u_k)}$).

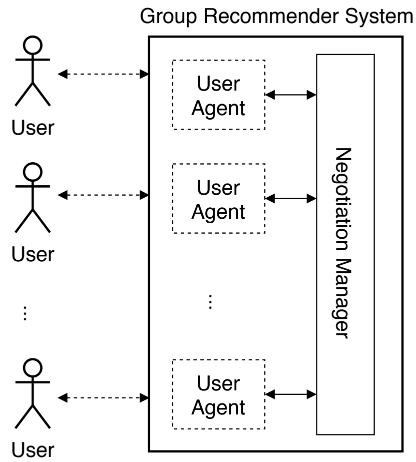

Figure 1: Overview of the proposed system.

3.2 Recommendation Flow

Figure 1 shows the overall structure of the proposed system. We place a *user agent* for each user. This agent manages the estimation of the corresponding user's preferences. Note that the agent is placed inside the recommendation system virtually and it does not know the user's exact preferences and does not act on behalf of the user. Rather, it is placed to simulate the negotiation process inside the recommender system and to produce a recommended item. Since the proposed mechanism simulates negotiation among agents to find an item to recommend, we also placed a *negotiation manager* that mediates agents inside the system.

Figure 2 shows the overall control flow of the proposed recommendation mechanism. First, the system asks each user their requirements for each attribute. Based on their responses, an agent is set up for each user inside the system. The agent knows the current value of the user's requirements from the conversation between the system and the user, but it is assumed that the agent does not know the threshold and the weight in calculating the items' utility for each user. The agent holds estimations of these values. They are initialized to pre-determined values and updated as the system and the user interact.

Next, we simulate the negotiation process among the user agents to determine which item to recommend. For a two-agent case, if the *monotonic concession protocol* is adopted for the negotiation and the agents behave according to the *Zeuthen strategy*, it is known that they reach an agreement where the product of their utility values is maximized (Zeuthen, 1930). Using this finding, the *one step protocol* was devised where the agreement between two agents is selected as the one that maximizes the product of the two agents' utility (Rosenschein and Zlotkin, 1994).

In the proposed system, we use an extended *one step protocol* for multiple users (Endriss, 2006). The user agent in the system calculates the estimated utility of items for its corresponding user and reports the values to the negotiation manager in the system. The negotiation manager aggregates the estimated utility values and selects the item that maximizes the product of all users' estimated utility values. That is,

$$\arg\max_i \prod_k \widehat{U}^{(u_k)}(t_i),$$

where $\widehat{U}^{(u_k)}(t_i)$ denotes an estimated value of $U^{(u_k)}(t_i)$. If there are multiple items with the largest product of utility values, we select one of these at random.

An item should satisfy the user when the item's true utility value for the user equals or exceeds the threshold of the user. Since the system only knows the estimated utility values and threshold of a user, the system cannot determine whether the selected item will be accepted by the user. The system presents the item and asks the user if they are satisfied with it. Here, we assume that the goal of the system is to find an item that is acceptable to all users in the group. If all the users are satisfied, the recommendation process ends. If there is a user who is not satisfied, another recommendation is sought.

3.3 Exploring Phase

When a user responds that they are not satisfied with a recommended item, the system updates its estimated threshold for the user. For example, assuming that item t_r is recommended and user u_k is not satisfied with t_r, the estimated threshold $\widehat{T}^{(u_k)}$ is updated to the estimated utility of t_r for user u_k, $\widehat{U}^{(u_k)}(t_r)$.

To explore other possibilities for a recommended item, the system asks if the requirement for the item can be relaxed. If the user agrees to relax their requirements, utility values for items may change. A previously recommended item might be accepted, or another item could be recommended.

More specifically, for the recommended item t_r and user u_k, the system finds attribute l that has the lowest evaluated score. That is, $l = \arg\min_j \frac{\widehat{w}_j^{(u_k)} \cdot eval_j^{(t_i)}(r_j^{(u_k)})}{\sum_{j=1}^{n} \widehat{w}_j^{(u_k)}}$. The system asks the user to relax the requirement for attribute l, $r_l^{(u_k)}$. Note that since the system does not know the exact value of weight $w_j^{(u_k)}$, its estimated value is used for calculation.

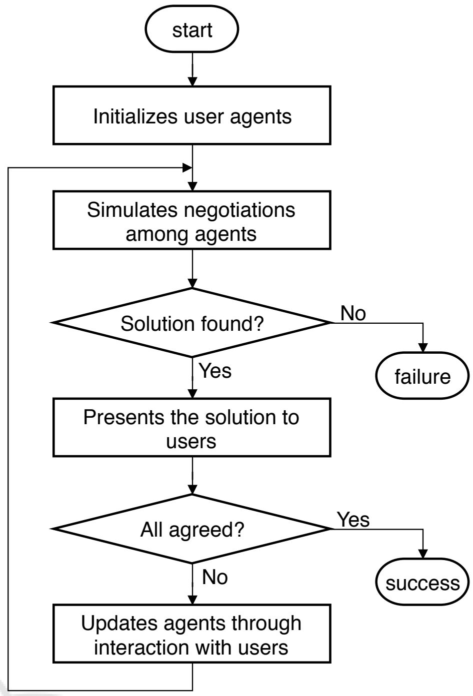

Figure 2: Overall control flow of the recommendation mechanism.

After attribute l is determined, the system asks user u_k to relax the requirement for attribute l, $r_l^{(u_k)}$. When doing so, the system may suggest how much the requirement $r^{(u_k)}$ be relaxed. We call this a *hint* in the sense that the user can decide how much the requirement should be relaxed to reach an agreement. This value is determined so that the utility of the recommended item for user u_k, $U^{(u_k)}(t_r)$, will become higher than user u_k's threshold, $T^{(u_k)}$.

One possible heuristic to determine the amount of concession to request is as follows. Since the estimated utility value of the recommended item for the user is the weighted average of the evaluation value of all the attributes, the system would ask the user to relax the requirement for attribute l so that the evaluated value of attribute l matches the average value.

A user responds to such a request from the system, either by rejecting the request or changing the requirement for attribute l. The system then updates the information about the user and recalculates the utility values of possible items for all the users. As in the initial round, negotiation among user agents is simulated and an item to be recommended next is calculated.

To avoid falling into an infinite loop, the system will recommend the same item at most L times. Items that have been recommended L times are removed from possible items to be recommended. When the system cannot find an item because there are no remaining items for recommendation, the recommendation process ends in failure.

4 SIMULATION EXPERIMENTS

To investigate the characteristics of the proposed mechanism, we conducted simulation experiments as follows.

4.1 Dataset

We defined six attributes ($n = 6$) to describe an item assuming it is a sightseeing spot, as shown in Table 1. Using these attributes, we randomly generated 120 items (sightseeing spots) ($m = 120$) for the simulation experiments. The range of values for each attribute are also provided in Table 1. Here, access, landscape, crowdedness, and barrier free are assumed to be five-star review scores. Thus, the value range of these attributes is between 1 and 5.

4.2 User Model

4.2.1 Parameters

In the simulation experiments, a user is characterized by the following parameters.

- *Initial Requirements* for each attribute ($r_j^{(u_k)}$)
  The system first asks each user for their requirements for each attribute. The requirements may change during the recommendation process.
- *Weight* for each attribute ($w_j^{(u_k)}$)
  The utility value for an item is calculated as the weighted average of the score of all the attributes of the item. The weights may be different from user to user.
- *Initial Threshold*
  When the system presents a recommended item (t_r), a user's utility value ($U^{(u_k)}(t_r)$) is calculated. When it equals or exceeds its threshold ($T^{(u_k)}$), the user should accept the item.
- *Threshold Decay*
  The criteria for accepting a proposal during a negotiation tend to become lower after many rounds of negotiations. Thus, to simulate such behavior, we define a parameter to decrease the threshold after each time the system presents a recommended item.
- *Concession Factor*
  When the system asks a user to relax the requirement for a particular attribute, the user may relax the requirement according to the *concession factor*. Note that for *avoiding* users, the *concession factor* is irrelevant since any item is acceptable to them as their threshold is set to 0.

4.2.2 Evaluation Function

To calculate the utility of items t_i for user u_k, we defined the evaluation function $eval_j^{(u_k)}$ for attribute j as follows. If j is price or distance,

$$eval_j^{(t_i)}(r_j^{(u_k)}) = \begin{cases} \frac{r_j^{(u_k)}}{t_i[j]} & \text{if } r_j^{(u_k)} \leq t_i[j] \\ 1 & \text{otherwise}, \end{cases}$$

where $t_i[j]$ represents the value of attribute j of item t_i.

If j is a review-type attribute such as access or landscape,

$$eval_j^{(t_i)}(r_j^{(u_k)}) = \begin{cases} 1 & \text{if } r_j^{(u_k)} \leq t_i[j] \\ \frac{5-r_j^{(u_k)}}{5-t_i[j]} & \text{otherwise}. \end{cases}$$

4.2.3 Personality Type

To determine these parameters, we used four types of interpersonal conflict-handling behavior as described in the Thomas-Kilmann Conflict Model (Kilmann and Thomas, 1975). This model considers personality types on two axes: *cooperativeness* and *assertiveness*, as shown in the left two columns of Table 2. This model is often used to evaluate group recommender systems (e.g., (Rossi et al., 2017), (Nguyen et al., 2019)).

In the simulation experiments, we set a higher threshold for the personality types with high *assertiveness*. For the personality types with high *cooperativeness*, we set a higher concession factor and threshold decay.

4.2.4 Simulation Conditions

We set up three users (u_A, u_B, and u_C) for the simulation experiments. Their initial requirements are shown in Table 3. These values were set to be relatively strict so that an agreement is less likely to be reached in the first round. Among the three users, u_A has the strictest requirements and u_C has the least strict requirements. All weights ($w_j^{(u_k)}$) used to calculate the utility values were set to 1.

A user is supposed to accept a recommended item if the utility value of the item for the user equals or

Table 1: Defined attributes for describing items.

Attribute		Description	Value range
	price	amount of money expected to be needed	1000 – 10000
	distance	distance from the nearby airport	10 – 120
review score	access	how easy to access	1 – 5
review score	landscape	how beautiful its landscape is	1 – 5
review score	crowdedness	how not crowded it is	1 – 5
review score	barrier free	how easy it is for people with disabilities to visit	1 – 5

Table 2: User personality type and parameters.

Personality type	*Cooperativeness*	*Assertiveness*	*Initial threshold*	*Threshold decay*	*Concession factor*			
					price	distance	review	*ratio*
collaborating	high	high	0.8	0.01	250	10	0.5	0.5
accommodating	high	low	0.7	0.01	100	10	0.25	1.0
competing	low	high	0.9	0	0	0	0	0.0
avoiding	low	low	0	0	–	–	–	–

Table 3: Initial requirements of users in the simulation.

Attribute		User		
		u_A	u_B	u_C
	price	500	500	1000
	distance	10	20	20
review score	access	4	4	4
review score	landscape	5	4	4
review score	crowdedness	5	5	5
review score	barrier free	5	5	4

exceeds the user's threshold. To reflect the personality types in the simulation model, the initial threshold was set according to the personality types as shown in Table 2. In addition, the threshold is decreased each time the item is presented by the amount specified as *threshold decay*, which is set to 0.01 for types whose *cooperativeness* is high (Table 2). For other personality types, this value is set to 0, meaning that the threshold does not change.

4.2.5 Requirement Concession

The system asks a user to relax the requirements for a particular attribute when the user does not accept the recommended item. The user may reject this request or relax the requirement according to the personality type. In the simulation experiments, when the system asks to relax the requirement without providing a *hint*, the user is supposed to change the requirements as specified in Table 2. For example, if the user's personality type is *collaborating*, and they are asked to relax the requirement of price attribute, they increase the requirement value by 250, and if they are asked to relax the requirement of access, which is one of the review attributes, they decrease the requirement value by 0.5.

When the system asks the user to relax the requirement with a *hint* specifying the amount of concession to make, a user should relax the requirement by the *hint* amount multiplied by a *concession factor ratio*, which is specified for each personality type as shown in Table 2. The *accommodating* user relaxes the requirement as suggested by the system, whereas the *collaborating* user only relaxes the requirement for half of the amount suggested by the system, and the *competing* user ignores the request and does not change the requirement. Note that for the *avoiding* user, no concessions are needed as they accept any item.

We assigned one personality type from four possible types to each user. Since there were three users (u_A, u_B and u_C) and four personality types, there are 64 ($= 4 \times 4 \times 4$) cases to consider. For each case, we ran a simulation of both conversation strategies with and without *hints* when the system asks a user for a concession about a particular attribute. In addition, the system is supposed to recommend the same item at most twice ($L = 2$) in the simulation experiments.

4.3 Results

Among 64 cases, agreement was reached for 27 cases for the conversation strategy without *hints* and the same 27 cases for the strategy with *hints*. Cases with even one *competing* user failed to reach an agreement in the simulation experiments. For the 27 cases with an agreement, we plotted the number of rounds before the agreement is reached for two strategies (Figure 3). As seen in this chart, for about half of these cases, the number of rounds was less when a *hint* was used when requesting the concession. The effect of a *hint* is large especially for cases (p1, p2, ..., p15) that involve at least one *collaborating* user except where the only *collaborating* user is not u_C, which has the least

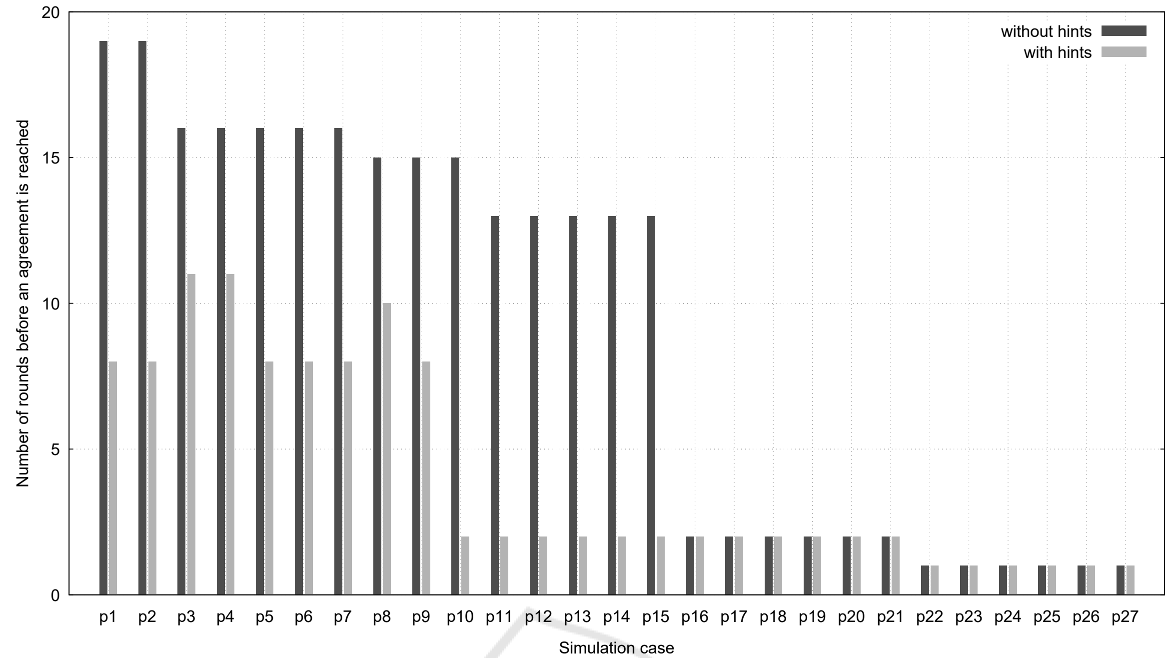

Figure 3: Effect of giving *hints* on the number of rounds before an agreement is reached for the 27 agreed cases.

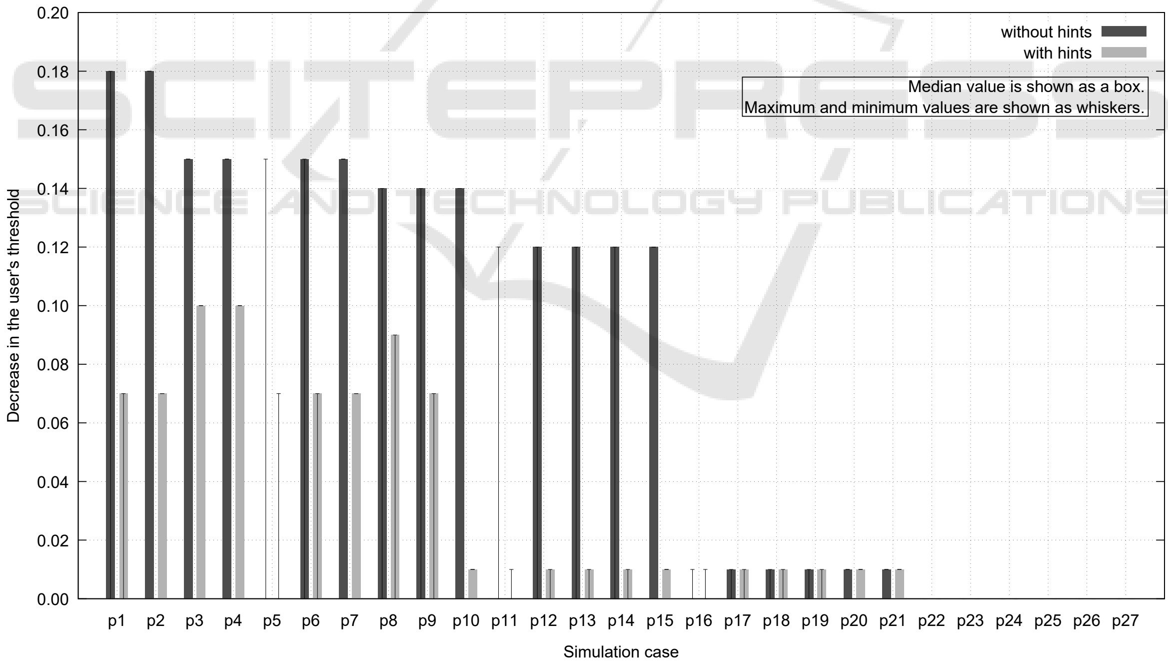

Figure 4: Effect of giving *hints* on the decrease in the user's threshold when an agreement is reached for the 27 agreed cases.

strict initial requirements. This reflects that a *collaborating* user with strict initial requirements tends to need to make more concessions to reach an agreement, and *hint* is effective for such cases.

To evaluate the quality of the solution obtained as an agreement, we examined how much the user's threshold was decreased when the agreement was reached compared with the initial threshold. Figure 4 shows the median value of the difference between the final and initial thresholds for the three users in each simulation case. This chart also shows the maximum and minimum of the difference as whiskers. As seen

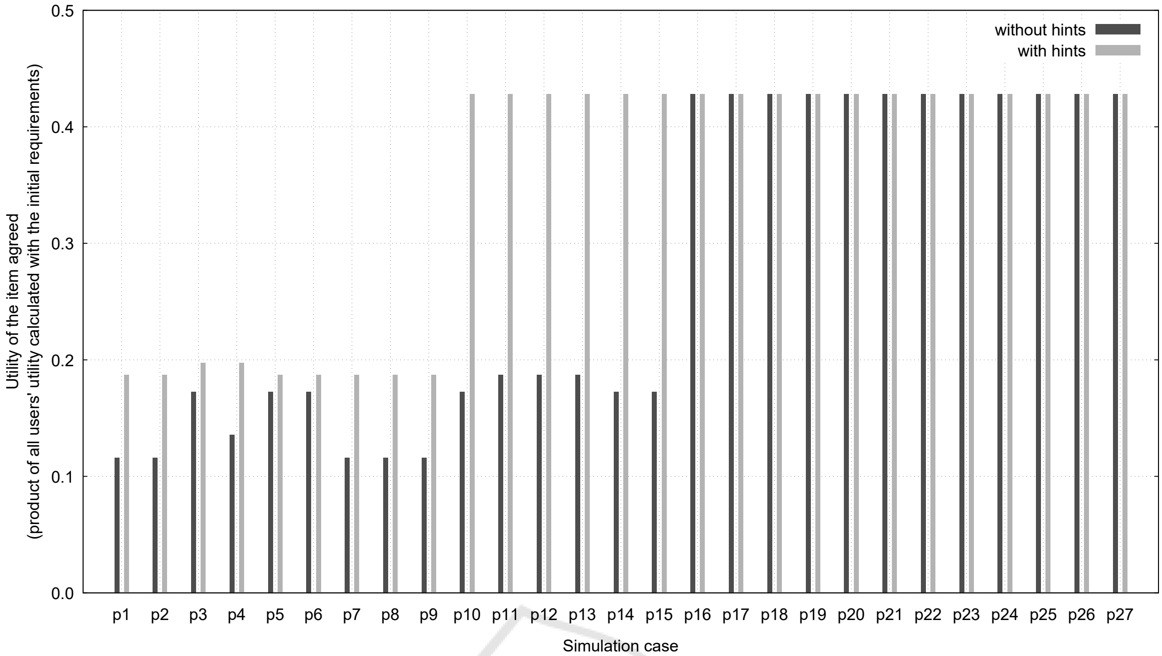

Figure 5: Effect of giving *hints* on the utility of an agreed item for the 27 agreed cases.

in this chart, the reduction in the number of negotiation rounds naturally leads to less concession required in the negotiation process (that is, less decrease in the threshold value compared with the initial one). Note that since the initial threshold of an *avoiding* user is set to 0, an *avoiding* user's threshold cannot be decreased further. Thus, in a simulation case that involves at least two *avoiding* users, the median value shown in the chart is inherently 0, even when the maximum difference is greater than 0 (p5, p11 and p16).

In addition, we calculated a utility value (the product of all users' utility calculated with the initial requirements) of the agreement reached in these cases, as shown in Figure 5. The figure demonstrates that when agreement is reached in fewer rounds, the utility value tends to be higher. This indicates that even with fewer rounds, the quality of the solution (or agreement) does not decrease. The results also indicate that giving *hints* as a conversation strategy is generally effective from the perspective of both the number of rounds before reaching agreement and the quality of the obtained solution.

5 CONCLUSION

This paper described an interactive group recommender system where agents that correspond to users are placed inside the system. Each agent is expected to hold the estimated values of the corresponding user's profile and is used to conduct virtual negotiation to find a recommended item.

The characteristics of the proposed system were examined through simulation experiments that introduced four personality types of negotiation and two conversation strategies. By adding *hints* when the system asks a user to relax a requirement, the number of items to be presented before reaching the agreement can likely be reduced while the quality of the agreement is maintained.

There is much work to be done. For example, currently when a user rejects the recommended item, the system only asks them to relax a requirement. If the system can ask the user why they rejected the item and make use of the response to find another item to be recommended, an agreement could be reached faster. We may also need to consider how to select the item to recommend. Currently, the item that maximizes the product of the users' utility values is selected, but other types of social welfare functions could also be used. Finally, we only simulated a three-user case. We plan to increase the number of users to see the effect of different compositions of user personality types.

REFERENCES

Bekkerman, P., Kraus, S., and Ricci, F. (2006). Applying cooperative negotiation methodology to group recommendation problem. In Felfernig, A. and Zanker, M., editors, *Proceedings of the ECAI 2006 Workshop on Recommender Systems*, pages 72–75.

Endriss, U. (2006). Monotonic concession protocols for multilateral negotiation. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 392–399, New York, NY, USA. ACM.

Felfernig, A., Boratto, L., Stettinger, M., and Tkalčič, M. (2018). *Group Recommender Systems – An Introduction*. Springer, Cham.

Garcia, I., Sebastia, L., Pajares, S., and Onaindia, E. (2011). Approaches to preference elicitation for group recommendation. In Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., and Apduhan, B. O., editors, *Computational Science and Its Applications - ICCSA 2011*, pages 547–561, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kilmann, R. H. and Thomas, K. W. (1975). Interpersonal conflict-handling behavior as reflections of jungian personality dimensions. *Psychological Reports*, 37(3):971–980.

Nguyen, T. N., Ricci, F., Delic, A., and Bridge, D. (2019). Conflict resolution in group decision making: insights from a simulation study. *User Modeling and User-Adapted Interaction*, 29(5):895–941.

Ricci, F., Rokach, L., and Shapira, B., editors (2015). *Recommender Systems Handbook, Second Edition*. Springer US.

Rosenschein, J. S. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press.

Rossi, S., Di Napoli, C., Barile, F., and Liguori, L. (2017). A multi-agent system for group decision support based on conflict resolution styles. In Aydoğan, R., Baarslag, T., Gerding, E., Jonker, C. M., Julian, V., and Sanchez-Anguix, V., editors, *Conflict Resolution in Decision Making*, pages 134–148, Cham. Springer International Publishing.

Villavicencio, C., Schiaffino, S., Diaz-Pace, J. A., Monteserin, A., Demazeau, Y., and Adam, C. (2016). A MAS approach for group recommendation based on negotiation techniques. In Demazeau, Y., Ito, T., Bajo, J., and Escalona, M. J., editors, *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, pages 219–231, Cham. Springer International Publishing.

Yodogawa, D. and Kuwabara, K. (2019). Co-exploring a search space in a group recommender system. In Nguyen, N. T., Gaol, F. L., Hong, T.-P., and Trawiński, B., editors, *Intelligent Information and Database Systems*, pages 264–274, Cham. Springer International Publishing.

Zeuthen, F. (1930). *Problems of Monopoly and Economic Warfare*. Routledge.