A New Method for Ranking Hotels Based on Group Decision Making in Honeybee Swarms

Takuya Arai, Tomonori Furuta, and Kazuyuki Ito
Hosei University

Abstract

Currently, studies on group decision making are attracting considerable attention in many fields. With the increasing use of the Internet, there is an abundance of information available easily to us. However, it is difficult to evaluate accurately the reliability of such information, because much of it is colored by preferences and bias or opinion. It is therefore of interest to develop a group decision-making algorithm that can make appropriate decisions by processing this kind of information. Group decision-making algorithms are employed in many situations, a typical example being the evaluation of a hotel's overall quality. However, current methods of evaluating hotels are not always reliable, because a guest may not be able to evaluate all hotels, and different guests have different preferences, or "inclinations." As such, it is difficult to select the best hotel from a review of hotel rankings. In this paper, in an attempt to solve this problem, we focus on decision making that takes place in a honeybee swarm to select a nesting site. Although each bee has different inclinations, the honeybee swarm is capable of choosing the best nesting site. We applied this advantage to generate hotel rankings. We then conducted simulations that assumed practical application and confirmed that reviewers agreed on the top hotels, and the reliability of the top rankings improved, at the expense of decreased reliability of the lower rankings.

1. Introduction

Currently, much research is being carried out on group decision making, and various algorithms [1-13] involving applications of fuzzy theory and Markov chain models have been proposed. However, these studies have been carried out on small groups of only 4–5 people, which is a very small number to apply the method to Web-based applications, because, with a popular Web site, the size of its user base is much larger, and such groups continue to grow with time. Finding hotels, restaurants, and other locations on the Web is a typical application. In such cases, users tend to rely on the reviews of previous guests, in addition to information from the official Web sites of such places. In this context, review-based rankings are important sources of information. Usually, such rankings are generated by calculating the average score in certain categories, such as the room size, site location, and available facilities. These rankings are intuitively understandable and are widely used.

However, disagreements may arise, as reviewers have different preferences. In addition, no reviewer can evaluate every candidate, and thus, a fair comparison is impossible.

To solve this problem, we focus on the behavior of honeybees when searching for a new nesting site. A honeybee swarm searches for a new nesting site by employing hundreds of scout bees. Each scout bee has a different preference, or "inclination," and visits only a few candidate locations. Nevertheless, the swarm can select the best site based on the recommendations of these hundreds of scout bees [14].

In this paper, we employ hotel rankings as a typical example, and apply an algorithm based on honeybee behavior to this example. To demonstrate the effectiveness of this algorithm, we conduct simulations and compare its ranking results with those obtained using a conventional method.

2. Problem domain

We consider the ranking of hotels and employ five factors for evaluation: dinner menu, room comfort, accommodation fee, room service, and spa conditions. In the simulation, the score for each factor is set randomly from 1 to 5. To express the inclination of reviewers, each reviewer is given a weight for their evaluation of each factor. Each weight is set randomly as a real number from 0 to 1. The evaluated score of the *x*-th factor of the *y*-th hotel by the *z*-th reviewer is given by a product of the score for the factor, $(H_x(y))$, and the weight of the reviewer, $(w_i(z))$, as shown below:

$$E_x(y, z) = H_x(y) \times w_x(z) \quad (1)$$

To investigate the effectiveness of the honeybee ranking algorithm, we compare its rankings with two others, conventional ranking and "desired" ranking. In conventional ranking, each guest evaluates a few hotels, and the ranking is generated on the basis of the average scores given by several guests. In desired ranking, each guest evaluates all the hotels, and the ranking is generated on the basis of the average scores given by all guests.

In the real world, it is impossible to generate the desired ranking. Our aim is to generate a ranking similar to the desired ranking using less information, and in this study, we investigate the performance of the honeybee-swarm-based algorithm as a means to achieving this.

3. Decision making in a honeybee swarm

3.1. Summary of the honeybee algorithm

In this section, we explain the decision-making algorithm used by honeybee swarms, as demonstrated by Thomas D. Seeley, P. Kirk Visscher, and Kevin Passino [15].

Scout bees play two roles: explorer and observer. First, hundreds of scout bees travel to candidate sites, and if a site is of a certain quality (i.e., it is over a predetermined threshold), they evaluate the site and then return to the cluster. They then perform dances for observer bees to express their site evaluation. After dancing, explorer bees revisit the site and return to the cluster repeatedly. Observer bees watch their dances in the cluster and some of them are recruited by the dancing scout bees. The recruited bees visit the site and evaluate it themselves before returning to the cluster, where they then dance as described above. Each time the bee returns to the cluster, the dance "strength" decreases until finally it reaches zero. Subsequently, the bee rests, and then it becomes an observer. When a certain number of bees concentrate on a site, the site is selected as a new nest site.

Equations for this decision-making process [15] are shown below. Table 1 defines the notation used in this paper.

The evaluation value of site *j* at time step *k* by bee *i* is given by

$$S_i^j(k) = N_j \cdot w_i(k) \quad (1)$$

The dance strength of bee *i* at time step *k* is given by

$$L_i(k) = \gamma \cdot S_i^j(k) - r_c \cdot \epsilon_s \quad (2)$$

The probability that an observer watches a dance is given by

$$p_o(k) = 1 - \exp\left(-\frac{1}{2} \cdot \frac{L_t(k)^2}{\sigma^2}\right) \quad (3)$$

The probability that a bee is recruited by a dancing bee *i* is given by

$$p_i(k) = \frac{L_i(k)}{\sum_{i=1}^{B_c(k)} L_i(k)} \quad (4)$$

Table 1 Notation used in this paper

Symbol	Definition
k	Time step
N^j	Quality of site *j*
w_i	Scout bee *i*'s evaluation noise
S_i^j	Site *j*'s score by scout bee *i*
ε_t	Quality threshold that decides whether a bee evaluates the site
γ	Initial dance strength
$L^i(k)$	Dance strength of scout bee *i* at time step *k*
r_c	Number of times a bee has returned to the cluster of a bee
ε_s	Dance decay rate
p_m	Probability that a resting bee will seek to observe dances
$p_o(k)$	Probability that a bee observes dances at time step *k*
$L_t(k)$	Total waggle runs at time step *k*
σ	Constant value to decide the probability that a bee observes dances
p_i	Probability that an observer is recruited by a dancing bee
$B_c(k)$	The number of bees dancing in the cluster at time step *k*
ε_q	Quorum threshold to decide best site

4. Application to hotel rankings

We apply the honeybee algorithm to the ranking of hotels. Table 2 summarizes the correspondence between honeybee behavior and the evaluation of a hotel. We denote *k* as a time step of one week in this algorithm. p_m denotes the probability that a guest travels again at each time step. p_o denotes the probability that a guest refers to reviews on a Web site. p_i denotes the probability that a guest visits the hotel based on the review. Table 2 summarizes the notational correspondence between the honey-bee algorithm and the hotel-ranking algorithm.

Fig. 1 shows a flowchart of the hotel ranking algorithm. A guest (correspond to an explorer bee) chooses a hotel of a certain quality (larger than the threshold ε_t) on the basis of the official Web site of the hotel. After his or her stay at the hotel, the guest posts an evaluation score on the Web site. As each guest has his or her own inclination, the guest's evaluation score of a hotel is denoted as equation (1). The guest returns home, and after a time step passes, he or she travels again with probability p_m.

Next, we describe a guest that selects a hotel based on its review score (corresponding to an observer bee). Each guest refers to reviews on the Web site with probability p_o, as shown by equation (3). Thus, if there are many high-quality hotels on the Web site, the number of guests that refer to the reviews will increase.

After a guest reviews the score $L_i(k)$, the guest chooses a hotel to visit based on the review score. The probability that a hotel is chosen on the basis of the ith review is given by p_i of equation (4). Thus, guests tend to visit hotels that have higher scores.

The guest stays at the selected hotel, and submits his evaluation of it. The review score is posted on the Web site, and after a time step, the guest will travel again with probability p_m.

The review score decreases every time step, as shown by equation (2).

By repeating these processes, a hotel that is of good quality gathers many guests, and, if the number of guests reaches a certain threshold, this process is resumed and a ranking of the hotels is generated on the basis of the number of visiting guests.

Table 2 Correspondence between the honeybee algorithm and hotel-ranking algorithm

Honey bee algorithm		Hotel ranking algorithm
Candidate site for nest	⟷	Hotel
Evaluation of scout bee	⟷	Evaluation of guest
Time step	⟷	One week
Site's quality	⟷	Hotel's quality
Each bee's threshold	⟷	Each guest's threshold
Initial dance strength	⟷	Initial hotel score
Dance strength of scout bee	⟷	Guest's evaluation score of hotel
Decay rate of dance strength	⟷	Decay rate of each guest's evaluation score of hotel
Probability that a resting bee will seek to observe dances	⟷	Probability that each guest Travel again
Probability that observer observes dances	⟷	Probability that guest refers to reviews
The number of waggle runs in a time step	⟷	Total of all hotels' scores at a time step
Probability that a observer recruited by a dancing scout bee	⟷	Probability that a guest chooses a hotel to visit based on reviews
Number of bees that have danced in the cluster	⟷	The number of guests that have posted a review
Number of bees that visited a site over a time step k	⟷	Number of guests who visited a hotel over one week

Table 3 Notation used in our algorithm

symbol	Honey bee	Hotel ranking
k	Time step	Time step (one week)
N_j	Quality of site j	Hotel j's quality
w_i	Scout bee i's evaluation noise	Guest i's noise of evaluation
ε_t	Quality threshold that decides whether a bee evaluates the site	Quality threshold that decides whether a guest visits the hotel
S_i^j	Scout i's evaluation of site j	Guest i's evaluation of hotel j
γ	Initial dance strength	Initial score of a hotel
$L_i(k)$	Dance strength of scout bee i at time step k	Hotel j's score of guest i at time step k
r_c	The number of times a bee returns to the cluster	Number of time steps elapsed since posting a review
ε_s	Dance decay rate	Decay rate of hotel score of hotel j
p_m	Probability that rester will seek to observe dances on the cluster	Probability that a guest travel again
$p_o(k)$	Probability that a bee observes dances at time step k	Probability that a guest refers to reviews on website
σ	Constant value to decide the probability that a bee observes dances	Constant value to decide the probability that a guest refers reviews on website
$L_t(k)$	Total waggle runs at time step k	Total of hotel's review score at time step k
$p_i(k)$	Probability that an observer is recruited by dancing bee i at time step k	The probability that the hotel is chosen on the basis of i th review at time step k
$B_c(k)$	The number of bees that have danced in cluster at time step k	The number of guests who have reviewed hotels at time step k
ε_q	Quorum threshold to decide the best nest	Quorum threshold to decide the best hotel

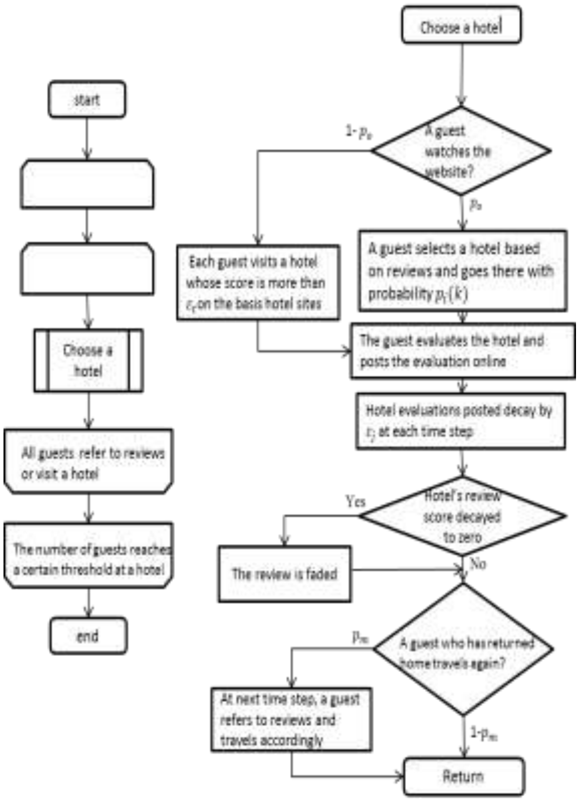

Fig. 1 Flowchart of the process of a guest visiting a hotel

4. Simulations

4.1. Simulation parameters

We conducted 12 simulations with different parameters. Table 4 gives an overview of each simulation, while Tables 5–7 show the chosen parameters in detail, and Table 8 lists the parameters that are fixed throughout the simulations.

Table 4 Overview of simulation parameters

Simulation	Variance of the guests' inclination	Variance of the hotel quality	Probability that a guest refers review
1	small	large	middle
2	small	small	middle
3	large	large	middle
4	large	small	middle
5	small	large	high
6	small	small	high
7	large	large	high
8	large	small	high
9	small	large	low
10	small	small	low
11	large	large	low
12	large	small	low

In simulation 1, the variance of the inclination of guests is small and that of the quality of hotels is large. Thus, this setting is easy to rank. In contrast, in simulation 4, the variance of the inclination of guests is large and that of the quality of hotels is small, and thus, this setting is difficult to rank.

In simulations 5–12, we modify the influence of the probability that a guest refers to reviews, p_o.

Table 5 Parameters for simulations 1–4

	Simulation 1	Simulation 2	Simulation 3	Simulation 4
Constant value to decide p_o	3000			
Average of noise of evaluation for all guests	0.499	0.500	0.499	0.500
Standard deviation of all guests' evaluation noise	0.100	0.010	0.100	0.010
Average of each hotel's factor score	3.003	3.003	2.916	2.916
Standard deviation of each hotel's factor score	0.570	0.570	1.178	1.178

Table 6 Condition setting for simulations 5–8

	Simulation 5	Simulation 6	Simulation 7	Simulation 8
Constant value to decide p_o	2000 (p_o is high)			
Average of noise of evaluation for all guests	0.499	0.500	0.499	0.500
Standard deviation of all guests' evaluation noise	0.100	0.010	0.100	0.010
Average of each hotel's factor score	3.003	3.003	2.916	2.916
Standard deviation of each hotel's factor score	0.570	0.570	1.178	1.178

Table 7 Condition setting for simulations 9–12

	Simulation 9	Simulation 10	Simulation 11	Simulation 12
Constant value to decide p_o	4000 (p_o is low)			
Average of noise of evaluation for all guests	0.499	0.500	0.499	0.500
Standard deviation of all guests' evaluation noise	0.100	0.010	0.100	0.010
Average of each hotel's factor score	3.003	3.003	2.916	2.916
Standard deviation of each hotel's factor score	0.570	0.570	1.178	1.178

Table 8 Fixed parameters

Number of guests	5000
Number of hotels	500
Probability that a guest travels again [%]	10
Each hotel's factor score	1~5
Hotels' initial scores	6
Decay rate of each guest's hotel evaluation score	15 (review score decays to zero over 10 weeks)
Quality threshold that decides whether a guest visits a hotel	10
Quorum threshold to decide best hotel	2500

3.2. Results

Table 9 shows the results of simulation 1. In this table, the first column shows hotels that are arranged according to desired ranking. The second and fourth columns show the ranks of the hotels that are generated by a conventional and the proposed methods, respectively. These ranks are an average taken from 100 simulations, and their standard deviations are shown in the third and fifth columns, respectively.

Table 9 Results of simulation 1

	Simulation 1			
	Previous method		Inspired by Honey bee	
Desired ranking	Rank Order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	1.24	0.51	1.00	0.00
2nd ranked hotel	2.02	0.65	2.00	0.00
3rd ranked hotel	3.28	0.87	3.42	0.49
4th ranked hotel	3.52	0.57	3.56	0.50
5th ranked hotel	5.50	0.70	5.40	0.53
55th ranked hotel	55.54	7.87	54.04	15.17
56th ranked hotel	55.72	10.28	56.28	12.90
57th ranked hotel	59.50	10.35	57.08	18.32
58th ranked hotel	59.96	9.31	58.32	19.18
59th ranked hotel	60.70	9.14	60.32	15.88
60th ranked hotel	60.92	9.58	62.22	14.66

Table 9 shows that the proposed algorithm is more reliable than the conventional algorithm for higher ranks. This is because the calculation resources concentrate on generating the top ranking, and as a result, the reliability at the top is improved at the expense of the reliability of the lower rankings. This tendency is observed in all simulation results. The proposed method is superior to the conventional method, particularly so in simulation 4, where the parameters make it hard to generate a reliable ranking.

Table 10 Results of simulation 2

	Simulation 2			
	Previous method		Inspired by Honey bee	
Desired ranking	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	3.08	2.48	1.14	0.35
2nd ranked hotel	3.22	2.24	1.90	0.41
3rd ranked hotel	4.08	2.40	2.98	0.24
4th ranked hotel	5.28	2.94	5.06	1.36
5th ranked hotel	6.88	3.54	6.14	2.10
55th ranked hotel	59.24	9.95	57.16	14.94
56th ranked hotel	59.50	8.06	57.20	17.17
57th ranked hotel	59.82	10.42	59.78	17.72
58th ranked hotel	62.04	9.22	59.98	14.59
59th ranked hotel	63.08	10.67	61.04	13.77
60th ranked hotel	63.58	10.00	61.18	13.83

Table 11 Results of simulation 3

	Simulation 3			
	Previous method		Inspired by Honey bee	
Desired ranking	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	4.72	4.69	1.10	0.30
2nd ranked hotel	4.84	3.83	2.34	1.09
3rd ranked hotel	6.32	4.84	3.56	0.90
4th ranked hotel	6.44	8.20	3.74	1.61
5th ranked hotel	7.44	7.60	7.76	2.93
55th ranked hotel	69.16	40.57	83.12	55.90
56th ranked hotel	71.02	32.70	83.38	76.54
57th ranked hotel	71.44	36.73	84.02	65.52
58th ranked hotel	71.54	29.67	86.90	50.72
59th ranked hotel	72.44	33.02	86.92	65.14
60th ranked hotel	72.94	31.34	87.72	48.51

Table 12 Results of simulation 4

	Simulation 4			
	Previous method		Inspired by Honey bee	
Desired ranking	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	2.84	2.53	1.48	0.50
2nd ranked hotel	4.56	3.71	1.52	0.50
3rd ranked hotel	6.10	4.41	5.30	4.10
4th ranked hotel	6.82	5.22	6.02	5.35
5th ranked hotel	8.38	4.74	7.44	3.90
55th ranked hotel	53.4	10.8	59.5	19.9
56th ranked hotel	53.0	12.3	61.5	21.4
57th ranked hotel	57.6	9.9	55.9	19.4
58th ranked hotel	54.9	9.7	61.6	20.7
59th ranked hotel	58.0	9.3	58.9	20.0
60th ranked hotel	60.2	9.2	62.0	20.4

Tables 13–20 show the results of simulations 5–12. In these simulations, we change the probability that a guest refers to reviews to select a hotel. From the results, we can see that, where the probability is higher, the calculation resources are concentrated toward generating higher rankings and the reliability of higher ranking is improved at the expense of the reliability of lower rankings.

Table 13 Results of simulation 5

	Simulation 5			
Desired ranking	Previous method		Inspired by Honey bee	
	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	1.00	0.00	1.00	0.00
2nd ranked hotel	3.00	1.15	2.64	0.69
3rd ranked hotel	3.02	1.05	2.80	0.82
4th ranked hotel	4.12	1.68	3.70	0.70
5th ranked hotel	5.04	1.39	5.26	0.91
55th ranked hotel	56.10	12.56	57.08	15.74
56th ranked hotel	56.60	12.48	58.02	17.70
57th ranked hotel	57.54	11.39	58.80	16.18
58th ranked hotel	57.66	10.68	59.30	14.99
59th ranked hotel	62.10	12.42	59.80	14.58
60th ranked hotel	62.60	9.95	60.70	19.88

Table 14 Results of simulation 6

	Simulation 6			
	Previous method		Inspired by Honey bee	
Desired ranking	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	2.12	1.85	1.48	0.61
2nd ranked hotel	3.22	3.81	1.56	0.50
3rd ranked hotel	5.32	4.32	3.62	0.96
4th ranked hotel	5.88	4.38	3.74	0.77
5th ranked hotel	9.32	7.15	6.82	2.98
55th ranked hotel	63.7	32.4	76.5	49.2
56th ranked hotel	65.8	34.8	76.8	48.8
57th ranked hotel	68.0	31.8	77.5	56.6
58th ranked hotel	68.8	30.8	79.3	58.9
59th ranked hotel	69.3	34.0	83.2	40.2
60th ranked hotel	69.9	33.1	84.7	53.9

Table 15 Results of simulation 7

	Simulation 7			
Desired ranking	Previous method		Inspired by Honey bee	
	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	3.14	2.56	1.08	0.27
2nd ranked hotel	3.48	2.62	2.10	0.54
3rd ranked hotel	5.26	4.18	3.42	1.17
4th ranked hotel	6.16	3.59	4.78	1.35
5th ranked hotel	6.34	5.03	6.10	2.65
55th ranked hotel	58.76	23.48	73.90	62.68
56th ranked hotel	59.48	19.71	74.20	51.17
57th ranked hotel	60.70	24.81	78.38	60.07
58th ranked hotel	62.52	26.73	79.76	67.18
59th ranked hotel	64.18	22.17	79.80	47.13
60th ranked hotel	65.58	22.78	79.80	68.43

Table 16 Results of simulation 8

	Simulation 8			
Desired ranking	Previous method		Inspired by Honey bee	
	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	4.72	5.56	1.00	0.00
2nd ranked hotel	13.64	13.43	4.46	2.84
3rd ranked hotel	13.80	12.00	4.48	3.29
4th ranked hotel	15.04	15.89	4.50	3.37
5th ranked hotel	15.28	14.62	6.52	4.14
55th ranked hotel	81.82	50.86	97.52	94.82
56th ranked hotel	82.36	63.36	97.90	70.69
57th ranked hotel	82.90	47.72	99.92	77.14
58th ranked hotel	83.04	49.22	103.06	72.28
59th ranked hotel	83.90	53.02	104.46	99.82
60th ranked hotel	84.30	60.03	105.98	100.74

Table 17 Results of simulation 9

	Simulation 9			
Desired ranking	Previous method		Inspired by Honey bee	
	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	1.00	0.00	1.00	0.00
2nd ranked hotel	2.22	0.58	2.00	0.00
3rd ranked hotel	4.26	1.34	3.84	1.07
4th ranked hotel	4.34	1.34	3.90	0.90
5th ranked hotel	4.96	1.91	4.80	1.02
55th ranked hotel	56.52	10.66	57.52	16.32
56th ranked hotel	58.08	12.20	58.12	17.90
57th ranked hotel	58.16	11.80	58.48	19.14
58th ranked hotel	58.84	10.53	58.72	16.17
59th ranked hotel	59.26	10.46	59.24	19.07
60th ranked hotel	59.28	9.26	59.30	15.25

Table 18 Results of simulation 10

	Simulation 10			
Desired ranking	Previous method		Inspired by Honey bee	
	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	2.50	1.57	1.50	0.57
2nd ranked hotel	3.14	2.16	1.56	0.54
3rd ranked hotel	3.26	1.62	3.00	0.35
4th ranked hotel	5.12	3.59	4.14	0.45
5th ranked hotel	6.08	3.49	6.18	3.49
55th ranked hotel	57.42	17.25	59.30	31.31
56th ranked hotel	58.18	17.40	61.46	33.54
57th ranked hotel	60.10	17.44	61.50	33.65
58th ranked hotel	60.20	16.54	61.60	32.61
59th ranked hotel	60.34	17.62	61.62	30.75
60th ranked hotel	61.10	21.92	64.66	25.26

Table 19 Results of simulation 11

	Simulation 11			
Desired ranking	Previous method		Inspired by Honey bee	
	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	3.10	2.61	1.02	0.14
2nd ranked hotel	4.44	3.27	2.28	0.85
3rd ranked hotel	5.06	3.30	3.74	1.13
4th ranked hotel	7.76	8.33	4.26	1.29
5th ranked hotel	8.44	8.30	5.14	1.83
55th ranked hotel	58.96	20.98	68.2	34.5
56th ranked hotel	59.48	28.10	69.4	77.3
57th ranked hotel	59.84	26.49	73.2	71.2
58th ranked hotel	59.96	28.42	77.4	64.5
59th ranked hotel	60.12	23.82	78.7	85.5
60th ranked hotel	62.32	27.35	79.4	41.1

Table 20 Results of simulation 12

	Simulation 12			
Desired ranking	Previous method		Inspired by Honey bee	
	Rank order	Standard deviation	Rank order	Standard deviation
1st ranked hotel	5.06	5.90	1.16	0.58
2nd ranked hotel	7.42	6.92	2.74	1.04
3rd ranked hotel	9.20	9.26	3.58	1.97
4th ranked hotel	9.28	9.37	4.46	1.96
5th ranked hotel	9.70	8.27	4.92	2.04
55th ranked hotel	76.0	49.0	101.0	78.8
56th ranked hotel	77.0	51.5	103.3	70.7
57th ranked hotel	79.4	52.1	103.8	94.5
58th ranked hotel	79.5	53.0	105.8	88.9
59th ranked hotel	81.8	53.2	108.3	103.4
60th ranked hotel	82.4	60.5	108.8	93.7

From these results, we can conclude that the honeybee algorithm is effective for generating top rankings when the number of reviewers is limited. The reason for this is that guests concentrate on relatively good hotels, and therefore, the accuracy of higher rankings is improved at the cost of deterioration in the accuracy of lower rankings.

In addition, we find that if the probability that a guest refers to reviews is high from the start, then in the initial stages of the simulations, guests visit the top-rated hotels and do not visit lower-quality hotels, so guests concentrate on evaluating the best hotels. On the other hand, if this probability is low, guests do not visit the top-rated hotels in the initial stages, and so there is a probability that they will visit lower-quality hotels.

In these simulations, we ignore the budget of guests, and as such, every guest tends to concentrate on the higher-ranked hotels. In future work, we will take this constraint into consideration.

5. Conclusion

In this paper, we attempted to improve the accuracy of a hotel ranking system by applying an algorithm based on the behavior of honeybees. Simulations were conducted to investigate the performance of the honeybee algorithm, and the results confirmed the effectiveness of the honeybee algorithm for the top-ranked hotels.

References

[1] Hojjat Mianabadi, Abbas Afshar, "Fuzzy Group Decision Making to Select the Best Alternative for Development of Groundwater Resources," Fuzzy Systems and Knowledge Discovery, FSKD '09, pp307-311(2009)

[2] Ron Chi-Wai Kwok, Jan Ma, Duanning Zhou "Improving group decision making: a fuzzy GSS approach," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, pp54-63(2002)

[3] Tundjungsari, V., Istiyanto, J.E., Winarko, E., Wardoyo, R., "A reputation based trust model to seek judgment in participatory group decision making," Distributed Framework and Applications (DFmA), pp1-7(2010)

[4] Yan Hong-yan, Zhang Fei-lian, "Research on multiple attribute group decision-making approach for construction programme choice," Information Science and Engineering (ICISE), pp1-5(2010)

[5] Huizhang Shen, Jidi Zhao, "A quick group decision-making planning method for emergency response," Intelligence and Security Informatics, pp25-31(2007)

[6] Wu Li, Wanan Cui, Yan Chen, Yingzi FuA, "Group decision-making model for multi-criteria supplier selection in the presence of ordinal data," Service Operations and Logistics, and Informatics, IEEE/SOLI 2008, pp1686-1690(2008)

[7] Pérez, I.J., Cabrerizo, F.J., Herrera-Viedma, E., "A mobile decision support system for dynamic group decision-making problems," Systems, Man and Cybernetics, Part A: Systems and Humans IEEE Transactions, pp1244-1256(2010)

[8] Herrera-Viedma, E., Martinez, L., Mata, F., Chiclana, F., "A consensus support system model for group decision-making problems with multigranular linguistic preference relations," Fuzzy Systems, IEEE Transactions, pp644-658(2005)

[9] Quan Zhang, Dan Shao, Wen Jia Feng, "Group decision making with interval judgment matrix," Control and Decision Conference, CCDC 2008, Chinese, pp2189-2194(2008)

[10] Camilleri, G., Zarate, P., Viguie, P., "A timing management banner for supporting group decision making," Computer Supported Cooperative Work in Design (CSCWD), pp551-555(2011)

[11] Lei Yang, Mingyue Jiang, "Dynamic group decision making consistence convergence rate analysis based on inertia particle swarm optimization algorithm," Artificial Intelligence, JCAI '09, pp492-496(2009)

[12] Gao Yan, Chenchen Liu, Zhuyan Shao, "Analysis of influencing factors for the grey multi-attribute group decision making," GSIS, pp1081-1086(2009)

[13] Ming Xie, Xianjun Cui, "Official selection model based on multi-attribute group decision making," Future Information Technology and Management Engineering (FITME), pp80-83(2010)

[14] Madeleine Beekman, Robert L. Fathke, Thomas D. Seeley, "How does an informed minority of scouts guide a honey bee swarm as it flies to its new home?" Animal Behavior, Vol. 71, pp161-171(2006)

[15] Kevin Passino, Thomas D. Seeley, P. Kirk Visscher, "Swarm cognination in honey bees," Behavioral Ecology Sociobiology, pp401-414(2008)

[16] Thomas D. Seeley, P. Kirk Visscher, Kevin Passino, "Group decision making in honey bee swarms", American Scientist, Vol. 94 pp220-229(2006)

[17] Nigel R. Franks, Stephen C. Pratt, Eamonn B. Mallon, Nicholas F. Britton, David J. T. Sumpter, "Information flow, opinion polling and collective intelligence in house-hunting social insects," Phil. Trans. R. Soc. Lond.B357, pp1567-1583(2002)