

Exploring Social Behaviour of Honey Bees Searching on the Web

Pavol Návrat, Lucia Jastrzemska, Tomáš Jelínek, Anna Bou Ezzeddine, Viera Rozinajová
Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
navrat@fiit.stuba.sk, ljastrzemska@gmail.com, tj.jelinek@gmail.com,
ezzedine@fiit.stuba.sk, rozinajova@fiit.stuba.sk

Abstract

This paper discusses applying the social behaviour of bees to the web search. We proposed an on-line search of the user's predefined group of pages. In particular, this approach is based on our model of a bee hive being augmented by a model of the behaviour of bees outside the hive and by the method of assigning the page quality. With regard to the advantages of this approach, the hive as a whole seems to be able to determine the best routes of the search and reject the bad ones. This has been indicated by our first exploration tests as we report in the paper. However, a comprehensive experimentation is necessary before definitive conclusions can be made.

1. Introduction

The majority of search engines used nowadays try to find out a universal answer to the given query, resulting usually in far too many answers. As they usually do not take into consideration for whom the search is performed, one promising way for the near future is for the search engines to take into account the user, his model and preferences. Another possible way of increasing quality of the answers is to improve the search process itself. This is one of the reasons for proposing a new approach to information search on the Internet, which we believe is able to follow good paths in the graph of web, allows to evaluate the quality of web page according to several points of view and is based on the social behaviour of honey bees. Most of the search engines work with off-line databases of indexed pages from the part of the Internet. Although we try to keep them up to date, there is still the possibility, that the database has changed since the last update. As the fast Internet connection is getting more accessible to the common users, we suppose that it might be possible to create the on-line web search engine, which would run on the end user's computer.

Apparently, if we decided to make an on-line search engine, it would be impossible to search the whole Internet. But where to start? We proposed to start on the user predefined sites which would reflect his/her preferences. Let's imagine the Internet as an oriented graph with vertices as web pages and edges as hypertext connections between the sites. Thus we need an algorithm, which would be able to follow the pages with the high quality in the graph and to reject the bad ones. Nowadays, most of the web crawlers do not have additional information about the others. But if we want to search only the good paths, our web crawlers need to communicate. While we tried to solve this problem, we found an interesting metaphor in the nature, particularly in the social behaviour of honey bees (*Apis mellifera*). A bee as an individual knows only partial information about its environment, but a hive as a whole can choose the best (or sufficiently good) of all sources in its neighbourhood. We take advantage of this property in our proposal of information search engine. The rest of the paper is structured as follows: in the second section we briefly discuss two popular approaches to the information search on the Internet. Bee hive model is introduced in the section 3 and the proposed modification of this model is presented in the section 4. The next section is devoted to the calculation of web page quality. We present some exploratory experiments which we performed with this model in the section 6.

2. Related works

The searching for the information on the Internet has been a challenge for many years. The most successful algorithm for evaluating the page relevance has been PageRank [1] which provides the basis for Google search engine. PageRank uses the vast link structure of the Web as an indicator of an individual page's value. However, Google does not differ between various topics the user is really interested in. Another approach is described in [4] as a Focused Crawler which seeks, acquires, indexes, and main-

tains pages on a specific set of topics that represent a relatively narrow segment of the Web. The Focused Crawler starts searching from a set of user predefined pages and tries to follow only good paths. The result of the focused crawling is a relatively small database of indexed pages on the same topic. The search is then performed in this database.

3. Bee Hive Metaphor

Various attempts to make a suitable bee hive model have been performed, including [2], [3]. Our work is based on the modification of the previous models [6], [7], [8], [9], by adding dispatch room and introducing an uncertainty. This modified model was applied to alternative calculation of PageRank [10]. We extended this model by adding the behaviour of the bee outside the hive [5].

3.1. Bee Hive Model

The hive in this model consists of the dance room, the auditorium and the dispatch room. The parameters of this model are as follows:

- number of all bees in the hive (NB)
- initial distribution of bees between observers and foragers
- maximal time allowed for dancing for a source (MDT)
- maximal time a bee can spend in the auditorium (MTA)

According to this model, as the search begins, the forager bees leave the hive from the dispatch room and fly to random sources. Afterwards they return to the hive with the estimated quality q of the source they have visited. Inside the hive the bee decides whether to stay with the found source (probability q) or abandon it (probability $1 - q$). If she decides to stay with the source, she can go to the dance room (probability q) and dance for the source. The bigger the quality of the source is, the longer the dance lasts, but not longer than the maximum dancing time. After finishing her dance the bee returns to the source. The other option is to return back to the source without dancing (probability $1 - q$). In case the bee abandons her source, she flies to the auditorium to observe dancing bees. She randomly chooses a dancing bee propagating source. After making this choice, the bee could follow the selected dancing bee (with probability given by the division of the number of bees dancing for the same source and the overall number of bees) or stay in the auditorium and choose another one. The bee cannot stay in the auditorium more than the predefined parameter maximal time in the auditorium. Once this time is elapsed, she has to go to the dispatch room.

4. Modified model

The model described above assumes that all the sources are known at the beginning of the search, so that the bee flies directly to the chosen source and having evaluated its quality, she returns to the hive. Our search method allows that only a small part of pages is known at the time the search begins and the other ones are reachable only from these sites by hypertext links. This opens door for on-line searching. We amended the original model and added a description of bee's behaviour outside the hive. When the bee visits the source (the web page), she calculates its quality q and either with the probability q returns to the hive or with the probability $1 - q$ starts searching for another source (she randomly chooses one of the hypertext links on the page). If the page contains no links, the bee will either return with probability q , with calculated quality q , or she will return with probability $1 - q$ with quality 0. If the bee returned with q in all cases, the page without hyperlinks would be advantaged, because all the bees would return to the hive (unlike the sites containing hyperlinks, where the bees split between those who are returning to the hive and those following some link). This behaviour is shown in the Figure 1.

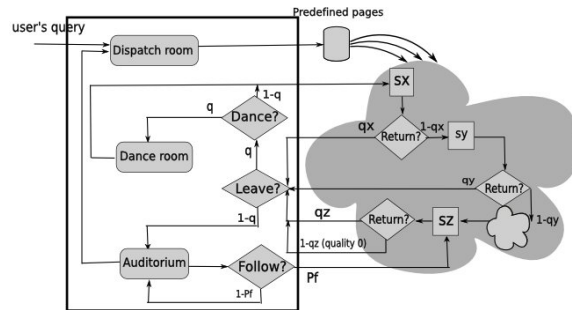


Figure 1. Model of the hive and the bee.

5. Quality calculation

So far we have expected the bee to evaluate the quality of the page but we have given no rules how to do it. The bee hive model implies that the quality is from the interval $< 0, 1 >$, 0 representing the lowest quality, 1 the highest.

There might be several points of view to the page quality, thus we calculate the overall quality of the page as a combination of several partial qualities (quality attributes). Each quality attribute has its defined maximal value. The only constraint is that they should be defined so that the sum of all maximal values is exactly one. This approach allows to add or remove any page quality attribute without the need

of changing the definitions of others (but those to maintain the constraint).

We have proposed three quality attributes of a web page: count, header, distance.

Count. To quantify the page quality depending on the number of user's queries we have to take into account that the quality difference between one and two occurrences of the query must be bigger than between say 20 and 21 occurrences. Therefore the function has to grow fast at the beginning of the domain, then slow down and approach its defined maximum value. We empirically proposed formula (1) to calculate the count quality attribute.

$$q_{count} = \frac{-1}{2(n + \frac{1}{2Q_{COUNT}})} + Q_{COUNT} \quad (1)$$

where n is the number of the found words and Q_{COUNT} is the defined maximum value for this quality attribute.

Header. We can assume that if the page contains the searched query in any of its headers, it is more relevant than the page containing the searched query only in the body text.

HTML defines six tags for describing headers on the page ($\langle h1 \rangle$ to $\langle h6 \rangle$) and one for the title of the page ($\langle title \rangle$). If we assign number 0 to the tag $\langle title \rangle$, number 1 to $\langle h1 \rangle$, ... and number 6 to $\langle h6 \rangle$, we can quantify the occurrence of the user's query in the header tags. If the query is present in more than one tag, the number of its occurrence h is given as a minimum of all numbers assigned to tags containing the query. The bigger the number of the tag containing the query, the less important this occurrence is. Therefore we propose this linear descending function (2).

$$q_{header} = Q_{HEADER} - h * \frac{Q_{HEADER}}{HEADER_{MAX} + 1} \quad (2)$$

Where h is the minimal header number, Q_{HEADER} is the defined maximum value for this quality attribute and the parameter $HEADER_{MAX}$ is a parameter defining the number of the last tag in hierarchy we want to count in the header quality (e.g if it is 2, we consider only tags $\langle title \rangle$, $\langle h1 \rangle$, $\langle h2 \rangle$).

Distance. Since the bees in the nature fly only to a certain distance from the hive, we introduced this partial quality attribute. Let us define a distance between two sites as the number of domains the bee had to visit when getting from the first site to the second one. This quality attribute guarantees that the pages closer to the user predefined group are of higher quality. After reaching the predefined maximal distance $DIST_{MAX}$ the quality attribute falls to zero (3, 4, 5).

$$d > DIST_{max} \Rightarrow q_{dist} = 0 \quad (3)$$

$$DIST_{max} = 0 \Rightarrow q_{dist} = Q_{DIST} \quad (4)$$

$$d \leq DIST_{max} \Rightarrow q_{dist} = Q_{DIST} - d \frac{Q_{DIST}}{DIST_{MAX}} \quad (5)$$

Where d is distance and Q_{DIST} is the defined maximum value for this quality attribute.

6. Exploratory experiments

We made several exploratory experiments to discover how the bee algorithm behaves on the Internet. They are divided into two parts. In the first part we tried to show how this behaviour is influenced by particular parameters and tried to find a suitable configuration. These exploratory experiments were made on the site www.cinema.com. We chose this site because it was dedicated to one specific topic (movies) and it was not too large so we were able to run the search engine on the personal computer (1800 MHz CPU, 512 MB RAM, 2Mbps download speed). In order to see how the parameters effect the search, we used the same randomly chosen keyword (titanic) in all exploratory experiments. In the second part we used the configuration found in the first part and tried to find some information about Lisbon.

6.1. Finding configuration

At first we defined the default values of the parameters of the quality calculation and the hive which were used in all exploratory experiments in this part (Table 1, Table 2). We tested the behaviour of the bee algorithm by changing only one parameter at the time. We consider the page most propagated in the dance room in the long term as the one recommended by the hive. In the exploratory experiments, we show the time when the bees first danced for the recommended page as well as the time when this page started to win. The most interesting information is the interval between these times, because it shows how quick is the hive able to decide. The site www.cinema.com contained several pages containing the word *titanic*, three of them even with the same calculated quality. When the hive recommended any of these three pages we considered the search as successful.

Table 1. Parameters which configurate the quality calculation.

| Q_COUNT | Q_HEADER | Q_DISTANCE | MD | MH |
|---------|----------|------------|----|----|
| 0.7 | 0.2 | 0.1 | 1 | 6 |

Table 2. Parameters which configurate the hive.

| NB | MTD | MTA |
|----|--------|----------|
| 30 | 100 ms | 10 turns |

6.1.1 Maximal Dancing Time (MDT)

We tested values 10ms, 100ms, 1000ms and 10000ms. The summary of this set of exploratory experiments is stated in the Table 3. The hive decided quickly in cases of 10 ms and 10000 ms, but this is not really desired, because the more bees propagate the particular source, the less bees search for new (possibly better) ones.

Table 3. Summary of the exploratory experiments with MDT.

| MDT | best page found | best page winning |
|----------|-----------------|-------------------|
| 10 ms | 3s | 6s |
| 100 ms | 12s | 25s |
| 1000 ms | 10s | 16s |
| 10000 ms | 3s | 5s |

6.1.2 Maximal Time in the Auditorium (MTA)

As you can see in the Table 4, the behaviour of the bee algorithm is unpredictable when the MTA is 1 turn and the hive decides too quickly if the MTA is set to 100 turns.

Table 4. Summary of the exploratory experiments with MTA.

| MTA | best page found | best page winning |
|-----------|-----------------|-------------------|
| 1 turn | 27s | 30s |
| 1 turn | 7s | 20s |
| 10 turns | 14s | 20s |
| 10 turns | 12s | 25s |
| 100 turns | 1s | 3s |
| 100 turns | 7s | 7s |

6.1.3 Number of Bees (NB)

This exploratory experiment is more than the others dependent on the hardware, because every bee is implemented as a single thread. If the CPU time is divided between too much bees the model does not behave as it was defined.

As you can see (Table 5) the bigger is the number of bees, the less time they need to decide.

Table 5. Summary of the exploratory experiments with NB.

| NB | best page found | best page winning |
|----|-----------------|-------------------|
| 10 | 3s | 15s |
| 30 | 14s | 20s |
| 60 | 5s | 8s |

6.1.4 Quality calculation

Since the site www.cinema.com does not point to the other domains too much (for example during one experiment 840 different sites were visited on this domain, but just 24 outside), we decided to use the distance quality 0.1 all the time.

The change between Q_{HEADER} and Q_{COUNT} resulted in the change of the composition of propagated pages. When the higher priority was set to Q_{HEADER} , only pages containing the word titanic in any of the headers were propagated.

Table 6. Chosen parameters for hive configuration.

| NB | MTD | MTA |
|----|---------|----------|
| 30 | 1000 ms | 10 turns |

6.1.5 Summary

The exploratory experiments showed the meaning of the particular parameters. They helped us discover a possible relation, i.e. that only the hive parameters affect the search process whereas the attributes for quality calculation have impact on the composition of the propagated pages. We decided to use such attribute values for the hive which ensure that the hive does not decide too quickly (Table 6). We considered the default quality attributes (Table 1) as satisfying and used them for the next set of the exploratory experiments.

6.2. Discovering Lisbon

In the following exploratory experiment we decided to find some information about one of the European cities - Lisbon. The progress of the exploratory experiment is shown on the graph (Figure 2) where the x axis represents the time line and the y axis represents the number of dancing bees.

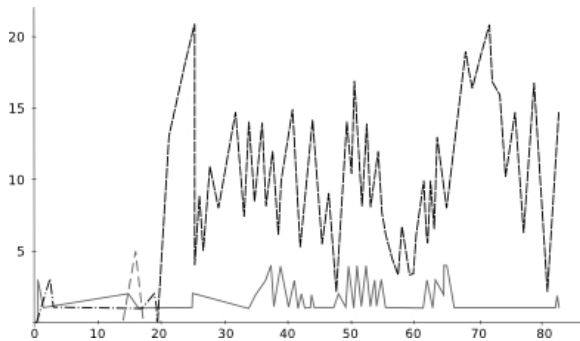


Figure 2. Progress of the search for the touristic information about Lisbon.

6.2.1 Lisbon as a touristic destination

To show that our search engine is able to start search from more than one page we used three starting pages dedicated to the Portugal as a tourist destination (<http://www.portugal.org/tourism>, <http://www.visitportugal.com/Cultures/en-US/default.html>, <http://www.portugal-info.net>). We searched for the word Lisboa. At the beginning of the exploratory experiment several pages were propagated, but not by more than five bees. At 20 seconds bees started to propagate the article about Lisboa which was winning until the end of the exploratory experiment. The hive was able to find the relevant page from three starting sites.

7. Conclusion

We presented a new approach to the web search - an online search engine inspired by the social behaviour of honey bees. The user preferences are taken into account by starting the search from user predefined group of pages. We performed several exploratory experiments on the real Internet which show that the bee hive metaphor is applicable to the web search. Based on these outcomes, in the next step, we plan to demonstrate viability of our approach by performing extensive experimentation.

Acknowledgments. Acknowledgments. This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/3102/06 and the Slovak Research and Development Agency under the contract APVT-51-024604.

References

- [1] S. Brin, L. Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine, Proceedings of the seventh international conference on World Wide Web 7, 1998, **107-117**.
- [2] H.E. Bullock, P. Dey, K.D. Reilly: A "Bee Hive" Model for Heterogeneous Knowledge in Expert Systems. ACM, 1986, **417-417**.
- [3] S. Camazine, J. Sneyd: A Model of Collective Nectar Source Selection by Honey Bees: Self Organization through Simple Rules. Journal of Theoretical Biology, 1991, 149(4): **547-571**.
- [4] S. Chakrabarti, M. van den Berg, B. Dom: Focused crawling: A new approach to topic-specific Web resource discovery (1999) (Make Corrections) (149 citations) Computer Networks (Amsterdam, Netherlands: 1999).
- [5] L. Jastrzemska, T. Jelínek: Bee Inspired Web Search - First On Line Experiments. In: Proc. IIT.SRC 2007, Bratislava 2007, **111-118**.
- [6] F. Lorenzi, S. dos Santos, D. Bazzan, A.L.C.: Negotiation for Task Allocation Among Agents in Case/Base Recommender Systems: A Swarm Intelligence Approach. In: Proc. IJCAI 2005 Conference, Workshop, 2005, **23-27**.
- [7] P. Návrát, M. Kovacik: Web Search Engine as a Bee Hive, In: 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06), Hong Kong, 2006, pp. **694-701**.
- [8] P. Návrát, M.Kovacik, A.Bou Ezzeddine, V.Rozinajová: Information Search Using Bees. In: Znalosti 2007, Ostrava 2007, pp. **63-74**.
- [9] P. Návrát: Bee Hive Metaphor for Web Search. In: CompSysTech 2006, B. Rachev, A. Smrikarov (Eds.), Veliko Turnovo, Bulgaria, June 2006, IIIA.12-1-7.
- [10] P. Návrát, M. Kováčik, A. B. Ezzeddine, V. Rozinajová: Bee Hive Metaphor - a Model for Searching and Recommending Information. In: Proc. Cognition and Artificial Life, Smolenice, Slovakia, April 2007, **249-256**.