The Bee Hive At Work: Exploring its Searching and Optimizing Potential

PAVOL NAVRAT
ANNA BOU EZZEDDINE
LUCIA JASTRZEMBSKA
TOMAS JELINEK

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovicova 3, 842 16 Bratislava 4 - Slovakia
(navrat,ezzeddine)@fiit.stuba.sk
(jastrzembska,jelinek)@gmail.com

Abstract. A model of the bee hive that clearly separates the self-organizing decision-making behaviour of the bees in the hive and the problem-specific behaviour of the bees outside the hive is presented. This separation allows for the applications of the model for different problem domains. The Bee Hive at Work model has been applied to several problem domains. Results of the application to three problem domains are presented - web search, function optimization and hierarchical optimization. In web search, the model has been successful in following a story as it was developing on web sites. Another its application was able to find global optima of various mathematical functions. We explored also to some extent a rather novel idea of building a hierarchical hive.

Keywords: bee hive model, web search, function optimization, hierarchical hive

(Received February 5th, 2012 / Accepted March 25th, 2012)

1 Introduction

Nature serves as an unfailing source of inspiration in general, and more and more in computing in particular. Behaviour of social insects is just one example of natural processes that have gained attraction of computing scientists. Years ago, ant colonies have become popular as model of distributed problem solving, effective for specific problem types, including optimization problems. More recently, bee hives started to be studied in this respect, too. We started to explore possibilities open by considering behaviour of social insects. Social insects (ants, honey bees, termites, wasps etc.) show a remarkable level of social behaviour. In particular, they communicate, albeit in a very elementary way, with each other. For example, they may communicate regarding the location of food sources. They even collaborate towards achieving some goal. In particular, they may collaborate regarding bringing the food back once it is found. They distinguish themselves by their organization skill without any centralized control [9]. The interactions among individuals, between the individuals and the environment along with the behaviours of the individuals themselves allow organization to emerge. Our hypothesis is that the behaviour the bees show might be an instructive inspiration to develop a model of solving problems from a suitable class. More specifically, having developed a model of a bee hive that can work as a kind of search engine [18], we propose to investigate a bee hive as a possibly useful metaphor for optimization. Another kind of problem that appear to be worth exploring is web search.

The rest of the paper is structured as follows. In Section 2, a brief overview of the related work is given. We present the model of the hive that we use in Section

3. Core of our paper is in Section 4, which contains three subsections: Web Search, Function Optimization, and Hierarchical Hive. Finally, in Section 5 we discuss conclusions.

2 Related Work

Honey bees *(Apis mellifera)* have a unique ability to choose the best nectar source of all sources found in the vicinity of the hive [23]. At first, foraging bees search randomly for food sources. When a bee finds a reasonably good source, she returns to the hive and may choose to perform a waggle dance to share the information about the found source (direction, distance). Other bees that watch the dance can decide to fly to the propagated source and become recruits of the dancing bee. After a while, most of the bees are foraging to the best source in the vicinity of the hive. Since the food sources are not constant (new ones appear and old ones become exhausted) the hive needs to employ a mechanism to conform these changes in its relevant environment.

The behaviour of the honey bees inspired various researchers in the fields of biology, mathematics or informatics, each with partly differing objectives. Their interests, however, are basically twofold:

- Modelling of bees behaviour ([8], [6])
- Constructing algorithms inspired by bees behaviour

Our work belongs to the latter category, since our aim is to develop a simple model of the bee hive applicable to solving problems in different domains.

There are several algorithms inspired by the bee behaviour, a good overview can be found in [2] or [12]. These can be divided into two categories. The first category contains algorithms based on the evaluation cycle where the bees do not communicate, all decisions are made by the central unit. These algorithms can have good performance, however, they do not make use of the natural self-organization capabilities of the hive. Examples are [20], [25], [7] applied to the optimization of mathematical functions, [26] applied to solving the traveling salesman problem (TSP) or [10] applied to the Maximum Weighted Satisfiability Problem. A good overview of bee colony optimisation approaches can be found in [24]. In [21], the concept of swarm intelligence is put into a wider context of computational intelligence.

The second category of bee inspired algorithms are multi-agent systems where the individual bees communicate and make own decisions. These algorithms are closer to the biological behaviour and are able not only to make use of decision-making mechanism of the bees but also of some specific aspects of their behaviour (for example the *Failed follower hypothesis* [3] or experiments with the bee behaviour outside the hive [13]).

An important possible application of the bee hive metaphor is to enhance web information retrieval. One of the earlier works [22] focused on web browsing. Later, there started to appear works on web recommendation [14] that influenced also our initial research on web search [16].

3 Model of the Hive

The algorithm described in this work is based on our model of the bee hive [18]. It is a state transition model of the bee behaviour in the hive where every bee makes own decisions (multi-agent system).

The bee according to this model can be in four states - In Dance room, In Auditorium, In Dispatch room and Outside the Hive. The transitions between the states are based on the quality of the bee's current source (Figure 1). The bees propagate their sources in the dance floor, the auditorium is a place, where the bees are able to watch the dancers in the dance floor. The dispatch room contains the sources where the bees can start their search for food (enumerated or randomly generated). The Outside the Hive is a very general state and the exact behaviour of the bee in this state is not defined. It can be specified according to the problem domain.

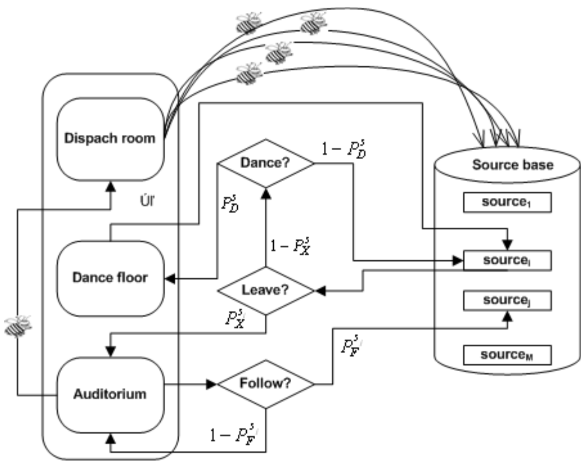

Figure 1: The model of bee hive ([18])

This model has the following set of parameters

- *Number of bees* Number of individuals in the hive
- *Maximal dancing time* Maximal time the bee can stay on the dance floor

- *Maximal time in auditorium* Maximal time the bee can stay in the auditorium
- *Observation error* Probability that the bee observes wrong information about propagated source and consequently visits the source close to the propagated one.

When the algorithm starts, all the bees are in the dispatch room and choose randomly from sources in it. When the bee chooses the source, she flies to visit it and to estimate its quality.

When the bee returns to the hive she can leave the source with probability q (the quality of the specific source in the interval $(0, 1)$) or stay with the source with probability $1 - q$.

When the bee decides to stay with the source she can decide whether to propagate it on the dance floor or revisit it directly. When the bee decides to propagate the source she will dance for the source for the time estimated as the quality of the source multiplied by the maximal dancing time. After this time she will return to her source.

Bees that have abandoned their sources go to the auditorium to observe the dance floor. Every bee chooses a new source to follow with probability equal to the number of bees dancing for the source divided by the number of all dancing bees ($P_f^{S_j}$). With the opposite probability the bee stays in the auditorium. If the bee does not choose any source within the specified time (maximal time in auditorium, beyond which starvation would be imminent) she leaves the auditorium and goes to the dispatch room to choose the source from there.

4 The Applications of the Model

In the previous section we described model of the bee hive. This model is general and does not define the behaviour of the bee outside the hive. This generality brings an opportunity to fine-tune the behaviour of the algorithm to the specific needs of the problem domain without the need to modify the basic behaviour of the hive. The bee hive metaphor can be thus used for such differing tasks as on-line web search or function optimization.

There are only two conditions the bee has to satisfy:

- She has to take a source
- After a while she has to return a source (not necessarily the same) with estimated quality

The generality of the model has allowed us to implement it as a general framework available at [1].

4.1 Web Search

We applied this algorithm successfully to the domain of on-line web crawling. The aim was to appoint the agents to download the quality pages and to use these pages in system for tracking the evolving story (e. g in the newspapers). The bees acted as crawlers, their environment was the Internet and their food sources were web pages. Outside the hive, the bees moved from page to page using the hyper-links searching for new quality pages. Due to their decision making and communication in the hive the bees were able to focus on the more promising sites but kept the ability to dynamically react on the changes in their environment.

When the bee flies outside the hive to the web page, it will estimate its quality and with the probability q it will stay on the current page, with probability $1 - q$ she will follow one of the links on the page to visit the new source. Then she will with probability q fly back to the hive with her current source or with the opposite probability stay outside the hive and search for better sources. The bee cannot stay outside the hive forever, therefore we used the concept of energy. Every time the bee visits some source, the energy will increment by the quality of the source and decrement by the specified parameter. If the bee has no more energy ($energy <= 0$) she will return to the hive.

The model has been applied to the problem of dynamically tracking a developing story [19], [17] with a considerable success. The bees have been able to follow a developing story on the web on-line, recommending web pages containg very relevant material (news, commentaries etc.) as they were emerging.

4.1.1 Quality calculation

The quality calculation is divided into three components. Each of them can reach their maximal value given by the parameters. The sum of these parameters should be equal to 1.

The count quality component counts the query occurences n on the page and using the Q_{COUNT} parameter calculates the quality according to the following formula:

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

The header quality searches all pages headers and page title for query occurence.It then chooses the minimal header number (for case of $< title >$ it was 0 ..., $< h_6 >$ it was 6) and calculates the quality according to the formula:

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

where Q_{HEADER} is the maximal value allowed for header quality, h is the minimal header number and $HEADER_{MAX}$ is the maximal header number we want to take into account.

The Flesch readability index is an integer indicating how difficult the document is to understand. It is computed according to the formula:

$$FI = 206.835 - (1.015 * ASL) - (84.6 * ASW) \quad (3)$$

where ASL is average sentence length (syllables/words), ASW is average number of syllables per word (words/sentences). We compute Flesch's readability index only in the case if at least one of the two before mentined qualities was of a non zero quality. The source obtains this partial quality if Flesch's readability index was from the interval $< 0, 50 >$.

4.1.2 Web story

We used the model described above to perform story tracking, an activity performed upon web on-line. Aim of the on-line search is not the single information, the aim is to find a relevant set of pages which would create a story. This should not be a general search engine. Instead, it is supposed to be used on news portals or any other sites containing frequently changing or added information. The best application is for headline stories where the new information is being added very often.

Thus our aim was to devise a method implementable on a personal computer that would be capable of supplying documents related to the developing story as they are emerging on the web for several hours or days. It seems that any such method must include searching the web. After collecting a set of related documents, it is necessary to filter and cluster the data set. News portals contain mainly articles, annotations, discussions, blogs and symposia. Articles are important from the aspect of information tracing. Information that was found and sorted was classified according to the publication date.

We used the event of earthquake in Haiti that has been widely monitored by media, at the time as a developing story. In our experiment, three news portals represent the start web pages: www.pravda.sk, www.sme.sk and www.ta3.com. The key words we looked for were: earthquake and Haiti.

The process of following the story was divided into three parts. The first part is represented by an experiment that took place from 13 January 2010 10:00 am to 14 January 2010 4:00 pm.

We used parameters and the settings from the table 1.

Table 1: Parameters used in web story

Parameter	Value
Number of bees	30
Maximal dancing time	7
Maximal time in auditorium	4
Default energy	1
Energy decrement	0,05
Max. count quality	0,7
Max. header quality	0,15
Max. header number	3
Flesch's readability index	0,15

In the first part 9327 web pages were found, out of which 1066 were of non zero quality. Web pages of non zero quality where divided into 5 classes: informative pages, list of articles, discussions, blogs, graphic contents. Discussions and blogs are irrelevant because they represents only reactions to the event. The list of articles has no informative value, but is important for page discovery.

During the first part of experiment 217 informative pages with a quality higher than 60 percent were found. The most frequently used words on those 217 informative web pages were: disaster, tragedy, victims, OSN, chaos, help. 13 pages with quality higher than 90 percent were recommended to the user. These web pages were classified according to the published date extracted from the page.

The second part of the experiment took place 16 and 17 January 2010 always at the same time: from 8:00 am to 5:00 pm. 11439 web pages were found, 1193 with non zero quality. 298 informative web pages with the quality higher than 60 percent were found. The most frequently used words which occured on the web pages after 16 January 2010 8:00 am were: cadavers, indigence, looting, despair, water, help. Gradually, it was recommended to user 36 informative pages with quality higher than 90 percent during this experiment.

The last part of the experiment took place from 18 to 21 January 2010, always at the time from 6:00 pm to 11:00 pm. 12576 web pages were found, out of which 1271 of non zero quality. 327 informative web pages possesing a quality of above 60 percent were found, 66 web pages was recommended to user. The most frequently used words on the web pages after 18 January 2010 6:00 pm were: water, collections, help, charity, putrefaction, physicians.

When repeating the experiment twice or three times, the algorithm found almost all the web pages that were marked as relevant by the previous algorithm run, hence supporting a hypothesis that our method based on the modified bee hive model is quite robust.

From the experiments we can conclude the following:

- the method is able to look for web pages and evaluate the quality of the found web pages automatically,
- it can collect relevant pages,
- it can reconstruct the story backwards in time,
- it can monitor the story that is developed during the search,
- it provides statistical results about the searching process,
- by means of this method we can obtain the most frequently used words on time distinguishable web pages.

4.2 Function optimization

Another problem domain where we have applied the bee hive model was the optimization of mathematical functions. Sources in this case are different vectors of values of function arguments, the behaviour of the bees outside the hive has the following parameters:

- *Step size* Bees outside the hive can visit more than one source before returning to the hive. When the bee flies from source to source she adds a random number from the interval $(-stepSize, +stepSize)$ to some arguments of the function.
- *Energy* This parameter is inspired by the [15]. It is the energy of the bee acting outside the hive. When the bee runs off her spare energy, she has to return to the hive with her current source.

The pseudocode of the algorithm follows:

1. source := input source
2. step size := step size * quality of the source (optional)
3. generate 3 sources in an immediate vicinity of the source
4. source := using a roulette wheel algorithm choose one of the three generated sources
5. decrement energy
6. if it still has more energy left, continue with step 3 otherwise with step 7
7. return to the hive with the source

We tested this algorithm on a set of benchmark functions. These functions' parameters are shown in the table 2. These data were taken from the [20].

Table 2: Functions' parameters on which the experiments were performed

ID	$Functionname$	$Interval$	$Globaloptima$
1	Rosenbrock 2D	$[-1.2, 1.2]$	$X(1,1)F = 0$
2	Rosenbrock 2D	$[-10, 10]$	X(1,1) F=0
3	Goldstein & Price	$[-2, 2]$	X(0,-1) F=3
4	Martin & Gaddy	$[0, 10]$	X(5,5) F=0
5	Rosenbrock 4D	$[-1.2, 1.2]$	X(1,1,1,1) F=0
6	De Jong	$[-2.048, 2.048]$	X(1,1) F=3905.93
7	Branin	$[-5, 10]$	X(-22/7,12.275) X(22/7,2.275) X(66/7,2.475) F=0.3977272
8	Hyper Sphere	$[-5.12, 5, 15$	X(0,0,0,0,0,0) F=0

Table 3 shows the results of applying our proposed algorithm to the benchmark functions as well as results of other commonly used stochastic optimization algorithms which were previously published in [20]. The results are shown as an average number of evaluations of the benchmark function needed to achieve the required result. The required result is achieved when its value differs less than 0.001 from the global optimum.

The proposed algorithm was able to solve all the functions published in [20] except for the Griewank function.

Table 3: Experimental results as average from 100 iteration - S: Simplex method, NS: stochastic simulated annealing optimization procedure, GA: Genetic Algorithm, ANT: Ant Colony System, BA: Bees Algorithm, BH@W: our proposed bees algorithm, **** - data not available

FN	S	NS	GA	ANT	BA	BH@W
1	10780	4508	10212	6842	631	2409
2	12500	5007	****	7505	2306	16019
3	****	****	5662	5330	999	6773
4	****	****	2844	1688	526	645
5	21177	3053	****	8471	28529	68249
6	****	****	10160	6000	868	6699
7	****	****	7325	1936	1655	1822
8	****	****	15468	22050	7113	17152

From these experiments we can conclude that the proposed algorithm is able to optimize nontrivial mathematical functions in a reasonably good time (in terms of number of evaluation of the function) comparing to other commonly used algorithms.

On the other hand, there is a non multi-agent bee inspired algorithm which has much better results for each function tried in the experiments.

4.2.1 Parameters Used in the Experiments

The parameters used in the described experiments are stated in the Table 5 and the names are in the Table 4.

Table 4: Parameter names

Parameter	Symbol
Energy	E
Number of bees	NB
Maximal time in auditorium	MAT
Maximal dancing time	MDT
Observation error	OE
Step size	SS

Table 5: Parameters used in experiments

F	OE	NB	MAT	MDT	E	SS
1	2	15	3	2	0.2	0.09
2	2	15	4	3	0.2	1.0
3	3	15	3	2	0.2	0.13
4	2	15	4	3	0.2	1.0
5	2	15	4	3	0.2	0.08
6	2	15	4	3	0.2	0.08
7	2	15	3	2	0.2	0.8
8	1	15	4	4	0.2	0.85

The first four columns are the parameters of the hive while the second two are the parameters of the bee. All these parameters were set empirically.

We can see an interesting stability among the parameters of the hive. In every experiment the number of bees was 15 and the observation error was 0.2. Another interesting thing is the connection between the maximal dancing time and maximal time in auditorium. Algorithm achieved the best results when the maximal time in auditorium was 1 + maximal dancing time. Only for one function (function 8) these two values were equal.

The bee has two parameters: energy and step size. The energy was always equal to two except for two cases (functions 3 and 8). The most varying part of the algorithm is the step size of the bee.

4.3 Hierarchical Hive

Let us proceed to the idea of introducing hierarchy in the concept of the beehive. We shall also present some preliminary results of our research with the proposed algorithm aiming to create a hierarchy of bee hives.

4.3.1 What is a Hierarchical Algorithm

There are algorithms which utilize some sort of hierarchy. For example the Hierarchical Subpopulation Particle Swarm Optimization Algorithm [5] uses a hierarchy of particles to solve the problem with premature convergence of the algorithm. It isolates better solutions from those worse ones into different hierarchy levels.

In [11], they use a modified genetic algorithm where chromosomes are composed from other chromosomes. This can be viewed as a hierarchy structuring within data - data are composed from other data.

There is also a concept of hierarchy introduced in [4]. It tries to utilize emergency of solution of some problem, where the bottom level algorithms can solve the problem only from local perspective but the upper level algorithm uses the emerged solution to use other heuristics and to drive the bottom level algorithms. The author used a simple example of a hierarchical problem: there is a hexagon-shaped picture composed from other hexagons and the goal is to find a symmetrical shape. To accomplish this task the author used a simple Simulated Annealing without hierarchy. The result is shown in Figure 2.

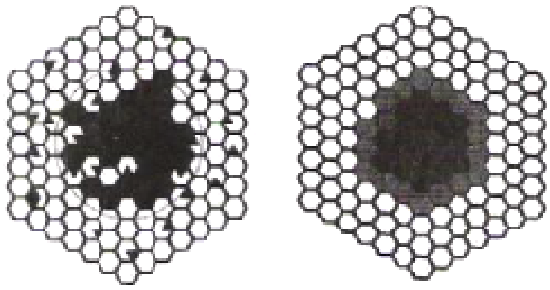

Figure 2: The hexagon to optimize (left side) and the result with simulated annealing (right side) [4]

The algorithm works as follows: in every hexagon there was one instance of the simulated annealing which sees only its hexagon and the hexagons in its vicinity. A heuristic was proposed to minimize the difference between its colour and the colour of the hexagons in its' vicinity.

If we look at Figure 2 we can see that the optimized shape is not fully symmetrical. The author proposed to use an algorithm which would be able to work on a different hierarchy levels with the emerged shape and to drive the algorithms on those shapes that are yet not symmetrical hexagons to redo their work. However, the author only outlined the algorithm in this conceptual way without proposing any specific implementation of it.

4.3.2 Our Concept of Hierarchy

Our concept of hierarchy is mostly based on the paper [4]. In section 4 we described two conditions the bee must satisfy to work with the hive.

The first condition is that the bee has to be able to take a source. If we consider a hive as a whole, it can have several sources in the dispatch room. These sources determine where the bees from the hive will fly. So the difference is that the hive can take more than one source, but still can take at least one source.

The second condition is that the bee has to be able to return a source. We can consider the most propagated source in the dance floor as the source chosen by the hive to be returned.

The fulfilling of the conditions implies that we can consider the hive as a more complex bee, but still the bee. So it should be possible to create a hive that would contain other hives that would contain bees.

The difference between our proposed hierarchy and the one described in [4] is that we have a structure that is genuinely recursive - the upper level hive is not kind of a different structure acting as a supervisor but it is something that is constructed from its bees (or hives). This gives it a capability to influence behaviour of the bottom level hives (without the need to know whether it is a bee or a hive). On the other hand, the concept of hierarchy in [4] is more similar to the human society - consider the supervisor and employee relationship.

4.3.3 Algorithm Description

In Figure 3 there is the topology of the hierarchy. Each hive from the low level (hives marked as Hive 1 through Hive M=128) is placed in one hexagon shown in the picture. Every low level hive contains N=30 bees.

The source is defined as the ID of the specific hexagon and its colour (i.e., various shades of grey). The bees in the low level hive are able to forage only for the source placed in its dispatch room (one hexagon). The foraging of the bees is defined as random changing the colour of the specific hexagon. The quality is defined as in [4].

The upper level applies a different heuristics. The quality of a source is given by two fractions and is calculated as their average. The former fraction is the quality estimated by the lower level hive. The latter fraction is a measure to which degree the specific source complies with the symmetry shaping of the whole picture. To accomplish this the root hive has to remember the best solution of each hexagon. The quality of the source is thus calculated as follows:

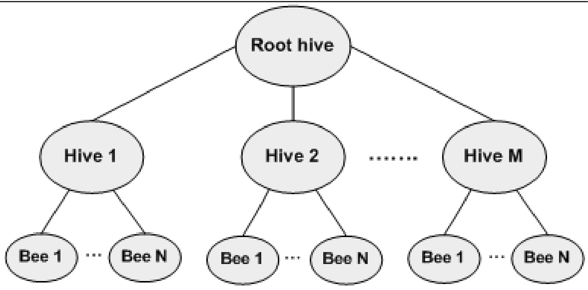

Figure 3: The topology of the hierarchy used in the experiments

1. Let X be the solution of optimizing the hexagon from the low level hive

2. Let Y be the best known solution of optimizing the same hexagon at the time

3. If the quality of X is better than for Y the X will be remembered and the calculated quality for X is returned for the source as its quality

4. If the quality of Y is better than for X the X will not be remembered and the calculated quality for Y is returned for the source as its quality

The resulting quality is then modified as follows: assume that this quality is X. Then the quality will be changed to $1 - X$ which means the better the source is, the smaller is the calculated quality. The reason for this is modification is to guide the hives to optimize unsolved hexagons.

4.3.4 Experimental Results

Using the algorithm described above we performed an experiment similar to one described in [4]. The combined results of this experiment are in Figure 4 as 9 runs of the experiment in a row.

As one can see in Figure 4 each run except of the first one produced symmetrical solutions.

The results are far from a proof of the capability of the hierarchical bee hive to have better performance as the not hierarchical algorithms. The objective of this experiment was to explore if this could be a promising area for future research.

5 Conclusions

In the paper, we presented a model of the bee hive based on the interactions of individual bees that allows to

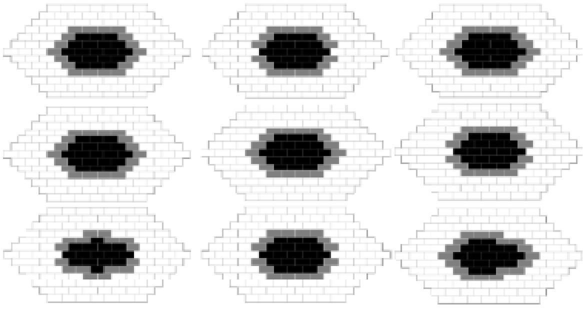

Figure 4: The results of the experiment with the hierarchical hexagon optimization

emerge the self-organizing decision-making behaviour of the hive.

The results of the function optimization are comparable with other commonly used algorithms, however, there is another bee algorithm that achieved better results on this area. The main advantage of the proposed algorithm is the clear separation of the decision-making mechanism inside the hive from the problem-specific behaviour of the bees outside the hive. This separation enabled us to apply the model to three different areas - web search, function optimization and even to a concept of hierarchical optimization.

With regard to web search, we explored chiefly the idea of employing bees in tracking a story that develops in time and is reflected in news reports, commentaries etc. that are emerging on web pages. The bees were able to keep on recommending the pages most recent articles relevant to the developing story.

The source codes of the model are freely available as a framework at [1].

Acknowledgement

This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10 and by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11 (2011-2014)

References

[1] BeeHive@Work source code avaible at: http://beehiveatwork.sourceforge.net.

[2] Baykasoglu, A. et al. Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem. In *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, pages 111–143, Austria, 2007.

[3] Beekman, M. et al. What Makes a Honeybee Scout? In *Behavioral Ecology and Sociobiology*, pages 985–995, Austria, 2007.

[4] Cerny, V. *Searching for the Best Model of the World: from Simulated Annealing to Hierarchical Systems (in Slovak).* IRIS, Bratislava, 2000.

[5] Chuan, L. and Quanyuan, F. A Hierarchical Subpopulation Particle Swarm Optimization Algorithm. In *Advances in Intelligent Systems Research*, 2007.

[6] de Vries, J., H. Biesmeijer. Modelling Collective Foraging by Means of Individual Behaviour Rules in Honey-Bees. In *Behav Ecol Sociobiol*, pages 109–124, Berlin, 1998.

[7] Dervis, K. An Idea Based on Honey Bee Swarm For Numerical Optimization, Technical Report-TR06,Erciyes University, Engineering Faculty, Computer, 2005.

[8] Gheorghe, M. et al. What Makes a Honeybee Scout? In *Computational Models of Collective Foraging. BioSystems 61*, pages 133–141, Austria, 2001.

[9] Gordon, D. The Organization of Work in Social Insect Colonies. *Nature*, 1996.

[10] Habiba, D. et al. Cooperative bees swarm for solving the maximum weighted satisfiability problem. In *IWANN*, pages 318–325, Poznan, 2005.

[11] Haubelt, C. et al. Solving hierarchical optimization problems using moeas. In *Evolutionary Multi-Criterion Optimization*, volume 2632 of *LNCS*, pages 162–176, Berlin, 2003. Springer.

[12] Karaboga, D. and Akay, B. A survey: algorithms simulating bee swarm intelligence. *Artif. Intell. Rev.*, 31:61–85, June 2009.

[13] Lemmens, N. et al. Bee Behaviour in Multi-agent Systems (A Bee Foraging Algorithm). In *Adaptive Agents and MAS III*, pages 145–156, Berlin, 2008.

[14] Lorenzi, D. S., F.; Santos and Bazzan, A. L. C. Negotiation for task allocation among agents in case-base recommender systems: a swarm-intelligence approach. In *Proceedings of the Workshop Multi-Agent Information Retrieval and Recommender Systems - Nineteenth International Conference on Artificial Intelligence (IJCAI 2005)*, pages 23–27, 2005.

[15] Menczer, F. and Monge, A. Scalable web search by adaptive online agents: An infospiders case study. In *Intelligent information agents: Agent-based Information discovery and management on the internet*, pages 323–340. Springer Verlag, 1999.

[16] Navrat, P. Bee hive metaphor for web search. *Communication and Cognition-Artificial Intelligence*, 23:15–20, 2006.

[17] Navrat, P. and Bou Ezzeddine, A. Bee hive at work: Following a developing story on the web. In *IFIP Advances in Information and Communication Technology*, volume 331, pages 187–196. Springer Verlag, 2010.

[18] Navrat, P. et al. Web Search Engine Working as a Bee Hive. *Web Intelligence and Agent Systems*, 6:441 – 452, 2008.

[19] Navrat, P., Jastrzembska, L., and Jelinek, T. Bee hive at work: Story tracking case study. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '09, pages 117–120, Washington, DC, USA, 2009. IEEE Computer Society.

[20] Pham, D. et al. The Bees Algorithm â A Novel Tool for Complex Optimisation Problems. IProms, 2006.

[21] Rambharose, T. and Nikov, A. Computational intelligence-based personalization of interactive web systems. *WSEAS Trans. Info. Sci. and App.*, 7:484–497, April 2010.

[22] Schultze, S. J. A collaborative foraging approach to web browsing enrichment. In *CHI '02 extended abstracts on Human factors in computing systems*, CHI '02, pages 860–861, New York, NY, USA, 2002. ACM.

[23] Seeley, T., Camazine, S., and Sneyd, J. Making in Honey Bees: How Colonies Choose Among Nectar Sources. In *Behav Ecol Sociobiol*, pages 277–290, PoznaÅ, 1991.

[24] Teodorovic, D. Bee colony optimisation (bco). In *Studies in Computational Intelligence*, volume 248, pages 39–60. Springer Verlag, 2009.

[25] Teodorovic, D. and Dell'orco, M. Bee colony optimization - A cooperative learning approach to complex transportation problems. In *Advanced OR and AI Methods in Transportation*, pages 51–60, Poznan, 2005.

[26] Wong, L.-P. et al. Bee Colony Optimization algorithm with Big Valley Landscape Exploitation for Job Shop Scheduling problems. In *Winter Simulation Conference*, pages 2050–2058, 2008.