

What Can Honeybees Tell Us about Social Learning?

Robin Clark
rclark@sas.upenn.edu

Steven O. Kimbrough
kimbrough@wharton.upenn.edu

March 22, 2019

Contents

1	Introduction	2
2	Methods	4
3	Results	8
3.1	Default Parameter Settings	8
3.2	Default Parameter Settings: Random Site Values	11
3.3	Varying the Quorum Requirement	15
3.4	Varying the Number of Scouts	18
3.5	Dance Decay Rate	19
3.6	Probability of becoming an observer, p_m (pm)	20
3.7	Search Area Size	22
4	Discussion and Conclusion	23
4.1	Summary	23
4.2	Social Learning	23
4.3	Metaheuristics	24
4.4	Other Future Research	24

Abstract

We independently implemented and studied the nest-site selection model described in Passino, K. M., and Seeley, T. D., “Modeling and analysis of nest-site selection by honeybee swarms,” 2006, focusing on the default parameter values they obtained by field calibration. We focus on aspects of the model pertaining both to imitation and social learning and to the model as kind of metaheuristic. Among other things, we find that the model is robust to different parameterizations of social learning, but that at least a modicum of social learning is essential for successful nest-site selection (in the model). Regarding the model as a metaheuristic, we find that it robustly produces good but significantly non-optimal nest-site selections. Instead of a single-criterion

metaheuristic, the algorithm is best seen as balancing three objectives: choose the best of the available sites in the neighborhood, make the choice quickly, minimize risk of failing to choose a site.

Keywords: social learning, imitation, metaheuristics, honeybees

1 Introduction

Honeybees are a remarkable collection of species (any member of the genus *Apis*). With their earliest fossils appearing 34 million years ago, they are on the order of 30 times older than our own species, and have survived, even flourished, in the face of drastic changes in the biosphere and geosphere, changes for which human flourishing and civilization would be problematic.

Among their many striking traits is the process by which a successful colony, having outgrown its nest site, sends out colonists and acquires a new nest site for a portion of the original colony, which continues to operate. This process has only reached mature scientific understanding in the last decade or so, beginning from initial studies in the 1950s. The science is thoroughly and delightfully described in Seeley’s book *Honeybee Democracy* [14].

Briefly and at a high level of description, nest site-selection works as follows. Bees in a successful hive sense the need or opportunity for form a new colony. They begin to nurture a new queen, who is created from an ordinary larva by special feeding. As the queen matures, a large number of bees (on the order of 10,000) in the hive go quiet and gorge themselves on honey, which they otherwise cannot take with them. Rather suddenly, the new colonists, including the old queen (once a new queen is hatched to stay with the old hive) leave the hive and form a large swarm near by. The swarm quickly identifies a staging place, typically on a nearby tree branch, and congeals in a small living ball of densely packed bees, who adjust their metabolisms so as to conserve energy and to maintain the ball at the proper temperature. A few score or few hundred scout bees are recruited from bees that had been foragers. These scout bees fly off from the swarm more or less in random directions, in search of candidate nest sites. Their searches may extend several kilometers from the swarm. When the scouts return, those that have found promising sites engage in the famous waggle dance, indicating both direction to the candidate nest site and the bee’s own assessment of its quality. Scouts that do not encounter satisfactory sites rest and observe the dances. After a time, they go exploring again, possibly heading to one of the danced-for sites (with the choice of site biased towards more highly evaluated sites that are being danced for). This is a form of imitation grounded not in directly copying an observed behavior, but in interpreting a signal transmitted by a compatriot, which allows the bee to imitate the behavior at a distance. If the bee does not imitate a dancer in this way, the bee explores randomly again, a form of innovation rather than imitation. The process continues until the bees sense a “quorum”, a sufficient number of bees at one of the candidate sites, individually exploring its characteristics. Once this happens, scouts

returning from the site begin “piping”, making distinctive sounds that cause the bees in the swarm to awaken and prepare for flight. Soon after, the compressed swarm explodes into flight, dispersed over several score meters. Directed by scouts, the swarm heads off towards the chosen nest site. When it reaches the new site, if all has gone well, the bees enter it in an orderly fashion, including the queen, and begin operating as a new hive, striving to build up honey reserves as quickly as possible.

Of course, much can go wrong and site selection ventures may often end in failure and the destruction of the would-be colonists. What is amazing is that over evolutionary time, millions of years, this mechanism has succeeded in preserving the species.

Honeybee nest-site selection was investigated and its mechanics discovered by biologists interested in ascertaining the basic biology involved. Investigations continue, but it seems clear that the basic biology of honeybee nest-site selection is understood and the findings are settled. Our interest in honeybee nest-site selection complements and extends—builds upon—the basic biology. This interest of ours is mainly focused on two rather distinct topics, which we shall now discuss.

Social learning is the first and indeed primary impetus driving our interest in honeybee nest-site selection. By social learning we mean, roughly, learning by individuals in a population (or society) driven principally by observation of, and interaction with, other individuals. This is to be contrasted with learning in which individuals mainly observe and interact with nature, that is, with the world outside the society. There is now a burgeoning field of study of social learning, motivated in large part by an overarching interest in social and cultural evolution (see, e.g., [4, 7, 8] for recent overviews). Following the important work of Boyd and Richerson (e.g., [2, 13]), who developed theory, complemented by empirical studies, that emphasizes the importance of imitation (and social learning more generally) in cultural evolution, a large amount of research in social and cultural evolution has been, and continues to be, directed at understanding social learning. (See [11] for an important example of a study of this kind.)

The second impetus for our interest arises from the fact that the metaheuristic (\approx heuristic optimization algorithm schema) instantiated by the honeybees in selecting new nest sites is itself interesting and worth exploring for other applications. Metaheuristics (the term was coined by Fred Glover) are standardly categorized into two kinds: local search heuristics (such as simulated annealing, GRASP (greedy randomized adaptive search procedure), mimetic algorithms, and tabu search) and population-base heuristics (evolutionary algorithms such as genetic algorithms and genetic programming, particle swarm optimization, and ant colony optimization) [1, 6, 9, 12, 16, 17]. There are literally scores of published metaheuristics, many of them “biologically inspired.” Particularly notable about the honeybee nest site algorithm (schema), is that it does not fit well in either the local search category or the population based category of metaheuristics, although of course it has flavors of both approaches. Instead, it might best be described as an individual-based learning algorithm [5], akin to reinforcement learning in some guise, that is distributed rather than centrally guided. The individual in question, of course, is the swarm of bees,

not the individual bees themselves, but the learning mechanism is distributed across all of the scouts and is not controlled by an executive. As such, it perhaps is most similar to ant colony optimization and particle swarm optimization in the canon of metaheuristics. Even so, in its use of copying and imitation, it also resembles mimetic algorithms. In any case, there is much to hold one’s interest.

This is an exploratory study. We reimplemented in Python 3.6.4 the honeybee nest-site selection simulation model presented and discussed by Passino and Seeley in [10]. Our implementation was from scratch, because [10] does not provide the source code or very much in terms of software documentation. Although their paper is very informative and reports much of interest and value, even with a very close reading we were unable to discern certain aspects of their model. Fortunately, our results cohere with those reported in [10]. Moreover, our implementation serves well for exploring the two main topics of interest to us: social learning and the use of the honeybee nest-site selection procedure as a metaheuristic.

We describe our implementation in §2 and our results in §3. §4 concludes the paper with a discussion of the results and promising avenues for future research.

We note that recent releases of NetLogo have come with a model of honeybee nest site selection, called the *BeeSmart Hive Finding* model. This is an interesting and useful model with nice graphics. No doubt it could be modified for the experimental purposes we have in mind, but that remains to be done. Our Python implementation is quite independent and is well suited for our purposes. Like the NetLogo model, we are happy to make the source code and documentation publicly available.

2 Methods

Table 1 presents the model’s parameters and default values, drawn from Passino and Seeley in [10]. The table is for purposes of reference. We refer to these variables in the sequel as needed. Table 2 presents, also for purposes of future reference, observed or computed quantities in the model.

Our documented source code is available from the authors. The original article by Passino and Seeley [10] supplies, in addition, valuable documentation and background for the model, including parameter calibration information. In consequence, and because of space limitations, we focus here on the core procedures of the model.

All of the modeled bees are scouts. In both Passino and Seeley [10] and in our implementation, the model assumes that a swarm has been formed and focuses on the actions of the scout bees as they explore for new sites and either come to agreement or not (within the allotted time of 64 ticks or 32 hours) regarding a new site.

In our implementation, each bee is an instance of the class `Scout` and a list of all modeled bees/scouts (alive or dead) is maintained during the run of the model. For present purposes, we call this list **Bees**. Each run of the model is effected by a call to the `main()` procedure,

Symbol	Default value	Description
Jxsize, Jysize	21, 21	side dimensions of the nest-site quality landscape, J
locations	[(8, 11), (14, 6), (2, 20), (16, 0), (15, 19), (16, 18)]	Grid coordinates of candidate sites
ε_q (epsilon _q)	20	quorum threshold
ε_s (epsilon _s)	15	dance decay rate
ε_t (epsilon _t)	0.2	quality/dance threshold
σ	400	exploration tendency
siteValues	[0.1,0.3,0.35,0.5,0.55,1]	site utility scores
B	100	Number of scout bees
maxTicks	64	maximum number of ticks in a run
mRate	0.1	nominal mortality rate
p_m (pm)	0.25	probability of becoming an observer
σ (sigma)	400	tendency to dance parameter

Table 1: Default parameter settings. Program variables given in `typewriter` font.

which in turn consists largely of a `for` loop over the range of ticks, $[0, 1, \dots, 63]$, 64 in all by default, `range(parameterSettings.maxTicks)` in Python. We focus now on the operations occurring in the governing `for` loop during each tick (modeled 30 minute time period).

1. `cullBeesAtRisk()`. Scouts foraging from the swarm are at risk of death, determined by parameter values as shown in Table 2. In this step each scout that is either in the `'Committed'` or the `'Exploring'` state is converted to the `'Cadaver'` state with probability p_d .

This step implements the assumption that committed/dancing bees visit the site they are committed to each tick, thereby making themselves available to be counted as part of a potential quorum for the site in question.

2. `goExplore`. Each bee in the the `'Exploring'` state, either it has acquired a site during the previous tick (by imitation and so has a `spot` value) or not. For those that have not, randomly assign them each a point of the search grid, J , which is a `Jxsize×Jysize` grid. (See Table 1.) For each (surviving) bee in the `'Exploring'` state, record in the `visitsCounter` at this tick if it visits one of the candidate sites and determine whether it has found an attractive site and becomes `'Committed'` with an initial dance strength to a candidate site or, if not, goes to the `'Resting'` state. (Thus, at the exiting of this procedure, all `'Exploring'` bees have either become `'Committed'` and are dancing or are in the `'Resting'` state.)

Symbol	Description
p_d	<code>mRate/maxTicks = mortalityRate=</code> probability of death on a single expedition
k_j^i	step (tick) at which bee i discovered site j
$L^{ij}(k)$	dance strength of bee i at step k for site j
$L^i(k)$	dance strength of bee i at step (tick) k
$L_t(k)$	total dance strengths at step k
$p_e(k)$	probability an observer becomes an explorer

Table 2: Important observed or computed values. Program variables given in `typewriter font`.

3. `decayDance()`. For each bee in the 'Committed' state, decay its dance strength and compute its new value of $L^{ij}(k)$. (See Table 2 and [10, page 430, right].) For any bee whose $L^{ij}(k)$ value falls below ε_t , set its dance strength to 0 and its state to 'Resting'.

At this point, all bees that are alive are either 'Committed' or 'Resting'.

4. `sleepersAwake()`. For each bee in the 'Resting' state, convert it to the 'Observing' state with probability p_m . (See Table 1.)
5. `exploreOrCommit()`. For each bee in the 'Observing' state, calculate $p_e(k)$. (See Table 2 and [10, page 432, expression (1)].) With probability $p_e(k)$, convert the associated bee to the 'Exploring' state. If there is insufficient dancing ("low enthusiasm") then $p_e(k) = 1$ or is high and every (or nearly every) bee is converted from 'Observing' to 'Exploring'.
6. `pickSiteCommit()`. For each bee now (still) in the 'Observing' state, probabilistically pick one of the sites that is being danced for and target it, that is, acquire the site as the target of its visit during the next tick. (Set its `spot` value to that of the bee it is imitating.) (See [10, page 433, expression (2)].) Convert the bee to the state 'Exploring'.

At this point every bee is either dead, exploring, committed, or resting.

7. `quorumCounts()`. Check to see whether there is a quorum evident in `totalVisits` at this tick. If so, break out of the loop, record statistics, and terminate the run.

* * *

Note that [10, page 432] set σ to 4000. This gave us implausible values when we implemented their formulas [10, page 432, expression (1)], which were inconsistent with their description. We believe it is a typo of sorts and found that 400 gave us plausible

values, so $\sigma=400$ is the default value for that parameter. All other parameters in our model are, so far as we know, identical with those in [10].

3 Results

Seeley and Passino comment as follows about the foci of their interests with regard to the honeybee site-selection model.

It seemed a priori that all three parameters—quorum threshold $[\varepsilon_q]$, [dance enthusiasm] decay rate $[\varepsilon_s]$, and exploration tendency $[\sigma]$ —could strongly affect the outcome and the timing of a swarm’s decision making. [10, page 428]

We examine all these factors in the present section. In addition, given our interests in social learning and imitation, we explore additional parameters as well, especially p_m (**pm**), the probability of becoming an observer.

3.1 Default Parameter Settings

Implementing our best interpretation of [10] (and see Table 1), we obtain, in trials of 100 runs, results that are in broad agreement with those reported by Passino and Seeley. The comparison cannot be terribly precise because they often report only results from single, “presentative” runs. In this reference trial, see Figure 1, the hives settled on the best candidate site 41 of 100 times, and never failed to find a quorum . The second-best site, with a value of 0.55, was picked 20 times. The average number of ticks to find best site, when the best site was chosen, was 7.49.

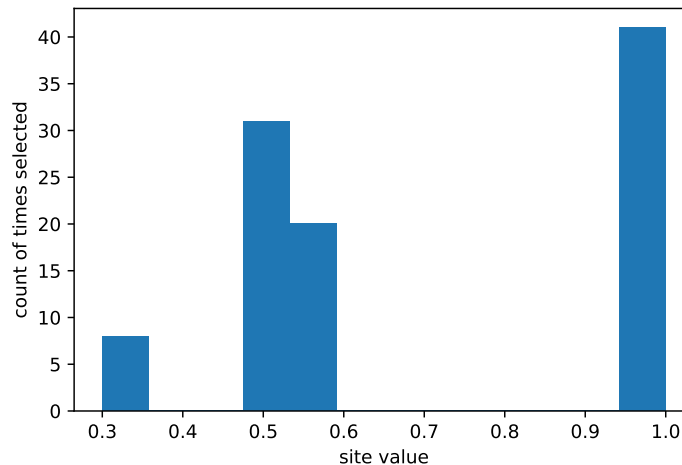


Figure 1: Histogram of the chosen sites from 100 runs in the default configuration. Sites chosen: 1, 41 times, 0.55, 20 times, 0.5, 31 times, 0.35, 6 times, 0.3, 2 times, and None, never.

Figure 2 plots the counts of visits to the various sites over the span of the 100 runs. The six sites are numbered from 0 to 5 in increasing order of site value. Thus, site 5 is the best and has a value of 1 in the default case; see Table 1. While noisy, Figure 2 evidences a clear pattern: better sites tend to receive more visits over 100 runs.

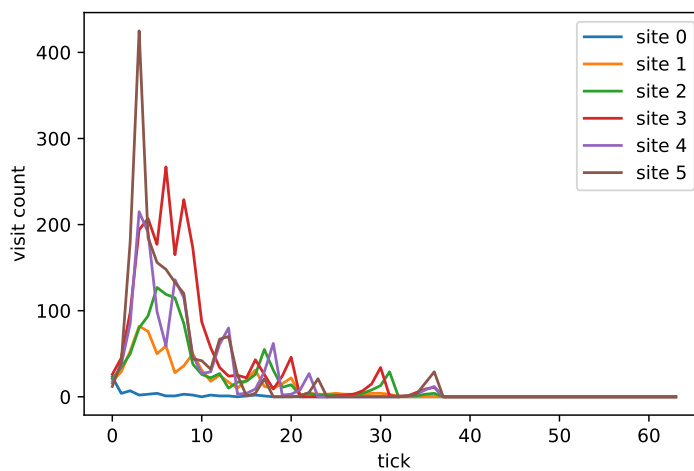


Figure 2: Visits records from 100 runs in the default configuration.

Table 3 gives a statistical summary of 30 replications of trials with 100 runs each under the default parameter assumptions. We see that on average a trial (of 100 runs) found the best site about 45 times and the second best site about 26 times. None of the 3000 runs resulted in failure to pick some site (there were no Nones; column D).

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	45.467	6.872	26.433	0.000	8.015
std	4.790	1.114	3.803	0.000	0.755
min	36.000	5.220	18.000	0.000	6.580
25%	42.000	6.011	24.000	0.000	7.490
50%	46.000	6.777	27.000	0.000	7.790
75%	49.000	7.294	28.750	0.000	8.310
max	54.000	10.023	34.000	0.000	9.970

Table 3: Statistical summary of 30 replications of 100 runs of the default settings. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

How impressive is this performance? From one perspective, more than half the time the swarm failed to settle on the best candidate nest site. From a second perspective, what matters is long-term survival and flourishing. If the candidate sites that are actually considered by the swarm are in fact close in expected value with respect to natural selection, then what matters most is that *some* adequate site is selected. The **None** result is an evolutionary dead end from the perspective of the swarm (and the sourcing hive). So this is a complex question, one that is beyond the scope of this focused study.

From a third perspective, we can compare the performance of the hive against chance. This, too, is a subtle issue, but given our primary interest in social learning, we can ask what happens if there is no imitation or social learning at all. To model this, we make two parametric modifications. First, we set $\varepsilon_s = 0$. This eliminates decay in dance strength by a scout, so that once a scout has found a danceable site, it keeps dancing for and visiting the site. Second, we set $\sigma = 1000000$. This has the effect of virtually eliminating any chance that an observing scout will imitate a dancing scout and adopt its candidate site. Thus, when an uncommitted scout returns to the swarm, either it has found an acceptable candidate or not. In the former case, it dances and maintains its commitment to that site for the duration. In the latter case of an uncommitted scout not finding a suitable site on its foray, the uncommitted scout remains uncommitted and forays again after a rest.

Figure 3 plots the visits record from a trial of 100 runs with this setup. It should be compared with Figure 2. In Figure 3, the values for each site increase at a more or less constant rate with the tick count. Site 0 is a comparatively poor site, so that a scout visiting it might actually return and not become committed. Site 1, similarly, but less so. The remaining four sites are sufficiently high in quality that every scout visiting them becomes committed, although each scout assesses quality with noise. But in the end, no site is every selected under this setup. Without imitation, the swarm fails to settle upon a

new nest site, even though its scouts are finding all of the candidates.

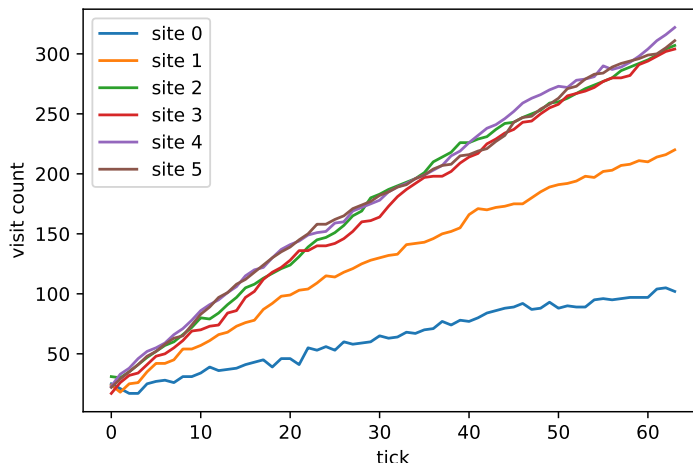


Figure 3: Visits records from a trial of 100 runs in the default configuration, but with no imitation ($\varepsilon_s = 0, \sigma = 1000000$).

3.2 Default Parameter Settings: Random Site Values

The default site values, $[0.1, 0.3, 0.35, 0.5, 0.55, 1]$, are given in Table 1. They perhaps make it easy on the hive to find the best site, since its value, 1, is so much larger than that of the second best site, 0.55. Also, the worst site, at 0.1, is unlikely to meet the quality/dance threshold of 0.2, `epsilon` in Table 1. This serves to quickly eliminate it in most cases from consideration by the hive.

In search of a more realism, we repeated the experiment by doing 100 runs with the default parameter settings, except that the site values were determined by drawing random values uniformly from the $[0, 1]$ interval.

In this experiment, the hive chose the best site available to it in 44 of the 100 cases, the second best in 22 instances, and no site in only 1 instance. Thus, the overall performance matches that of the default case, which is seemingly easier for the hive.

Figure 4 is the analog of Figure 1, but has a rather different interpretation. In the runs underlying Figure 4, the six site values were constant: $[0.1, 0.3, 0.35, 0.5, 0.55, 1]$, given in Table 1 and used by Passino and Seeley. Thus, the results could in principle be binned into 6 categories. The runs underlying Figure 4 each had randomly-drawn site values. These were, however, ranked in increasing order: site 5 being the best, site 0 the worst. Thus, in principle there could be 6×100 bins of distinct site values. Figure 4 imposes 12 bins on

these data. The figure indicates that generally, the hives were able to select higher-valued candidate sites.

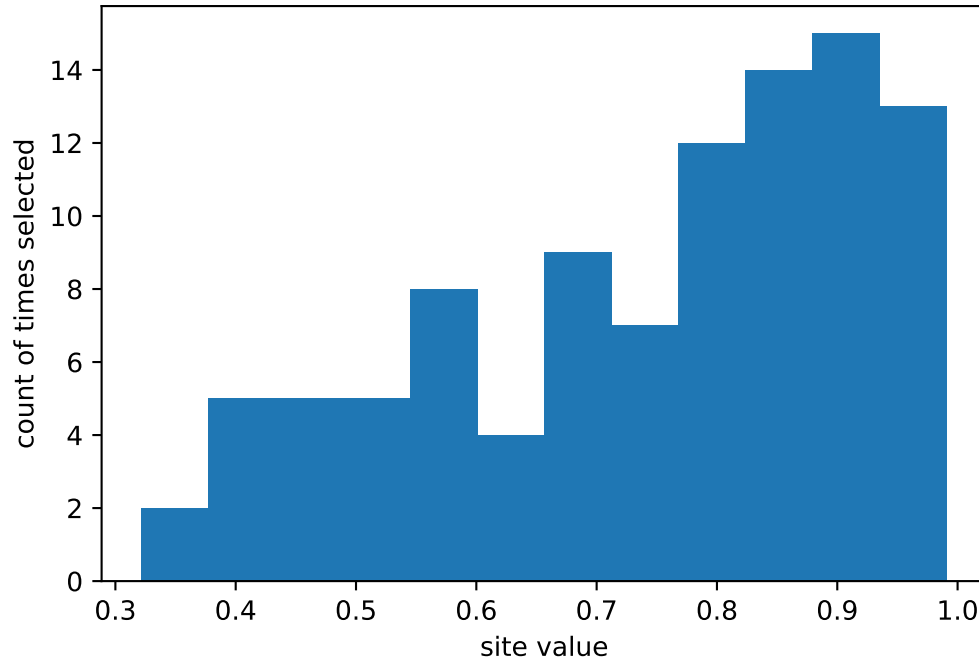


Figure 4: Histogram of the chosen sites from a trial of 100 runs in the default configuration, but with random sites values drawn anew for each run. Sites chosen: Best, 44 times, second best, 22 times, and None, 1 time. Average number of ticks to find best site (when the best site was chosen): 8.64. Average number of ticks overall: 9.32.

Comparing Figure 5 with Figure 2, we see a similar but more noisy pattern, in which the best and second best sites are distinctly more visited than the others.

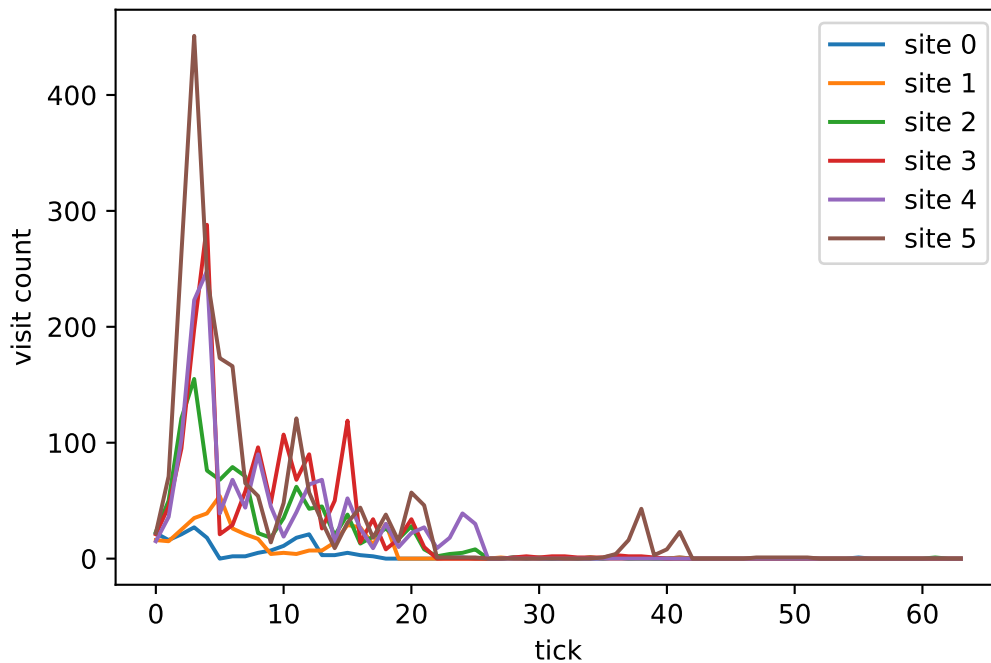


Figure 5: Visits records from 100 runs in the default configuration, but with randomly-drawn site values for each run.

Table 4 corresponds to Table 3, but with random site values being chosen for each constituent run. There is perhaps some slight degradation in the performance, but if so, it is small. Here, the swarm found the best available site (regardless of its absolute score) nearly 39% of the time on average, the second best 29% of the time and failed to settle only 0.6% of the time.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	38.633	7.664	29.167	0.600	7.770
std	4.359	1.375	4.496	0.894	0.925
min	31.000	5.590	19.000	0.000	5.970
25%	36.000	6.521	27.000	0.000	7.168
50%	38.500	7.381	30.000	0.000	7.715
75%	41.500	8.629	31.750	1.000	8.190
max	48.000	11.091	39.000	3.000	10.330

Table 4: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

3.3 Varying the Quorum Requirement

The default quorum value, ε_q , is 20, which was determined by Seeley and collaborators through observation. Table 5 summarizes the results from a 100-run trial with $\varepsilon_q=15$. Comparing these results to those in Table 4, there is little or no noticeable difference.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	38.367	7.671	28.867	0.233	7.423
std	4.131	1.358	3.937	0.430	0.776
min	29.000	5.244	23.000	0.000	6.030
25%	35.250	7.007	25.250	0.000	7.110
50%	39.000	7.488	29.000	0.000	7.310
75%	41.000	8.079	32.000	0.000	7.730
max	45.000	11.619	37.000	1.000	9.700

Table 5: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values and $\varepsilon_q = 15$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

We now consider the effects of increasing the quorum size required to 50. Table 6 summarizes the results of 30 repetitions of 100-run trials. Surprisingly, the results are not much different from those in Table 5.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	42.600	10.753	30.200	0.833	11.143
std	4.553	1.255	3.755	0.950	1.081
min	32.000	8.316	23.000	0.000	9.470
25%	39.000	9.958	27.000	0.000	10.360
50%	43.500	10.656	30.000	1.000	11.095
75%	45.750	11.633	33.000	1.000	11.780
max	53.000	13.841	36.000	4.000	14.200

Table 6: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values and $\varepsilon_q = 50$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

Table 7 summarizes results for when $\varepsilon_q = 70$. Surprisingly, at least to us, there is seemingly little degradation in the performance of the swarm. The chance of finding either the best or second best sites remains at about 72%, while the probability on average of not settling on a site rises to 5.1%. What does change significantly is the average length of a run (column E), which goes from 7.4 ticks when $\varepsilon_q = 15$ to 17.8 when $\varepsilon_q = 70$. We suspect that the 5.1% failure rate is comparatively highly disadvantageous, and that, given the exposure and energy use of the swarm, the lengthening of the average number of ticks to 17.8 is also a large loss over evolutionary time.

Finally, Table 8 summarizes results for when $\varepsilon_q = 85$. Unsurprisingly, the numbers would seem to be catastrophic over evolutionary time, with a failure rate in excess of 56%. When $\varepsilon_q = 90$, the results are much worse than this. These findings are perhaps not relevant biologically, but they do serve to help validate the model and our implementation of it.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	44.533	15.647	28.400	5.100	17.832
std	5.412	1.519	4.889	1.989	1.405
min	35.000	12.119	18.000	0.000	14.860
25%	40.250	14.757	26.250	4.000	16.742
50%	45.000	15.724	28.000	5.000	17.870
75%	48.750	16.185	30.750	6.750	19.015
max	54.000	19.154	42.000	9.000	20.230

Table 7: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values and $\varepsilon_q = 70$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	25.267	15.768	12.533	56.633	42.031
std	4.051	1.852	3.928	4.460	2.311
min	17.000	12.800	7.000	47.000	37.660
25%	23.000	14.558	9.000	53.250	40.808
50%	25.000	15.456	12.000	56.500	41.785
75%	27.750	16.730	16.000	59.750	43.692
max	34.000	19.381	19.000	66.000	47.500

Table 8: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values and $\varepsilon_q = 85$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

3.4 Varying the Number of Scouts

The default value for the number of scouts, B , is 100. It was determined by Seeley and collaborators through observation. With $B=75$, instead of 100, we see in Table 9 compared to Table 4 on page 14.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	40.167	11.652	29.167	1.367	11.803
std	3.779	1.427	4.457	1.066	1.025
min	34.000	8.867	21.000	0.000	9.750
25%	36.250	10.395	27.000	1.000	11.168
50%	41.000	11.577	30.000	1.000	11.820
75%	43.000	12.638	31.750	2.000	12.518
max	46.000	14.432	40.000	4.000	13.790

Table 9: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values and $B = 75$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

Table 10 summarizes results when the number of scout bees is set to 150, compared to the default value of 100. We see no discernible improvement in success in finding nest sites, but strong improvements in the speed of doing so.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	38.933	4.525	28.433	0.100	4.568
std	5.723	0.704	4.446	0.403	0.355
min	29.000	3.300	19.000	0.000	3.970
25%	35.250	4.006	25.000	0.000	4.338
50%	39.000	4.479	28.000	0.000	4.510
75%	42.000	5.055	31.750	0.000	4.908
max	55.000	6.237	38.000	2.000	5.220

Table 10: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values and $B = 150$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

3.5 Dance Decay Rate

The dance decay rate parameter, ε_s (epsilons) has a default value of 15, from Seeley’s calibrations. Setting it to 5 yields results quite similar to the default results shown in Table 4 on page 14. At 50, it yields data summarized in Table 11. Success rates (chances of finding the best or second best site) are slightly higher than in the default configuration, about 75% compared to about 68%, but with an increase in **Nones** (failures, column D) and a nearly 50% increase in the time (ticks) taken to reach a resolution.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	46.467	11.677	29.833	2.133	11.684
std	4.637	1.936	3.833	1.306	1.254
min	37.000	7.490	22.000	0.000	8.940
25%	42.500	10.668	27.000	1.000	10.832
50%	46.500	11.426	29.000	2.000	11.775
75%	49.000	13.078	32.000	3.000	12.610
max	55.000	16.370	37.000	6.000	14.760

Table 11: Statistical summary of 30 replications of 100 runs of the default settings, but with random nest site values and $\varepsilon_s = 50$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

3.6 Probability of becoming an observer, p_m (pm)

The default value of p_m , the probability in a given tick that a resting bee awakens and begins observing the dances, is 0.25. Table 12 summarizes results with $p_m = 0.15$. The results quite similar to the default results shown in Table 4 on page 14. The main difference here is in the time needed to resolve the search, columns B and E, which is increased. More rest for the bees leads to longer times to select nest sites, but no evident change in quality of choice.

When p_m increases to 0.5, Table 13, there may be a slight decrease in success rate, but there is plausibly an increase in the time taken for resolution.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	43.467	12.896	28.700	2.500	13.522
std	5.296	1.637	4.419	1.570	1.578
min	32.000	10.250	19.000	0.000	11.130
25%	41.000	11.962	25.250	1.250	12.275
50%	43.500	12.853	28.500	2.000	13.170
75%	46.750	13.538	30.750	3.000	14.853
max	58.000	18.204	37.000	7.000	17.490

Table 12: Statistical summary of 30 replications of trials of 100 runs of the default settings, but with random nest site values and $p_m = 0.15$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	35.533	4.240	28.333	0.100	4.298
std	4.183	0.613	4.097	0.305	0.368
min	28.000	2.853	20.000	0.000	3.670
25%	33.250	3.897	26.000	0.000	4.090
50%	35.500	4.178	28.000	0.000	4.270
75%	38.750	4.524	31.000	0.000	4.408
max	43.000	5.794	37.000	1.000	5.270

Table 13: Statistical summary of 30 replications of trials of 100 runs of the default settings, but with random nest site values and $p_m = 0.5$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

3.7 Search Area Size

The bees search randomly (in the model) in a $Jxsize \times Jxsize$ grid, with $Jxsize = Jysize = 21$ by default. Table 14 presents summary results for performance on a 41×41 grid. Performance degrades substantially compared to the default 21×21 configuration. What is perhaps surprising is that the success rate (rate of selecting the best or second best candidate) is only modestly diminished. Much strong effects are seen in the increase in failures, **Nones** in column D, and the time needed to reach resolution (columns B and E).

	A	B	C	D	E
count	30.000	30.000	30.000	30.000	30.000
mean	32.833	19.963	28.033	9.833	23.455
std	4.907	2.682	6.408	2.755	1.787
min	22.000	14.816	15.000	4.000	18.840
25%	29.250	18.019	24.000	7.250	22.690
50%	32.500	19.396	28.000	10.000	23.650
75%	36.750	21.758	31.000	12.000	24.475
max	41.000	27.636	45.000	14.000	26.400

Table 14: Statistical summary of 30 replications of trials of 100 runs of the default settings, but with random nest site values and $Jxsize = Jysize = 41$. A = best candidate site chosen in 100 runs. B = ticks to choose the best site, when it is picked. C = second best candidate chosen in 100 runs. D = no site chosen in 100 runs. E = ticks per trial of 100 runs.

4 Discussion and Conclusion

4.1 Summary

We independently implemented and studied the nest-site selection model described in [10], focusing on the default parameter values they obtained by field calibration.

We have systematically explored the behavior of the nest-site selection model in response to individual changes to all the important parameters of the model about their default values, Table 1. In doing so, we have obtained some findings that, while not inconsistent with [10], extend what they report. We also have eschewed for most of our analyses the fixed, probably atypical, distribution of site values used by [10]. Instead, we generate random values with each run.

We find the model to be highly robust in the neighborhood of the default parameter values, although we have not reported interaction effects (except for **Jxsize** and **Jysize** together). **Jxsize** and **Jysize** are again an exception. Quadrupling the size of the search area from 21×21 to 41×41 noticeably degrades the nest-site selection process. Across changes to each of the other parameters, the main effects of changes concentrate less on success—defined as selecting the best or second best nest site—and much more on risk of failure to select some site and on time taken to resolve the search. In many cases, we have examined parameter value changes larger than the **Jxsize** and **Jysize** move to 41 each.

Throughout these parameter changes the chance of success, as just defined, is robustly in the neighborhood of 70%. This suggests the possibility that natural selection may be acting more on time to decision and probability of avoiding failure to select any nest, rather than putting a premium on finding the best nest site. Is there something about the nest-site selection process as an algorithm that fits especially well with such values (robust on reasonably good, quick, avoiding failure)? These are large and important questions, to be addressed in future research.

4.2 Social Learning

The (somewhat problematic) parameter σ regulates a tradeoff in the model between imitation and exploration. The larger σ is, the more the tradeoff is tilted in favor of exploration. We found, however, that a minimum amount of exploitation, in the form of imitation, is necessary (within this model structure) if sites are to be chosen. Further, we found that σ can also be too low, in which case sites are chosen but not expeditiously. Here we see an explicit case of a social-individual learning dilemma that is at bottom a form of exploitation-exploration dilemma.

As noted earlier, social learning is the first and indeed primary impetus driving our interest in honeybee nest-site selection. By social learning we mean, roughly, learning by individuals in a population (or society) driven principally by observation of, and interaction with, other individuals. This is to be contrasted with learning in which individuals mainly observe and interact with nature, that is, with the world outside the society. Also as

noted above, the form of imitation in evidence here is grounded not in directly copying an observed behavior, but in interpreting a signal transmitted by a compatriot, which allows the bee to imitate the behavior at a distance. The observing bee copies not the dance itself, but the site (**spot**) indicated by the dance. The bee then examines the site herself, makes her own evaluation, and returns to the swarm to dance as she decides. What other forms of imitation are possible and how might they be used in heuristic algorithms or indeed by biological populations?

4.3 Metaheuristics

Regarding the model as a metaheuristic, we find that it robustly produces good but significantly non-optimal nest-site selections. Instead of a single-criterion metaheuristic, the algorithm is best seen as balancing three objectives: choose the best of the available sites in the neighborhood, make the choice quickly, minimize risk of failing to choose a site. This inherent multiobjectivity in a metaheuristic is unusual from the point of view of the computational study of search and optimization, but is the norm in biological systems and, therefore, deserves careful study. In the normal course of things, single objective metaheuristics are modified or managed in some way to support multiobjective optimization. Of course, the nest-site selection algorithm is not a general purpose multiobjective optimization algorithm. An interesting question for future research is whether and how it can be adapted for other three objective problems and, most intriguing, whether there are families of inherently multiobjective metaheuristics and if so, what their characteristics are. These matters, fascinating as they are, are well beyond the scope of this paper, but we hope it has contributed to raising them as important subjects.

4.4 Other Future Research

We close with very brief remarks that augment what we have already said about related future research.

Two complementary directions are we think very promising. The first is to generalize the nest-site selection model to include not just modifiable parameters, as is present now, but also to include swappable procedures. For many aspects of the model it is possible to imagine more than one applicable procedure. For example, a bee that has decided to imitate a dancer in effect chooses a dancer randomly, weighted by dance strength. That is a sensible criterion but there are others. Why not the site of the bee with the highest dance strength? Or the site with the most dancing bees? Procedures could be multiplied and configurations investigated just as parameter values are.

Second, and complementary, it is natural to inquire into how and why the observed (or hypothetical) parameter values and procedure configurations might have evolved. This is interesting both for social learning and for metaheuristics, and it suggests taking a *simulation optimization* approach to studying the model. (This is another idea championed

by Fred Glover. <http://leeds-faculty.colorado.edu/glover/fred%20pubs/332%20-%20Exploding%20Domain%20sim%20opt%204-24-06.pdf>. See also, e.g., [3, 15].) By for example learning weights on objective criteria we might learn what weights produced the observed parameter values and thus learn something about the forces behind the evolution of nest-site selection in honeybees.

References

- [1] BLUM, C., AND ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys* 35, 3 (2003), 268–308.
- [2] BOYD, R., AND RICHERSON, P. J. *Culture and the Evolutionary Process*. University of Chicago Press, Chicago, IL, 1988.
- [3] CARSON, Y., AND MARIA, A. Simulation optimization: Methods and applications. In *Proceedings of the 1997 Winter Simulation Conference* (1997), S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, Eds., pp. 118–126. <https://www.informs-sim.org/wsc97papers/0118.PDF>.
- [4] HOPPITT, W., AND LALAND, K. N. *Social Learning: An Introduction to Mechanisms, Methods, and Models*. Princeton University Press, Princeton, NJ, 2013.
- [5] KIMBROUGH, S. O. *Agents, Games, and Evolution: Strategies at Work and Play*. CRC Press, Boca Raton, FL, 2012.
- [6] KIMBROUGH, S. O., AND LAU, H. C. *Business Analytics for Decision Making*. CRC Press, Boca Raton, FL, 2016.
- [7] LALAND, K. N. *Darwin’s Unfinished Symphony: How Culture Made the Human Mind*. Princeton University Press, Princeton, NJ, 2017.
- [8] MESOUDI, A. *Cultural Evolution: How Darwinian Theory Can Explain Human Culture & Synthesize the Social Sciences*. University of Chicago Press, Chicago, IL, 2011.
- [9] NIKOLAEV, A. G., AND JACOBSON, S. H. Simulated annealing. In *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., 2nd ed., vol. 146 of *International Series in Operations Research & Management Science*. Springer, New York, NY, 2010, pp. 1–39.
- [10] PASSINO, K. M., AND SEELEY, T. D. Modeling and analysis of nest-site selection by honeybee swarms. *Behavioral Ecology and Sociobiology* 59, 3 (January 2006), 427–442.
- [11] RENDELL, L., BOYD, R., COWNDEN, D., ENQUIST, M., ERIKSSON, K., FELDMAN, M. W., FOGARTY, L., GHIRLANDA, S., LILICRAP, T., AND LALAND, K. N. Why

- copy others? Insights from the social learning strategies tournament. *Science* 328 (9 April 2010), 208–213.
- [12] RIBEIRO, C., AND HANSEN, P., Eds. *Essays and Surveys in Metaheuristics*. Kluwer, Boston, MA, 2001.
 - [13] RICHERSON, P. J., AND BOYD, R. *Not by Genes Alone: How Culture Transformed Human Evolution*. The University of Chicago Press, Chicago, IL, 2005.
 - [14] SEELEY, T. *Honeybee Democracy*. Princeton University Press, Princeton, NJ, 2010.
 - [15] TEZER, T., YAMAN, R., AND YAMANC, G. Evaluation of approaches used for optimization of stand-alone hybrid renewable energy systems. *Renewable and Sustainable Energy Reviews* 73 (2017), 804–853.
 - [16] VOSS, S. Metaheuristics: The state of the art. In *Local Search for Planning and Scheduling*, A. Nareyek, Ed., vol. 2148 of *Lecture Notes in Computer Science*. Springer, Heidelberg, Germany, 2001, pp. 1–23.
 - [17] VOUDOURIS, C., AND TSANG, E. Guided local search. In *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. Kluwer, 2003, pp. 185–218.