Addition to

BEEHAVE ODD Model Description

(BEEHAVE-Version 2013,

Becher et al., 2014)

as a supplement to the publication

Schödl et al., 2022

Author: Isabel Anna SCHÖDL
isabel.schoedl@ufz.de

Contents

Contents **2**

1 Preface **4**

2 The new varroa control module: ODD (Overview, Design concepts, and Details) **5**

2.1 Purpose and patterns 5
2.2 Entities, state variables, and scales 7
2.3 Process overview and scheduling 7
2.4 Design concepts 12
2.5 Initialisation 12
2.6 Input data 14
2.7 Submodels 14
2.7.1 Egg laying 14
2.7.2 Drone brood removal 18
2.7.3 Varroa treatment 19
2.8 Empirical drone data 21

3 Additional buttons in BEEHAVE **23**

4 Behaviour-Space Experimente **25**

5 Global variables **26**

6 Code for the output of an error file **28**

7 Code addition in the procedure *ParameterizationProc* **30**

8 Code addition in the procedure *MitesPhoreticPhaseProc* **35**

9 Code for counting mites **36**

10 Code for the sum of mites in the *miteOrganiser* **37**

11 Code for the egg laying procedure **38**

12 Code for the drone brood removal procedure **41**

13 Code for the varroa treatment with acaricides **46**

13.1 Varroa treatment with formic acid 46

13.2 Varroa treatment for oxalic acid 52

14 Code for running a burn-in phase in BEEHAVE **54**

Bibliography **56**

1. Preface

The original BEEHAVE model and corresponding user's guide & ODD are available at ; `https://www.beehave-model.net` . The model is programmed and executed using the freely available software platform NetLogo (Wilensky, 1999). For this study the BEEHAVE implementation BEEHAVE_BeeMapp2016 was updated to be used in NetLogo 6.2 (Wilensky, 1999). A full model description following the standard format ODD (Grimm et al., 2006, 2020) is included in the orignal model download.

This document contains information on the modified varroa module according to the good beekeeping practice in Germany following the ODD format.The corresponding NetLogo file *Beehave_BeeMapp2016updateversion6_IS_droneBroodRemoval_varroaTreatment.nlogo*, from which the model code excerpts in this document were copied and therefore line numbers correspond to this file, can be found in the COMSES library.

2. The new varroa control module: ODD (Overview, Design concepts, and Details)

The model description for the extension of the existing BEEHAVE model with drone brood removal and varroa treatment as good beekeeping practice in Germany follows the ODD (Overview, Design concepts, and Details) protocol for the description of individual- and agent-based models (Grimm et al., 2006, updated by Grimm et al., 2020).

2.1 Purpose and patterns

The BEEHAVE model (Becher et al., 2014) was developed to investigate how different stress factors (individually or in combination) affect the performance and possible decline/collapse of individual managed honey bee colonies.
The aim of the extension of BEEHAVE in this study was to map the good beekeeping practice of varroa control in Germany. The relevant measures within the varroa control strategies include drone brood removal as a varroa trap and the treatment of bee colonies with organic acaricides to kill varroa mites. [1].

The updated procedures in the model were evaluated based on the ability to reproduce the following patterns.

Patterns for drone brood removal:

> Pattern 1: Reduction of mite pressure of 50 - 67% by consistent drone brood removal. Calis et al., 1999, Charrière et al., 2003 and S. Berg, R. Odemer ()pers. comm.) indicate a mite pressure reduction of up to 50%. G. Liebig states that a colony in which no drone brood is cut is about two to three times more infested with mites than if drone brood were regularly removed. This results in a reduction of mite pressure by 50 – 67 % (pers. comm.).
>
> Pattern 2:
> Comparing the intervals between drone brood removal days, there should be no signifi-

cant difference between a two week and three week gap. This is due to fact that after 21 days (i.e. three weeks) drone pupae on the drone brood frame are not yet fully developed to emerge from their cells (S. Berg, pers. comm.).

Pattern 3:
Drone brood removal is not sufficient as sole control measure against the Varroa mite (R. Odemer, pers. comm.).

2.2 Entities, state variables, and scales

No new units or state variables were introduced for the extensions and the scales were not changed. The ODD model description by BEEHAVE can be found in Appendix S1, Supplement in Becher et al. (2014).

2.3 Process overview and scheduling

In order to be able to adapt drone brood removal to the good beekeeping practice in Germany, the egg laying of the queen was newly implemented within the colony model. The model is progressing in daily steps. The queen's egg laying occurs in the model step relatively at the beginning of the procedures, after the ageing of the worker and drone egg cohorts (Table 1). Drone brood removal and varroa treatment take place on the respective calendar days in BEEHAVE within the procedure for beekeeping practices (*BeekeepingProc*), and in the model step after running the mite model towards the end of all procedures.

Table 1: Order of the procedures of a model step in BEEHAVE, which can be found in the *Go* procedure; the procedures relevant and changed/added in this work are printed in bold.

1. *DailyUpdateProc*
2. *SeasonProc_HoPoMo*
3. *WorkerEggsDevProc*
4. *DroneEggsDevProc*
5. ***NewEggsProc***
6. *SwarmingProc*
7. *WorkerEggLayingProc*
8. *DroneEggLayingProc*
9. *WorkerLarvaeDevProc*
10. *DroneLarvaeDevProc*
11. *NewWorkerLarvaeProc*
17. *WorkerIHbeesDevProc*
18. *DronesDevProc*
19. *BroodCareProc*
20. *NewIHbeesProc*
21. *NewDronesProc*
22. *MiteProc*
23. ***BeekeepingProc***
24. *DrawIHcohortsProc*
25. *GenericPlotClearProc*
26. *Start_IBM_ForagingProc*
27. *CountingProc*

Continued on next page

Table 1 – *continued*

12. *NewDroneLarvaeProc*
13. *WorkerPupaeDevProc*
14. *DronePupaeDevProc*
15. *NewWorkerPupaeProc*
16. *NewDronePupaeProc*
28. *CountMitesInMOProc*
29. *PollenConsumptionProc*
30. *HoneyConsumptionProc*
31. *DoPlotsProc*

If drone brood removal is not used in the model, egg laying occurs at 20 % drone eggs per day within the drone egg laying season. This corresponds to continuous drone egg laying. With drone brood removal, drone egg laying is no longer strictly continuous, but is organised in so called drone egg laying events. This modelling is based on the realisation of drone brood removal by beekeepers: the beekeeper hangs the drone brood frame in the hive on a certain day. If the drone egg-laying season has already started, the queen can immediately start to lay eggs in the drone brood cells (yellow boxes in figure 1). In BEEHAVE, the number of drone eggs per day is calculated by the number of drone cells per drone brood frame (*cellsDroneBroodFrame*) and the number of days the queen needs for this (*droneFrameEgglayingDuration*) (orange boxes in figure 1). On the drone brood frame, the drone brood develops from eggs to larvae to pupae depending on the day of egg laying. In practice, if the drone brood frame is cut, i.e. the drone brood is removed, the drone brood can be at most as many days old, as have passed since the frame was put in the bee hive. This dependency of drone egg laying events on drone brood removal is illustrated with the corresponding parameters in BEEHAVE in figure 2. A drone egg laying event must always precede a drone brood removal day. The number of drone egg laying events and drone brood removal days depends on how often the beekeeper removes the drone brood frame.

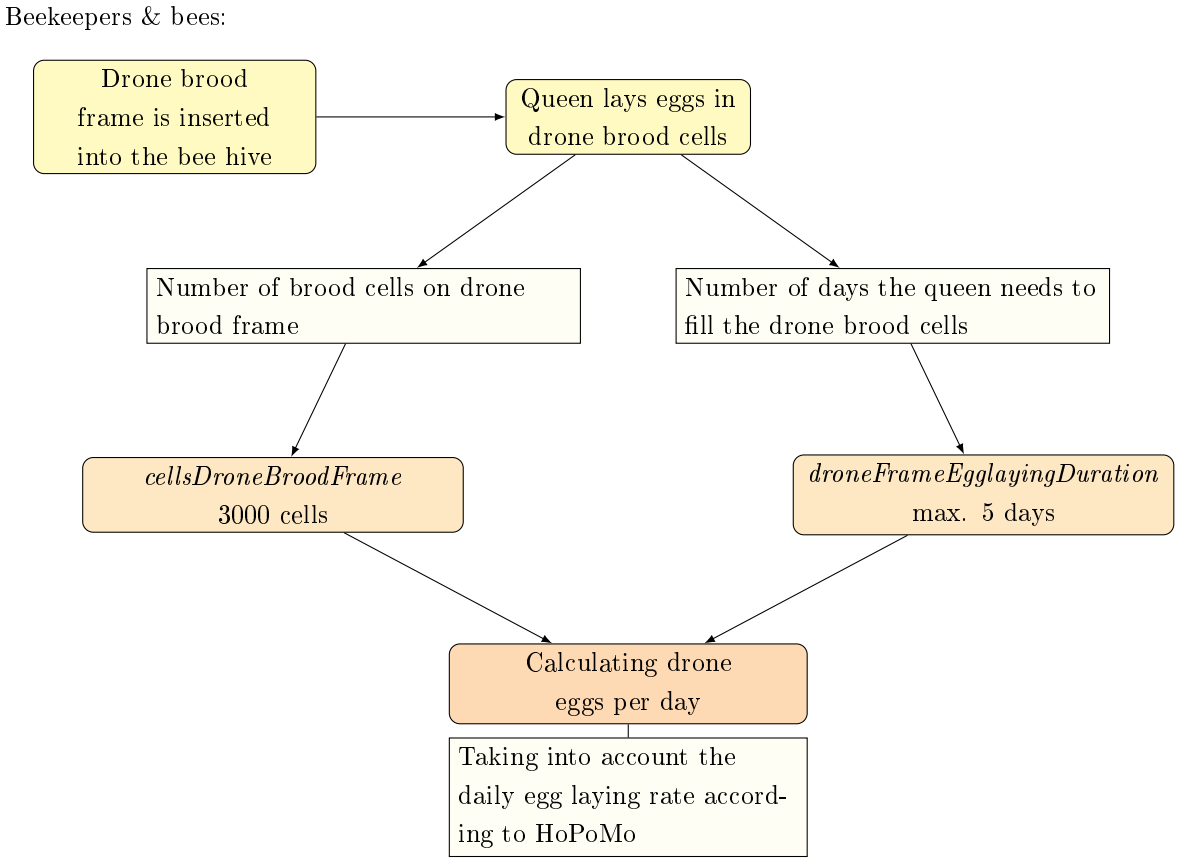

Figure 1: Modelling egg laying in BEEHAVE;
HoPoMo is an established honey bee model whose calculation of the daily egg laying rate of the queen is used in BEEHAVE (after Schmickl and Crailsheim, 2007).

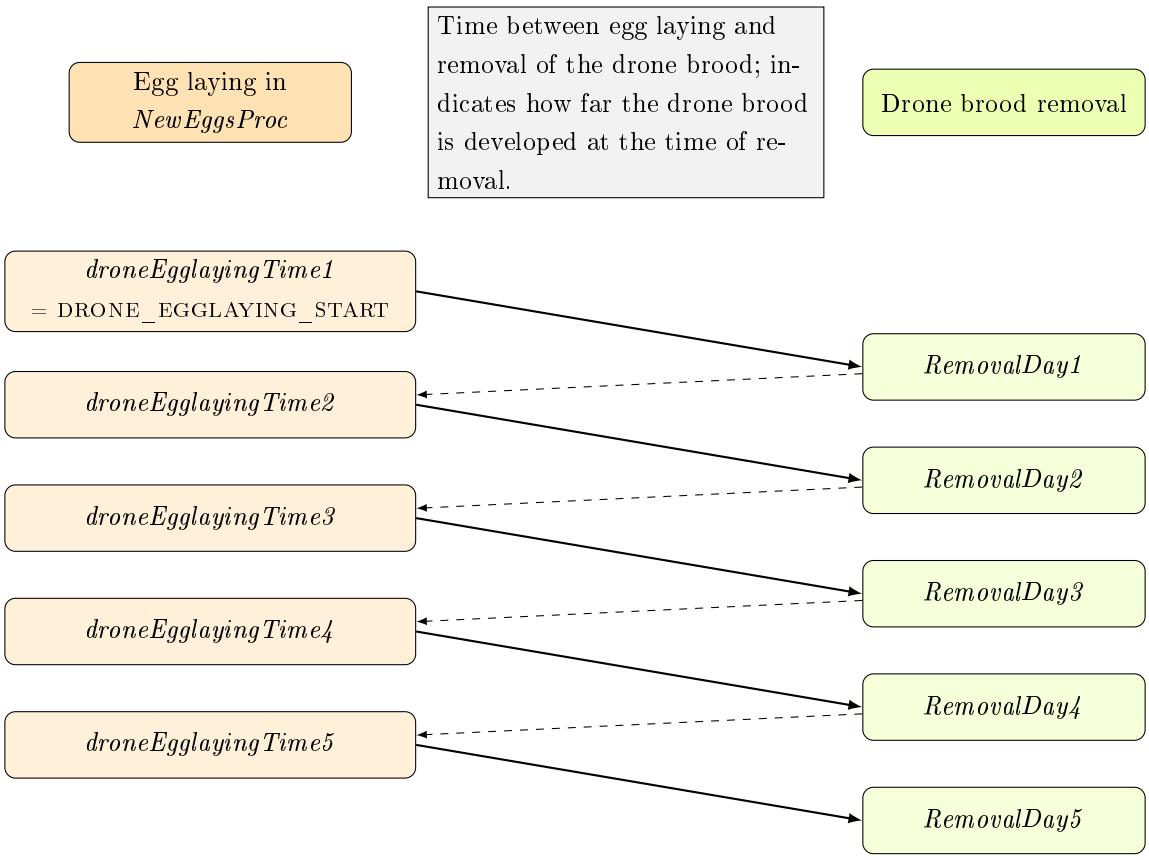

Figure 2: Modelling the relationship between egg laying and drone brood removal in BEEHAVE.
Arrow meaning:
——→ When is the drone brood being removed, after 2 weeks, 3 weeks?
←- - - - After the drone brood removal: new drone brood frame inserted and thus new drone egg laying event?

For the treatments with the organic acaricides, formic and oxalic acid, different conditions apply and the applications differ. Table 2 gives a summary and comparison of these two acids, on the basis of which varroa treatment has been implemented in BEEHAVE.

Table 2: Summary and comparison of varroa treatment with the organic acids formic and oxalic acid (S. Berg, R. Odemer, C. Otten, J. Pistorius, pers. comm.).

	Formic acid	**Oxalic acid**
Reason for application	Mite treatment after the end of honey harvest	Residual mite removal before overwintering period to allow the colonies to start the following year with a low initial mite load
Conditions	Temperature on the day(s) of application must be between 20 °C and 30 °C	Colony must be broodless; temperature should be below 5 °C
Effect	Effects phoretic mites and mites in broodcells	Effects only phoretic mites
Timing	Typical application: two treatments four weeks apart between July & September	Usually carried out in November/December
Application	A possible course of treatment is the evaporation of the formic acid solution over three days during the first treatment. If a second treatment is performed, the ideal evaporation amount per day is often no longer achieved due to the weather, which is why the treatment duration is increased (e.g. to six days).	The oxalic acid is dripped into the closely packed honeycomb aisles. The beekeeper cannot reach all the bees with this method. However, the bees spread the oxalic acid further in the colony by their cleaning instinct.

Continued on next page

Table 2 – *Continued*

	Formic acid	**Oxalic acid**
Negative effects	Damage to open brood and partly to freshly slipped bees	May only be applied once, as a second treatment would increase the mortality rate of the bees too much; winter bees would be treated twice → treatments in summer/autumn affect different generation of bees and can therefore be repeated several times

In BEEHAVE, the parameter temperature is only indirectly included in the proportion of hours of possible foraging. For the temperature-dependent treatment with formic acid, it is therefore assumed that the temperature requirements are met on the treatment days set in BEEHAVE.

2.4 Design concepts

Existing concepts in BEEHAVE are still valid.

2.5 Initialisation

The following illustrations show the part of the BEEHAVE graphical user interface related to the egg laying, drone brood removal and varroa treatment. Section 2.7 describes the individual processes and their parameters in more detail.

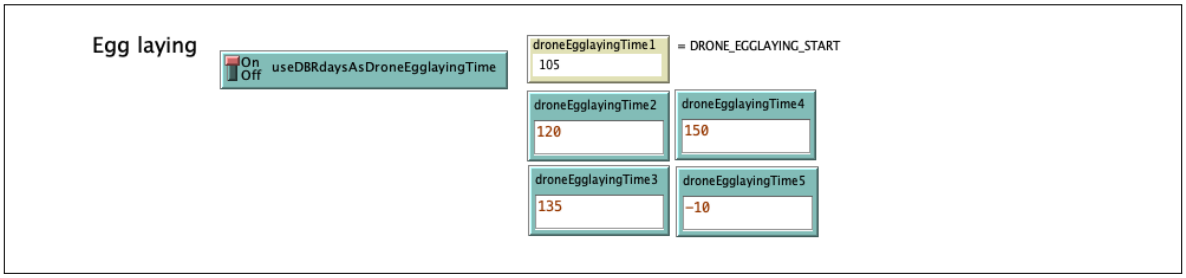

Figure 3: Graphical user interface of BEEHAVE: egg laying configuration; negative values mean that this drone egg laying event does not exist.

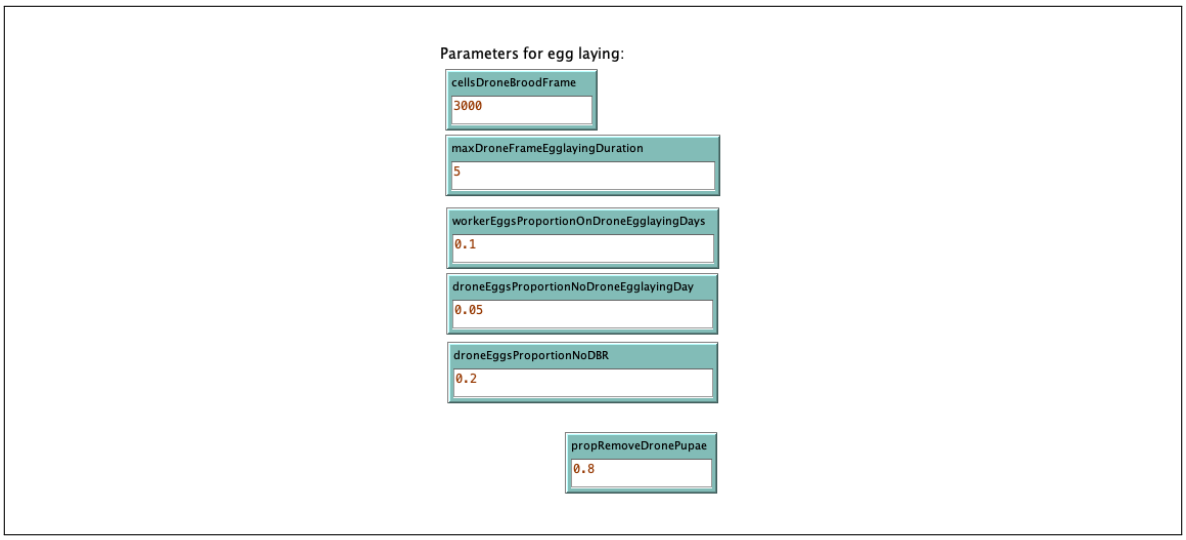

Figure 4: Graphical user interface of BEEHAVE: fixed egg laying parameters.

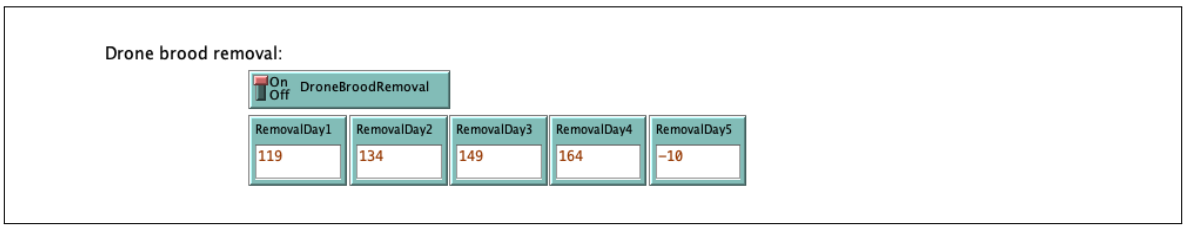

Figure 5: Graphical user interface of BEEHAVE: drone brood removal; the gap between the start of the drone egg laying event (one day after the last drone brood removal day) and the next drone brood removal is 14 days in this example; negative values mean that this drone brood removal day is not used.

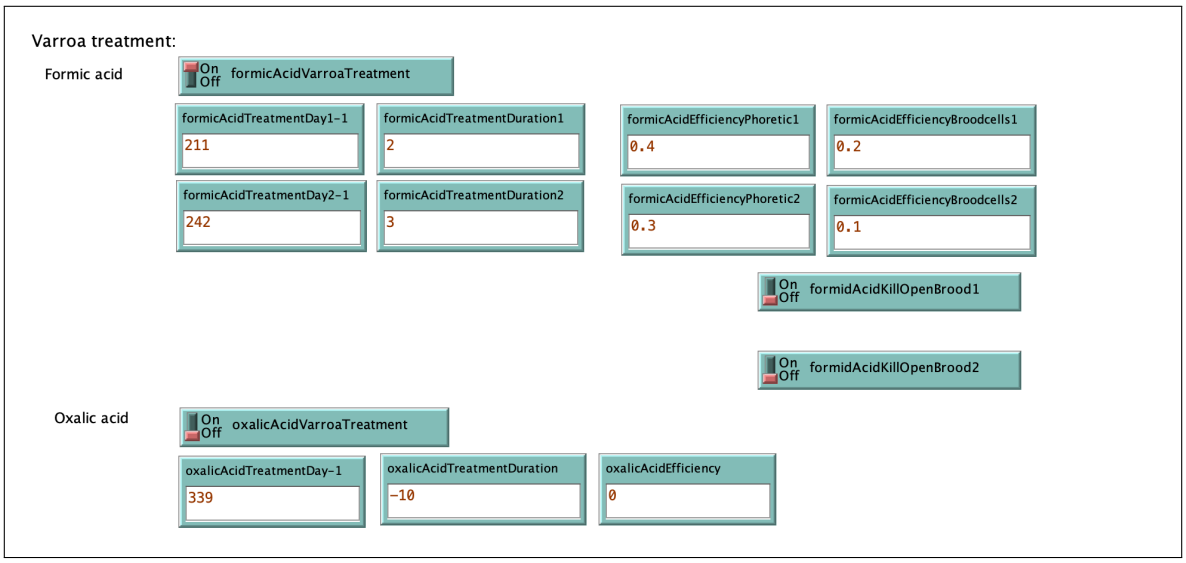

Figure 6: Graphical user interface of BEEHAVE: varroa treatment.

2.6 Input data

No landscape and weather data were imported.

2.7 Submodels

2.7.1 Egg laying

In BEEHAVE, the proportion of worker and drone brood laid by the queen is pre-determined in the model code. In a real bee colony this is determined by the workers in the colony building the cells for the desired brood (worker/drone brood). The queen then lays the appropriate eggs (fertilised/unfertilised) in them (see e.g. Pratt, 1998). In order to correctly represent the number of worker and drone brood, the mite dynamics and the drone brood removal in BEEHAVE, the egg laying of the queen was restructured in the procedure *NewEggsProc*. First, a summary of the literature on the proportion of drone brood in a colony is given below, as this is a crucial parameter for the development of the number of mites in the colony.

Whether an increased number of drone brood is at the expense of worker brood depends on the individual colony (Imdorf et al., 2008, pp. 53–54). In the literature, there are studies confirming a reduction of workers and thus a loss of colony strength (see e.g. Büchler, 1996; Liebig, 1997). In other studies, this is not the case, suggesting that bees did not invest in drone

brood until the supply of a sufficient number of workers (brood) was ensured (see e.g. Allen, 1965; Dettli, 2009; Dettli, 2019). In the following, the term natural bee colonies refers to bee colonies that live in the wild. Natural hive colonies are accordingly managed hive colonies, which, however, are left to build the combs and divide the brood into workers and drones by themselves. Natural bee colonies and colonies that are left to build their own combs with empty frames produce more drone brood than colonies where a large part of the brood area is pre-determined to be used for either worker or drone brood. In the literature, the proportion of drone brood in the total brood area is given as $17 \pm 3\%$ (Seeley and Morse, 1976) for natural bee colonies and $20 - 30\%$ (Büchler, 1996; Liebig, 1997; Dettli, 2009) for natural hive colonies.

In the previous versions of BEEHAVE, drones and their number in the colony were only relevant for the mite model and therefore played a subordinate role overall. In addition to the daily egg laying of the worker bee eggs, 4% of these were defined as drone eggs. These 4% were determined according to Wilkinson and Smith (2002), who calculated them from Allen (1963) and Allen (1965). In the two latter papers the area on which drone brood was placed on the brood combs was determined and compared, not the actual number of drone brood(eggs/larvae/pupae). These areas can be converted into numbers: per cm^2 brood area Allen (1958) states 4 worker cells, Dadant (1975) 4.29. For drone cells per cm^2 2.60 (Dadant, 1975) and 3 (Free, 1977) can be found in the literature. In this case, 3 drone brood cells and 4 worker brood cells per cm^2 were used for conversions. By this, the ratios of the numbers of drone brood to worker brood and drone brood to total brood can be calculated on the respective measurement days. It is not completely clear how Wilkinson and Smith (2002) arrive at the 4%; one possible explanation is the following: Tables 2 and 3 in Allen (1965) show, among other things, the averaged maximum brood area of worker and drone brood. If one converts this averaged maximum area into brood cells, the drone brood accounts for about 7.1% of the worker brood (year 1962) and 4.7% (year 1964). From this, the authors could have derived the daily 4% drone brood of the worker brood. In addition, Wilkinson and Smith (2002) give the following scenarios for daily proportions of drone brood out of worker brood based on their own considerations (without citing sources):

- 0% drone brood: any drone brood created by the bees is removed before it can hatch; as bees need drones for reproduction, this is not a standard for colony management.
- 4% drone brood: the colony is managed with drone frames which are removed regularly; drone brood that is placed on other frames is left in the colony.
- 20% drone brood: the bees are allowed to raise as much drone brood as they like, the beekeeper does not interfere.

Rowland and McLellan (1987) also refer to Allen (1965) and give a rate for drone eggs per day

of 24% of worker eggs on that day. The difference between the two rates from Allen (1965) (4% in Wilkinson and Smith (2002)) could be explained by the fact that Wilkinson and Smith (2002) used the data from the control bee colonies, which were not given a drone brood frame, for the calculation. Rowland and McLellan (1987) may have used the data from colonies where drone frames were available ad libitum and the bees reared more drone brood. If we calculate the proportion of drone brood to worker brood (converted to brood cells) with the maximum brood areas (Tables 2 and 3 in Allen 1965) as described above for the 4%, we arrive at about 16.2% for the treated group in 1962, and about 22.2% in 1964. The latter is at least close to the 24% used by Rowland and McLellan (1987).
In DeGrandi-Hoffman et al., 1989, according to Jay, 1974, no more than 5% of the total brood is expected to be drone brood. The bee-mite model of Torres and Torres (2020) calculates an average of 2.5% drone eggs in the total number of eggs; the maximum in summer is 4%. The empirical comparative values used by the authors refer to Sumpter and Martin (2004), which in turn refer to data from Seeley (2002). In Sumpter and Martin (2004) the proportion of drone eggs to worker eggs per day is given as 1% in spring and autumn and 3.3% in summer. In table 3, section 2.8, a selection of empirical data on the number of drone brood and the proportion of drone brood in the total brood is presented. Depending on the beekeeping conditions, the percentage of drone brood in the worker brood or the total brood varies during the year and between the sources within one point in time.
The figures in the table are averaged over several bee colonies. Nevertheless, they show the high variability, which is already caused by different regions/locations/management of the bee colony.
In summary, the different proportions of drone brood in the literature show a strong dependence on external (e.g. colony location) and beekeeping (e.g. hive frame) conditions. Uncertainty about the proportion of drone brood in the colony and the lack of literature on this becomes clear.
Although the literature listed gives mean values calculated from several bee colonies, the outstanding individuality of each colony should not be ignored. In the updated *NewEggsProc* the daily egg laying continues to follow the curve calculated in the established HoPoMo model (Schmickl and Crailsheim, 2007). However, a distinction is now made between continuous and periodic ("pulsed") drone egg laying. The latter is relevant for drone brood removal.

The following egg-laying scenarios are now possible:

a) Pulsed egg laying with drone brood removal:
   The queen starts to fill the imaginary drone brood frame in the model with drone eggs with the start of the drone egg laying season. For a maximum duration of 5 days, mostly drone eggs are produced. A background level of worker eggs of 10% of the total eggs per

day was set to avoid a significant drop in colony strength over the season. The queen lays 90% drone eggs until a total of about 3000 drone eggs (fixed number of cells on a drone brood frame) have been laid. If this number has not been reached after 5 days, there are fewer eggs in this drone egg laying event and the drone brood frame has not been completely filled. If one were to assume that the queen always lays eggs until 3000 are reached, this would be at the expense of worker colony strength, which is not desirable. It is possible that a total of slightly more than 3000 eggs will be laid in the course of the current drone egg-laying event. Due to the natural fluctuations in the bee colony and the possibility to lay drone brood outside a drone brood frame, this does not represent a contradiction to the fixed number of brood cells on a drone brood frame. The following drone egg laying event is determined by the next drone brood removal day, since after removal the beekeeper re-places a drone brood frame in the colony. If one is not on a day of a drone egg laying event, 95% worker eggs will be laid and hence 5% drone eggs. Even in a colony managed with drone brood frames, some drone brood cells are laid outside the drone brood frame, which is what this is meant to represent.

b) Pulsed egg laying without drone brood removal:
It simulates managing a colony with a drone brood frame, but without cutting it. Every 25 days ($\rightarrow$ development time of a drone: 24 days) the queen re-lays drone eggs in the brood cells of the frame. This is a stylised representation of the drone brood frame, as in reality the queen probably re-lays drone eggs in the cells as soon as a drone has hatched from them. This postponed re-egg-laying is negligible, as it should not strongly influence the overall dynamics.

c) Continuous egg laying:
Within the drone egg laying season, 20% of the eggs laid daily are drones, 80% are worker eggs. Out of season, 100% are worker eggs. This is significantly more than the previous 4%. The continuous drone egg laying simulates more a natural colony; according to the literature, such a colony should have more drone brood than a colony managed with middle walls and a drone frame. In a) the number of drone eggs is obtained by filling the drone brood frame with 3000 cells. In a drone brood removal scenario (see section 2.7.2 for further information on drone brood removal) with typically four removal days, about 10% of the total number of eggs per year are drone eggs. This proportion seems relatively high, but as described it is caused by the cells on the drone brood frame. According to the literature, the colony with continuous drone egg laying should have a higher proportion of drone brood. To achieve the same proportion of 10% drone eggs per year, 20% drone eggs are needed of the daily laid eggs within the drone egg laying season. As 20% is already a high proportion, no more than this is used. This 20% was used so that in the whole year at least approximately as many drone eggs are laid with

continuous egg laying as with pulsed egg laying.

The egg-laying part of the BEEHAVE graphical user interface is shown in figure 3, while the fixed parameters can be seen in figure 4. The corresponding model code can be found in chapter 11.

2.7.2 Drone brood removal

The drone brood cutting implemented in the 2016 version of BEEHAVE has not yet been tested and the effects on the bee colony with regard to the control of the varroa mite population by drone brood removal have not yet been investigated. The implementation of drone brood removal in this first version is such that only the capped drone pupae are removed. Removing only the drone pupae is equivalent to manually cutting the drone brood without a special drone brood frame. In the original version of drone brood cutting in BEEHAVE, one can choose between two variants: either the drone pupae are removed continuously, or one specifies up to five calendar days on which all drone pupae are removed.

The drone brood removal procedure was updated so that it is adapted to the practice of beekeepers in Germany. What beekeepers can influence is the time when the drone brood frame is inserted into and removed from the bee hive. Consequently, only the drone brood that was placed on the drone brood frame in the last drone egg laying event can be removed on a drone brood brood removal day. A minimum interval of 14 days is assumed between placing and removing the drone brood frame. The number of days needed for the queen to fill the drone brood frame was set to a maximum of 5 days. In BEEHAVE, the procedure for removing drone brood is as follows: the drone brood frame remains in the colony for at least 14 days, therefore there can be no more drone eggs or larvae on the drone brood frame (as these have an age of $1-2$ ad $3-9$ days, respectively). In order to determine which cohorts of drone pupae are removed, the age of those to be removed is first determined depending on the duration of the last drone egg laying event and the interval between the drone egg laying event and the day of the drone brood removal. For example, if cutting is done 14 days after the start of the last drone egg-laying event, then if the egg laying event lasts 5 days, the oldest drone pupa cohort will be 14 days old and the youngest 10 days old. These age cohorts are then set to 0 in BEEHAVE and, if present, so are the mites in the associated *miteOrganiser*. Mites can be present in all ages of drone pupa cohorts.

The corresponding part on the graphical user interface of BEEHAVE is shown in figure 5. If the drone brood is to be removed less than five times, -10 is entered. However, the days on which drone brood removal is to be carried out should not be interrupted by a -10. The option *useDBRdaysAsDroneEgglayingTime* on the graphical user interface in BEEHAVE should be

switched on for drone brood removal. It causes the subsequent drone egg laying events to be carried out accordingly when the individual drone brood removal days are specified. For example, if cutting is done on day 119, this option will automatically set day 120 as the start day of the next drone egg laying event.
The model is code can be found in chapter 12.

2.7.3 Varroa treatment

In the 2016 version of BEEHAVE, one or two days can be specified for a varroa treatment with a chemical agent, from which an acaricide acts in the colony for a specified duration (default value: *TreatmentDuration* = 40). This implementation of a synthetic acaricide acts only on the phoretic mites and the daily additional mortality is given with a default value of 11.5% (*EfficiencyPhoretic* = 0.115). Thus, increasing the mortality of the phoretic mites by 11.5% each day for 40 days gives a total efficiency of

$$1 - ((1 - EfficiencyPhoretic)^{TreatmentDuration}) = 0.998. \quad (1)$$

It should be noted that due to the given implementation, the acaricide is actually effective for (*TreatmentDuration* + 1)-days, but the total effectiveness is calculated with the value of *TreatmentDuration*. In addition, the option to remove uncapped brood and/or all mites in the cells every day during the treatment period can be selected.

This way of implementing varroa treatment is not consistent with the strategies of varroa control as practised in Germany. BEEHAVE was adapted so that the two organic acids formic acid and oxalic acid can be used for varroa treatment. Both acids are only effective for a few days in the bee colony. In contrast to the use of synthetic acaricides, they have a short but strong effect.

The treatment of the bee colony in BEEHAVE with formic acid can be carried out on up to two treatment days, where the start of the treatment, the duration and efficiency are stated. The formic acid acts both in the brood cells and on the phoretic mites, but with different efficiencies. The effect on the latter is comparatively higher (Calderón et al., 2000; Steube et al., 2021).
The removal of phoretic mites is not difficult in the implementation and the number of phoretic mites is reduced daily by the value of the efficiency on that day. For example, if *formidAcidEfficiencyPhoretic1* = 0.5 and there are 1000 mites in the colony, 500 mites are removed from the colony. To remove the mites in the brood cells, one has to work with the *miteOrganiser*. In contrast to drone brood removal, both the drones (*droneCellListCondensed* in the *miteOrganiser*) and the workers (*workerCellListCondensed* in the *miteOrganiser*) are considered.

In both CellListCondensed, each entry in the mite abundance classes with one or more mites reduces by the desired efficiency. The cells thus removed from a mite abundance class are then moved to the class without mites. To illustrate, 50 % of the mites are removed from the example *workerCellListCondensed* [10 4 8 0 2], which contains a total of $4 \cdot 1 + 8 \cdot 2 + 0 \cdot 3 + 2 \cdot 4 = 28$ mites. Remove 2 cells with 1 mite, 4 cells with 2 mites and 1 cell with 4 mites. The altered *workerCellListCondensed* then looks like this: [17 2 4 0 1], with $2 \cdot 1 + 4 \cdot 2 + 0 \cdot 3 + 1 \cdot 4 = 14$ mites. For non-integer values (e.g. if 1.5 mites are to be removed), BEEHAVE uses the difference between this decimal number and the next lower integer (here: 1) to determine at random whether another mite is to be removed.

Since only the phoretic mites are affected by the treatment with oxalic acid, nothing more needs to be done in BEEHAVE than to reduce the number of phoretic mites by the desired amount.

In view of the fact that the formic acid, if it affects the colony, harms the open brood, there is an option to kill the existing open brood after the ant acid treatment. However, this option was not used in the BEEHAVE simulations carried for this paper.

The part of the graphical user interface for varroa treatment is shown in figure 6. Note that on the graphical interface of BEEHAVE the calendar day of the treatment minus one must be given. Thus, if treatment is to take place on day 215, day 214 must be entered. This programming aspect is necessary so that the indicated duration of the treatment also corresponds to the number of days with treatment. Otherwise, the actual duration is extended by one day. As described above, this is the case with the originally implemented version of varroa treatment. If you want to skip a treatment, you have to set -10 to the corresponding day on the graphical interface. If you only want to treat once with formic acid, put -10 at *formidAcidTreatmentDay2-1*.
The respective model code can be found in chapter 13.

2.8 Empirical drone data

Table 3: Empirical data (mean values) on the number of drone brood and the proportion of drone brood in the total brood in bee colonies under different conditions. In some cases the numbers were calculated from the areas of brood with 4 worker cells/cm^2 (Allen, 1958) and 3 drone cells/cm^2 (Free, 1977). The following conditions applied to the data collected, which were averaged over several colonies:
Allen (1963): Control = no drone brood frame, only frames with middle walls, Treated = one drone brood frame additional to the frames with middle walls
Allen (1965): Control = no drone brood frame, Treated = drone brood frame ad libitum;
Mesquida (1976): without special treatment;
Dettli (2019): Middle wall = only frames with middle walls, Natural = empty frames, which the bees expand themselves

Time	# drone brood	% drone brood/ worker brood	% drone brood/ total brood	Bee hive condition	Source
March	0.0	0.0	0.0	/	3
	115.0	0.4	0.3	Middle wall	4
	219.0	0.6	0.6	Natural	4
End of April	77.0	0.9	0.9	Treated	1
	97.0	1.1	1.1	Control	1
	4437.0	63.7	38.9	Middle wall	4
	7708.0	80.5	44.6	Natural	4
Mid/ end of May	464.0	3.0	2.9	Treated	1
	290.0	1.9	1.9	Control	1
	987.0	7.5	7.0	Treated	2
	156.0	1.1	1.1	Control	2
	4593.0	22.3	16.9	/	3
	4102.0	138.0	38.0	Middle wall	4
	7629.0	204.9	67.2	Natural	4
	3638.0	14.8	12.9	Treated	1

1 Allen (1963); 2 Allen (1965); 3 Mesquida (1976); 4 Dettli (2019)

Continued on next page

Table 3 – *Continued*

Time	# drone brood	% drone brood/ worker brood	% drone brood/ total brood	Bee hive condition	Source
End of June	1180.0	4.8	4.6	Control	1
	4161.0	/	/	Treated	2
	1143.0	/	/	Control	2
	8418.0	29.9	23.0	/	3
	3008.0	26.9	21.2	Middle wall	4
	4675.0	40.5	28.9	Natural	4
Mid/ end of July	4141.0	18.2	15.4	Treated	1
	1625.0	5.3	5.0	Control	1
	4761.0	20.2	16.8	Treated	2
	639.0	2.5	2.4	Control	2
	2160.0	10.2	9.2	/	3
	572.0	2.5	2.4	Middle wall	4
	1116.0	4.9	4.7	Natural	4
Mid/ end of August	2322.0	/	/	Treated	2
	543.0	/	/	Control	2
	137.0	0.3	0.3	Middle wall	4
	357.0	0.4	0.8	Natural	4

1 Allen (1963); 2 Allen (1965); 3 Mesquida (1976); 4 Dettli (2019)

3. Additional buttons in BEEHAVE

Buttons were added to the graphical interface of BEEHAVE in order to be able to carry out the experiments analysed in the paper Schödl et al., 2022 (Fig. 7). Clicking on the respective button in BEEHAVE sets the correct values of the parameters on the graphical interface. BEEHAVE simulations are then run with a burn-in phase of one year, therefore, e.g. mites are added in the second year. Hence, varroa control measurements start in the second year as well.

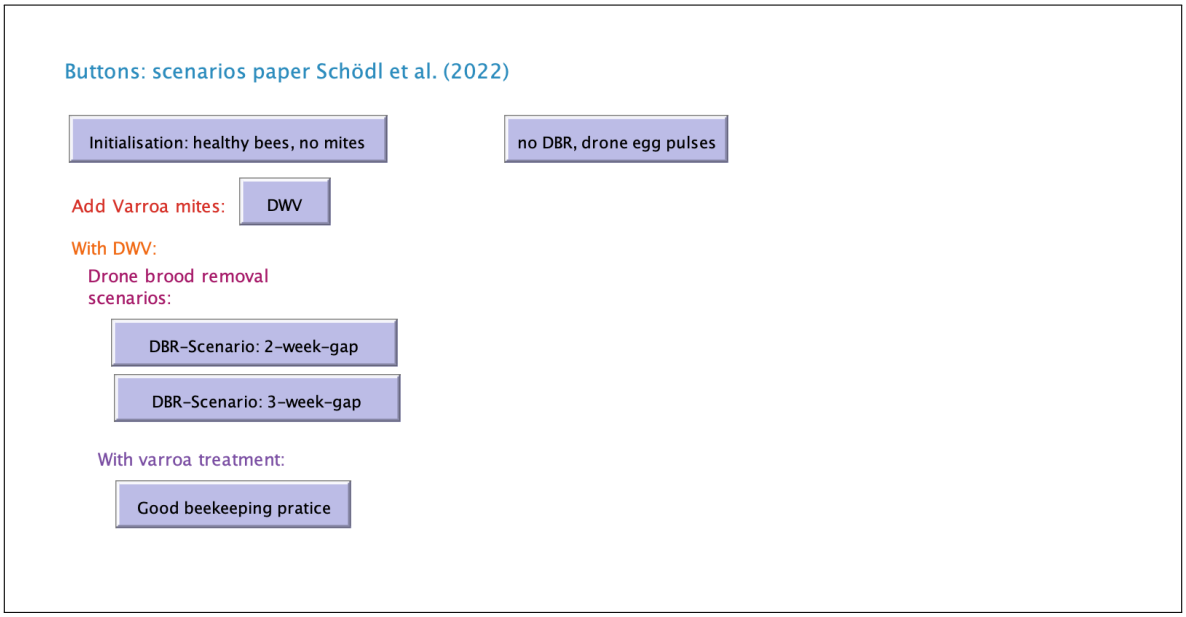

Figure 7: Graphical user interface in BEEHAVE: additional buttons.

Button description:

Initialisation: healthy bees, no mites
Intitialisation of a healthy bee colony without varroa mites; with continuous drone egg laying.

no DBR, drone egg pulses
Healthy colony without varroa mites with pulsed drone egg laying.

DWV

Healthy mites infected with DWV are added to the colony.

DBR-Scenario: 2-week-gap

Drone brood removal scenario with 4 drone brood removal days and a 2 week gap between them.

DBR-Scenario: 3-week-gap

Drone brood removal scenario with 4 drone brood removal days and a 3 week gap between them.

Good beekeeping practice

Good beekeeping practice: drone brood removal, scenario 2-week-gap, in combination with two formic acid treatments as varroa treatment.

4. Behaviour-Space Experimente

In the following, the figures in Schödl et al., 2022 are assigned to the behaviour space experiments carried out in BEEHAVE.

Fig. 2: Figure2_egglaying_pulsedEggs & Figure2_egglaying_contEggs

Fig. 3: Figure3_beeColonyWithMites

Fig. 4: Figure4and9_beeColonyWithMites_contEggs_noDBR & Figure4and7_beeColonyWithMites_4removalDays_2weekGap & Figure4_beeColonyWithMites_4removalDays_3weekGap

Fig. 5: Figure5_beeColonyWithMites_4removalDays_2weekGap_varyFactorDrones & Figure5_beeColonyWithMites_contEggs_noDBR_varyFactorDrones

Fig. 6: Figure6_beeColonyWithMites_4removalDays_2weekGap_reinvasion & Figure6_beeColonyWithMites_contEggs_noDBR_reinvasion

Fig. 7: Figure4and7_beeColonyWithMites_4removalDays_2weekGap & Figure7and8_beeColonyWithMites_goodBeekeepingPractice

Fig. 8: Figure7and8_beeColonyWithMites_goodBeekeepingPractice

Fig. 9: Figure4and9_beeColonyWithMites_contEggs_noDBR & Figure9_beeColonyWithMites_4removalDays_2weekGap_varyPropRemoveDronePupae & Figure9_beeColonyWithMites_4removalDays_3weekGap_varyPropRemoveDronePupae

5. Global variables

Added global variables in *globals*.

```
204 current_seed ; To report the current seed in the Behaviour-Space
       experiments.
205 BugMessage ; To report a bug message and write it into a file.
206
207 allMitesInMO ; Counter for all mites in brood cells (healthy + infected).
208 mitesInfectedInMO ; Counter for all infected mites in brood cells.
209
210 ; For NewEggsProc.
211 ;cellsDroneBroodFrame ; Parameter for the number of total cells on a
       drone brood frame (estimated, parameter comprises both sides of the
       brood frame).
212 ;maxDroneFrameEgglayingDuration ; Maximum duration (days) of a drone
       egglaying event.
213 droneFrameEgglayingDuration ; Duration of last drone egglaying event.
214 currentNumDroneEggs ; Counter for how many drone eggs have been laid so
       far in the current drone egglaying event.
215 countDroneEgglayingDays ; Counter of duration of current drone egglaying
       event.
216
217 ;workerEggsProportionOnDroneEgglayingDays; Background worker egg laying
       to avoid a siginficant drop in worker strength.
218 ;droneEggsProportionNoDroneEgglayingDay; Background drone egg proportion.
219 ;droneEggsProportionNoDBR; In drone egg season: proporiton of x% drone
       eggs per day of the total egg number on this day.
220
221 countTotalWorkerEggs ; Counter for total worker eggs layed in one year.
222 countTotalDroneEggs ; Counter for total drone eggs layed in one year.
223
224 ; For DroneBroodRemoval.
225 current_dronePupaeRemain ; To store the number of drone pupae removed in
       the cohort.
226 countRemovalDay ; Counter for the number of removal days realised so far.
```

```
227 removal_days ; List to save the removal days.
228 drone_egglaying_times ; List to save the drone egglaying start days.
229 days_between_removals ; List to save the days between each drone
        egglaying start day and corresponding removal day.
230 countMitesRemoved; Counter for mites removed.
231
232 ; For mite procedure:
233 ;these three parameters were local before, now they are global.
234 totalInfectedWorkers
235 totalHealthyWorkers
236 factorWorkers
237
238 infectedBees ; To count infected bees.
```

6. Code for the output of an error file

Own procedure: *export-parameters-to-file*.

In order to be able to output the 'BugMessage' in this file, a global variable was added to BEEHAVE, in which the error message is then stored. A Stop command was attached to each 'BugAlarm'. This ensures that the correct error message is printed in the error file. If the programme is stopped at the end of a simulation day, the error message may be overwritten (several times). The changes were made for all error messages after the *ParameterizationProc*.

```
7526 ;; Procedure to create an output file in case of a BugAlarm -> seed and
        specified parameters get reported into an .txt file.
7527 ; Saving location is the same as for this BEEHAVE NetLogo file.
7528 to export-parameters-to-file
7529   let filename (word "output_parameters_BugAlarm_" current_seed ".txt")
7530   Code for the output of an error file}
7531   Own procedure: \textit{export-parameters-to-file}.\.
7532
7533   In order to be able to output the 'BugMessage' in this file, a global
         variable was added to BEEHAVE, in which the error message is then
         stored. A Stop command was attached to each 'BugAlarm'. This
         ensures that the correct error message is output. If the programme
         is stopped at the end of a simulation day, the error message may be
         overwritten (several times). The changes were made for all error
         messages after the \textit{ParameterizationProc}.
7534
7535   if is-string? filename  ; to make sure filename is a string
7536   [
7537     if file-exists? filename   ; if the file already exists, it is
           deleted
7538     [
7539       file-delete filename
7540     ]
7541     file-open filename
7542     file-print
7543     (word
7544     "BugMessage; " BugMessage "\n"
```

```
7545      "current_seed; "   current_seed "\n"
7546      "Day; " Day "\n"
7547      "useDBRdaysAsDroneEgglayingTime; " useDBRdaysAsDroneEgglayingTime
            "\n"
7548      "DroneBroodRemoval; " DroneBroodRemoval  "\n"
7549      "RemovalDay1; " RemovalDay1  "\n"
7550      "RemovalDay2; " RemovalDay2  "\n"
7551      "RemovalDay3; " RemovalDay3  "\n"
7552      "RemovalDay4; " RemovalDay4  "\n"
7553      "RemovalDay5; " RemovalDay5  "\n"
7554      "formidAcidVarroaTreatment; " formidAcidVarroaTreatment  "\n"
7555      "formidAcidTreatmentDay1-1; " formidAcidTreatmentDay1-1  "\n"
7556      "formidAcidTreatmentDuration1; " formidAcidTreatmentDuration1  "\n"
7557      "formidAcidEfficiencyPhoretic1; " formidAcidEfficiencyPhoretic1  "\n"
7558      "formidAcidEfficiencyBroodcells1; " formidAcidEfficiencyBroodcells1
            "\n"
7559      "formidAcidTreatmentDay2-1; " formidAcidTreatmentDay2-1  "\n"
7560      "formidAcidTreatmentDuration2; " formidAcidTreatmentDuration2  "\n"
7561      "formidAcidEfficiencyPhoretic2; " formidAcidEfficiencyPhoretic2  "\n"
7562      "formidAcidEfficiencyBroodcells2; " formidAcidEfficiencyBroodcells2
            "\n"
7563      "oxalicAcidVarroaTreatment; " oxalicAcidVarroaTreatment  "\n"
7564      "oxalicAcidTreatmentDay-1; " oxalicAcidTreatmentDay-1  "\n"
7565      "oxalicAcidTreatmentDuration; " oxalicAcidTreatmentDuration  "\n"
7566      "oxalicAcidEfficiency; " oxalicAcidEfficiency
7567      )
7568    ]
7569    file-close ; Close file at the end.
7570  end
```

7. Code addition in the procedure *ParameterizationProc*

```
807 if DroneBroodRemoval = true
808 [
809   ; In case of useDBRdaysAsDroneEgglayingTime = true: use drone brood
        removal days for calculating the following drone egglaying start
        day.
810   ; -10 is used for a parameter not used.
811   if (useDBRdaysAsDroneEgglayingTime = true)
812   [
813     if RemovalDay1 != -10 [set droneEgglayingTime2 RemovalDay1 + 1]
814     ifelse RemovalDay2 != -10 [set droneEgglayingTime3 RemovalDay2 +
          1][set droneEgglayingTime2 -10]
815     ifelse RemovalDay3 != -10 [set droneEgglayingTime4 RemovalDay3 +
          1][set droneEgglayingTime3 -10]
816     ifelse RemovalDay4 != -10 [set droneEgglayingTime5 RemovalDay4 +
          1][set droneEgglayingTime4 -10]
817     if RemovalDay5 = -10 [set droneEgglayingTime5 -10]
818   ]
819
820   ; Check if the drone egglaying start days are within the drone
        egglaying period.
821   if ( (droneEgglayingTime2 != -10 and (droneEgglayingTime2 <
        DRONE_EGGLAYING_START or droneEgglayingTime2 >
        DRONE_EGGLAYING_STOP))
822   or (droneEgglayingTime3 != -10 and (droneEgglayingTime3 <
        DRONE_EGGLAYING_START or droneEgglayingTime3 >
        DRONE_EGGLAYING_STOP))
823   or (droneEgglayingTime4 != -10 and (droneEgglayingTime4 <
        DRONE_EGGLAYING_START or droneEgglayingTime4 >
        DRONE_EGGLAYING_STOP))
824   or (droneEgglayingTime5 != -10 and (droneEgglayingTime5 <
        DRONE_EGGLAYING_START or droneEgglayingTime5 >
        DRONE_EGGLAYING_STOP)) )
825   [
```

```
826     set BugAlarm true
827     set BugMessage "BUG Alarm: drone egglaying times are not within the
            drone egglaying period."
828     user-message BugMessage
829   ]
830
831   ; Check if the removal days are within the drone egglaying period (+
        the drone emerging age of 24 days).
832   if ( (RemovalDay1 != -10 and (RemovalDay1 < DRONE_EGGLAYING_START or
        RemovalDay1 > (DRONE_EGGLAYING_STOP + DRONE_EMERGING_AGE) ))
833   or (RemovalDay2 != -10 and (RemovalDay2 < DRONE_EGGLAYING_START or
        RemovalDay2 > (DRONE_EGGLAYING_STOP + DRONE_EMERGING_AGE)))
834   or (RemovalDay3 != -10 and (RemovalDay3 < DRONE_EGGLAYING_START or
        RemovalDay3 > (DRONE_EGGLAYING_STOP + DRONE_EMERGING_AGE)))
835   or (RemovalDay4 != -10 and (RemovalDay4 < DRONE_EGGLAYING_START or
        RemovalDay4 > (DRONE_EGGLAYING_STOP + DRONE_EMERGING_AGE)))
836   or (RemovalDay5 != -10 and (RemovalDay5 < DRONE_EGGLAYING_START or
        RemovalDay5 > (DRONE_EGGLAYING_STOP + DRONE_EMERGING_AGE))) )
837   [
838     set BugAlarm true
839     set BugMessage "BUG Alarm: drone brood removal days are not within
            the drone egglaying period (+ drone emerging age)."
840     user-message BugMessage
841   ]
842
843   ; Check that the egglaying times and removal days are without gaps,
        i.e. if only three removal days should happen
844   ;RemovalDay4 & RemovalDay5 are set to -10, and not e.g. RemovalDay1 &
        RemovalDay3.
845   if ( (RemovalDay1 = -10 and (RemovalDay2 > -10 or RemovalDay3 > -10 or
        RemovalDay4 > -10 or RemovalDay5 > -10))
846   or (RemovalDay2 = -10 and (RemovalDay3 > -10 or RemovalDay4 > -10 or
        RemovalDay5 > -10))
847   or (RemovalDay3 = -10 and (RemovalDay4 > -10 or RemovalDay5 > -10))
848   or (RemovalDay4 = -10 and RemovalDay5 > -10) )
849   [
850     set BugAlarm true
851     set BugMessage "BUG Alarm: check that the removal days are without
            gaps, i.e. -10 not followed by actual removal days."
852     user-message BugMessage
853   ]
854   if ( (droneEgglayingTime2 = -10 and (droneEgglayingTime3 > -10 or
        droneEgglayingTime4 > -10 or droneEgglayingTime5 > -10))
```

```
855   or (droneEgglayingTime3 = -10 and (droneEgglayingTime4 > -10 or
          droneEgglayingTime5 > -10))
856   or (droneEgglayingTime4 = -10 and droneEgglayingTime5 > -10) )
857   [
858     set BugAlarm true
859     set BugMessage "BUG Alarm: check that the egglaying times are
            without gaps, i.e. -10 not followed by actual egglaying times."
860     user-message BugMessage
861   ]
862
863   ; Check that a removal is preceded by drone egglaying event to make
          sure drone brood on the drone egglaying frame can be removed.
864   if ( (DRONE_EGGLAYING_START > RemovalDay1 and RemovalDay1 != -10)
865   or (droneEgglayingTime2 > RemovalDay2 and RemovalDay2 != -10)
866   or (droneEgglayingTime3 > RemovalDay3 and RemovalDay3 != -10)
867   or (droneEgglayingTime4 > RemovalDay4 and RemovalDay4 != -10)
868   or (droneEgglayingTime5 > RemovalDay5 and RemovalDay5 != -10)
869   or (droneEgglayingTime2 = -10 and RemovalDay2 != -10)
870   or (droneEgglayingTime3 = -10 and RemovalDay3 != -10)
871   or (droneEgglayingTime4 = -10 and RemovalDay4 != -10)
872   or (droneEgglayingTime5 = -10 and RemovalDay5 != -10) )
873   [
874     set BugAlarm true
875     set BugMessage "BUG Alarm: check that a removal event is preceded by
            a drone egglaying event."
876     user-message BugMessage
877   ]
878
879   ; In case a drone egglaying time is not followed by a removal day:
          user-message if that is intended.
880   if ( (DRONE_EGGLAYING_START > -10 and RemovalDay1 = -10)
881   or (droneEgglayingTime2 > -10 and RemovalDay2 = -10)
882   or (droneEgglayingTime3 > -10 and RemovalDay3 = -10)
883   or (droneEgglayingTime4 > -10 and RemovalDay4 = -10)
884   or (droneEgglayingTime5 > -10 and RemovalDay5 = -10) )
885   [
886     ifelse (RemovalDay1 = -10 and RemovalDay2 = -10 and RemovalDay3 =
            -10 and RemovalDay4 = -10 and RemovalDay5 = -10) ;except if all
            removal days are -10 -> pulsed egglaying without removal.
887     []
888     [
889       user-message "A drone egglaying time is not followed by a removal
              day - intended?"
```

```
890       ]
891     ]
892
893     set removal_days (list RemovalDay1 RemovalDay2 RemovalDay3 RemovalDay4
          RemovalDay5) ; Save the removal days in a list.
894     set drone_egglaying_times (list DRONE_EGGLAYING_START
          droneEgglayingTime2 droneEgglayingTime3 droneEgglayingTime4
          droneEgglayingTime5) ; Save the drone egglaying times in a list.
895     set days_between_removals (map - removal_days drone_egglaying_times) ;
          Calculate the time between egg laying & removal days.
896     set days_between_removals remove 0 days_between_removals ; Remove 0s
          (arising from -10 - (-10) if this drone egglaying time and
          corresponding removal day is not used.
897
898     ; Check correct ordering, i.e. droneEgglayingTime2 <
          droneEgglayingTime3 and so on.
899     let drone_egglaying_times_temp remove -10 drone_egglaying_times ;
          Remove the -10 from the list to allow correct sorting.
900     let drone_egglaying_times_ordered sort drone_egglaying_times_temp
901     if drone_egglaying_times_temp != drone_egglaying_times_ordered
902     [
903       set BugAlarm true
904       set BugMessage "BUG Alarm: check that the values of the drone
            egglaying times are ordered according to their parameter name; if
            useDBRdaysAsDroneEgglayingTime = true: check also order of drone
            brood removal days."
905       user-message BugMessage
906     ]
907
908     ; Check correct ordering, i.e. RemovalDay1 < RemovalDay2 and so on.
909     let removal_days_temp remove -10 removal_days ; Remove the -10 from
          the list to allow correct sorting.
910     let removal_days_ordered sort removal_days_temp
911     if removal_days_temp != removal_days_ordered
912     [
913       set BugAlarm true
914       set BugMessage "BUG Alarm: check that the values of the drone brood
            removal days are ordered according to their parameter name."
915       user-message BugMessage
916     ]
917   ]
918
919   set countRemovalDay 0 ; Counter for the removal days during the year.
```

```
920 set cellsDroneBroodFrame 3000 ; Assumption: x cells per drone brood
    frame -> x/2 per side.
921 set maxDroneFrameEgglayingDuration 5 ; Maximum days of one drone
    egglaying event.
922
923 set currentNumDroneEggs 0
924 set countDroneEgglayingDays 0
925
926 set countTotalWorkerEggs 0
927 set countTotalDroneEggs 0
928
929 ; Check that the days for the varroa treatment are in the correct months.
930 if (formidAcidVarroaTreatment = true and ((formidAcidTreatmentDay1-1 !=
    -10 and (formidAcidTreatmentDay1-1 <= 182 or
    formidAcidTreatmentDay1-1 >= 273 )) or  (formidAcidTreatmentDay2-1 !=
    -10 and (formidAcidTreatmentDay2-1 <= 182 or
    formidAcidTreatmentDay2-1 >= 273 ))) )
931 [
932   set BugAlarm true
933   set BugMessage "BUG Alarm: treatment day for formid acid treatment is
        not in July/August/September!"
934   user-message BugMessage
935 ]
936
937 if (oxalicAcidVarroaTreatment = true and (oxalicAcidTreatmentDay-1 !=
    -10 and (oxalicAcidTreatmentDay-1 <= 305 or oxalicAcidTreatmentDay-1
    >= 365 )))
938 [
939   set BugAlarm true
940   set BugMessage "BUG Alarm: treatment day for oxalic acid treatment is
        not in November/December!"
941   user-message BugMessage
942 ]
943
944 set factorWorkers 0.56 ; Boot et al. 1995
```

8. Code addition in the procedure *MitesPhoreticPhaseProc*

```
5335 ; healthy and infected IN-HIVE bees:
5336 set totalInfectedWorkers 0 ;; Changed by I.S. from let to set.
5337 set totalHealthyWorkers 0 ;; Changed by I.S. from let to set.
 ...
5371 ask IHbeeCohorts
5372 [
 ... ...
5382   repeat number_healthy  ; only healthy bees can become newly infected
5383   [
5384     if number > 1 ; Added by I.S.: to avoid 0 ^
          infectedMitesSwitchingHostsInThisCohort.
5385     [
5386       if random-float 1 > (1 - (1 / number)) ^
            infectedMitesSwitchingHostsInThisCohort
5387       ; "number" (i.e. all bees in this cohort) as mites can also jump
            on already infected bees
5388       [
 ...         ...
5394       ]
5395     ]
5396   ]
 ... ...
5436 ] ; end "ask IHbeeCohorts "
```

9. Code for counting mites

Own procedure: *CountMitesInMOProc*; MO = **M**ite**O**rganiser.

```
5621 ; Count the number of infected mites in all miteOrganisers.
5622 ; Called daily in the 'Go' procedure.
5623 to CountMitesInMOProc
5624   set allMitesInMO 0; Reset counter for all mites in brood cells.
5625   set mitesInfectedInMO 0 ; Reset counter for infected mites in brood
         cells.
5626
5627   ask miteOrganisers
5628   [
5629     set allMitesInMO (allMitesInMO + cohortInvadedMitesSum) ; Sum up the
          number of all mites in the miteOrganisers.
5630     set mitesInfectedInMO (mitesInfectedInMO + (cohortInvadedMitesSum *
          (1 - invadedMitesHealthyRate))) ; Sum up the number of infected
          mites in the miteOrganisers.
5631   ]
5632 end
```

10. Code for the sum of mites in the *miteOrganiser*

Own reporter: *sumMitesinMiteOrganiser*.

```
6252 ; ------- Sum Mites in miteOrganiser Reporter --------
6253 to-report sumMitesinMiteOrganiser [current_workerCellListCondensed
        current_droneCellListCondensed]
6254   let counter 0
6255   let cohortInvadedMitesSum_local 0
6256   ; Sum up mites in the current cellListCondensed -> 'x * counter' as
         the CellListCondensed are mite-frequency classes for 0,1,2,...
         mites.
6257   foreach current_workerCellListCondensed
6258   [
6259     x ->
6260     set cohortInvadedMitesSum_local cohortInvadedMitesSum_local + (x *
           counter)
6261     set counter counter + 1
6262   ]
6263   set counter 0
6264   foreach current_droneCellListCondensed
6265   [
6266     x ->
6267     set cohortInvadedMitesSum_local cohortInvadedMitesSum_local + (x *
           counter)
6268     set counter counter + 1
6269   ]
6270   report cohortInvadedMitesSum_local
6271 end
6272 ; ----- Sum Mites in miteOrganiser Reporter END -----
```

11. Code for the egg laying procedure

Within the procedure *NewEggsProc*.

```
1641 ; In case of drone brood removal with a drone brood frame: the egglaying
       of drone brood is not constant with a certain number of eggs per day,
       but takes place in egglaying events.
1642 ifelse DroneBroodRemoval = true
1643 [
1644   ; Start of drone egglaying + maximum maxDroneFrameEgglayingDuration
         days & time periods as specified by the
         droneEgglayingTime-parameters + maximum
         maxDroneFrameEgglayingDuration days each.
1645   ; Based on the HoPoMo curve as before; egg laying until the queen has
         layed at least 'cellsDroneBroodFrame' drone eggs (with a maximum of
         maxDroneFrameEgglayingDuration days).
1646
1647   ; If day is one-of droneEgglayingTime-parameters + max.
         maxDroneFrameEgglayingDuration days each.
1648   ifelse ( ( (day >= DRONE_EGGLAYING_START and day <
         DRONE_EGGLAYING_START + maxDroneFrameEgglayingDuration )
1649   or (day >= droneEgglayingTime2 and day < droneEgglayingTime2 +
         maxDroneFrameEgglayingDuration)
1650   or (day >= droneEgglayingTime3 and day < droneEgglayingTime3 +
         maxDroneFrameEgglayingDuration)
1651   or (day >= droneEgglayingTime4 and day < droneEgglayingTime4 +
         maxDroneFrameEgglayingDuration)
1652   or (day >= droneEgglayingTime5 and day < droneEgglayingTime5 +
         maxDroneFrameEgglayingDuration) )
1653   and countDroneEgglayingDays < maxDroneFrameEgglayingDuration and
         currentNumDroneEggs < cellsDroneBroodFrame )
1654   [
1655     set NewDroneEggs round ELRt ; Majority of eggs on this day are drone
          eggs.
1656     set NewWorkerEggs floor(NewDroneEggs *
          workerEggsProportionOnDroneEgglayingDays); Background worker egg
          laying to avoid a siginficant drop in worker strength.
```

```
1657      set NewDroneEggs NewDroneEggs - NewWorkerEggs ; Reduce number of
              worker eggs, so that num worker eggs + num drone eggs = 100 % of
              that day.
1658
1659      ; Counter for new drone eggs and days the queen took to lay the eggs.
1660      set currentNumDroneEggs currentNumDroneEggs + NewDroneEggs
1661      set countDroneEgglayingDays countDroneEgglayingDays + 1
1662    ]
1663    [ ; Else: other days, that are not around the DBR days -> majority is
           worker eggs.
1664      set NewWorkerEggs round ELRt  ; ROUND! in contrast to HoPoMo
1665
1666      ifelse (Day < DRONE_EGGLAYING_START or Day > DRONE_EGGLAYING_STOP)
1667      [ ; Not in drone season: set drone eggs to 0.
1668      set NewDroneEggs 0
1669      ]
1670      [ ; Else: background drone egg proportion.
1671      set NewDroneEggs floor(NewWorkerEggs *
              droneEggsProportionNoDroneEgglayingDay)
1672      ]
1673
1674      set NewWorkerEggs NewWorkerEggs - NewDroneEggs ; Reduce worker eggs
              by number of drone eggs.
1675    ]
1676  ]
1677  [ ; Else: DroneBroodRemoval = false.
1678    set NewWorkerEggs round ELRt  ; ROUND! in contrast to HoPoMo.
1679
1680    ifelse (Day < DRONE_EGGLAYING_START or Day > DRONE_EGGLAYING_STOP)
1681    [ ; Not in drone season: set drone eggs to 0.
1682    set NewDroneEggs 0
1683    ]
1684    [ ; Else, in drone egg season: specified drone egg proportion on each
           day.
1685    set NewDroneEggs floor(NewWorkerEggs * droneEggsProportionNoDBR)
1686    ]
1687
1688    set NewWorkerEggs NewWorkerEggs - NewDroneEggs ; Reduce worker eggs by
           number of drone eggs.
1689  ]
1690
1691  ; Reset counter -> max. number of days for the current drone
       egglaying-event has passed.
```

```
1692 if ( day = DRONE_EGGLAYING_START + maxDroneFrameEgglayingDuration
1693 or day = droneEgglayingTime2 + maxDroneFrameEgglayingDuration
1694 or day = droneEgglayingTime3 + maxDroneFrameEgglayingDuration
1695 or day = droneEgglayingTime4 + maxDroneFrameEgglayingDuration
1696 or day = droneEgglayingTime5 + maxDroneFrameEgglayingDuration )
1697 [
1698   set droneFrameEgglayingDuration countDroneEgglayingDays ; Save current
         number of drone egglaying days for next drone brood removal day.
1699   ; Reset values for next drone egglaying event.
1700   set currentNumDroneEggs 0
1701   set countDroneEgglayingDays 0
1702 ]
1703
1704 ; Sum up total worker & drone eggs in one year.
1705 set countTotalWorkerEggs countTotalWorkerEggs + NewWorkerEggs
1706 set countTotalDroneEggs countTotalDroneEggs + NewDroneEggs
1707
1708 if day = 365
1709 [
1710   ;;print word "drone eggs % " round((countTotalDroneEggs /
         (countTotalWorkerEggs + countTotalDroneEggs)) * 100) ; Print
         percentage: drone eggs of total eggs in one year.
1711   set countTotalWorkerEggs 0
1712   set countTotalDroneEggs 0
1713 ]
```

12. Code for the drone brood removal procedure

Within the procedure *BeekeepingProc*.

```
6217 ; Drone brood removal (DBR).
6218 if (DroneBroodRemoval = true and (day = RemovalDay1 or day = RemovalDay2
         or day = RemovalDay3 or day = RemovalDay4 or day = RemovalDay5))
6219 [
6220   ; Call the procedure to remove drone brood (Added & outsorced by I.S.).
6221   DroneBroodRemovalProc
6222
6223   ; Call 'CountingProc' -> as already previously called at the end of
         the drone brood removal.
6224   CountingProc
6225
6226   set countRemovalDay countRemovalDay + 1 ; Count for current number of
         removal day (0 -> RemovalDay1, 1 -> RemovalDay2 and so on).
6227   ; Reset the counter after the last removal event in the current year.
6228   if (countRemovalDay = length days_between_removals)
6229   [
6230     set countRemovalDay 0
6231   ]
6232
6233   ; Reset counter for removed mites.
6234   set countMitesRemoved 0
6235 ]
```

Own procedure: *DroneBroodRemovalProc*.

```
6276 ; ------- Drone Brood Removal Procedure --------
6277 to DroneBroodRemovalProc
6278     ; Remove only drone pupae cohorts with specific ages on each
           drone-brood-removal day.
6279   ; Drone pupae are aged 10 to 24.
6280   ; The age of the youngest and oldest drone pupae cohort to be removed
       depends on the days between the start of the last
       droneEgglayingTime and the current removalDay.
```

```
6281
6282    ; Get the number of days between last egg laying and current drone
            brood removal.
6283    let daysBetweenRemoval item countRemovalDay days_between_removals
6284
6285    ; Calculate the possible ages of drone pupae cohorts.
6286    let min_dronePupae_ages map [i -> daysBetweenRemoval - i] (range 0
            maxDroneFrameEgglayingDuration) ; Calculate the possible ages of
            drone pupae cohorts based on daysBetweenRemovals.
6287    ; In 'min_dronePupae_ages': first entry corresponds to
            droneFrameEgglayingDuration =  1, second to
            droneFrameEgglayingDuration =  2 and so on until
            maxDroneFrameEgglayingDuration.
6288    let maxDronePupaeAge daysBetweenRemoval ; Maximum age is the current
            'daysBetweenRemoval'.
6289
6290    ; Determine ages of pupae to be removed from the number of days that
            eggs were layed on the last drone egglaying event.
6291    let itemPosition droneFrameEgglayingDuration - 1 ; Netlogo
            lists/arrays start with 0.
6292    let minDronePupaeAge item itemPosition min_dronePupae_ages ; Determine
            age of youngest larvae cohort that emerged from last drone
            egglaying event.
6293    let countPupaeRemoved 0 ; Count the number of pupae removed.
6294
6295    ask dronePupaeCohorts with [age >= minDronePupaeAge and age <=
            maxDronePupaeAge and number > 0] ; Only pupae cohorts with suitable
            ages get killed, and which are non-empty.
6296    [
6297      ; Kill only a certain proportion of the specific age cohorts.
6298      set current_dronePupaeRemain round(number * (1 -
              propRemoveDronePupae))
6299      set countPupaeRemoved countPupaeRemoved + (number -
              current_dronePupaeRemain); Count removed pupae.
6300      ; Update numbers in this cohort accordingly.
6301      set number round(number * (1 - propRemoveDronePupae))
6302
6303      ; Determine randomly which number (healthy or infected) gets reduced
              first -> the other number is calculated from this one and the
              determined total number of drone pupae 'number'.
6304      ifelse random 1 < 0.5
6305      [
6306        set number_healthy round(number_healthy * (1 -
```

```
          propRemoveDronePupae))
6307      set number_infectedAsPupa number - number_healthy
6308    ]
6309    [
6310      set number_infectedAsPupa round(number_infectedAsPupa * (1 -
          propRemoveDronePupae))
6311      set number_healthy number - number_infectedAsPupa
6312    ]
6313
6314    ask miteOrganiser invadedByMiteOrganiserID ; Ask the to the current
        drone pupae cohort corresponding miteOrganiser.
6315    [
6316      ; Call reporter to reduce mites in all mite classes by the desired
          proportion.
6317      set droneCellListCondensed (UpdateCellListCondensed
          propRemoveDronePupae droneCellListCondensed
          current_dronePupaeRemain)
6318
6319      ; CHECK that the the number of drone pupae remaining is the same
          as the number of cells stored in the MiteOrgansiser.
6320      if current_dronePupaeRemain != sum droneCellListCondensed
6321      [;; Global parameter BugMessage added by I.S. to save message in
          file.
6322        set BugAlarm true
6323        set BugMessage (word "BUG ALARM in DroneBroodRemovalProc: number
            of drone pupae != number of cells in MiteOrganiser"
            current_dronePupaeRemain word " != " droneCellListCondensed)
6324        type (word "number of drone pupae != number of cells in
            MiteOrganiser" current_dronePupaeRemain word " != "
            droneCellListCondensed)
6325      ]
6326    ]
6327  ]
6328 end
6329
6330 ; ------- Drone Brood Removal Procedure END ----
```

Own procedure: *UpdateCellListCondensed*.

```
6333  ; ------- Reduce Mites in MiteOrganiser Reporter  --------
6334  ; Called in DroneBroodRemovalProc.
6335  to-report UpdateCellListCondensed [proportion
        current_cellListCondensed numDronePupae]
```

```
6336  ; For every drone pupae removed in the DroneBroodRemovalProc, one cell
          in the corresponding MiteOrganiser has to be removed.
6337  ; First, the desired proportion that should be removed (e.g. 80%) is
          removed from every mite frequency class.
6338  ; As, due to rounding, more or less cells than desired might be
          removed, it is then checked whether cells then have to be removed
          or re-added.
6339  ; To always make sure
6340
6341  let cellListCondensed current_cellListCondensed ; Local copy of the
          current cell list.
6342
6343  let num_cells_remain 0 ; Local parameter of cells that will remain in
          the current mite-frequency-class.
6344
6345  ; Create a shuffeled list of indices indicating positions in
          cellListCondensed.
6346  let shuffledIndices shuffle n-values length cellListCondensed [i -> i]
6347
6348  foreach shuffledIndices ; Loop through every entry in shuffledIndices
          list.
6349  [
6350    x -> ; Do the following for all entries in shuffledIndices,  while
            'x' being the current position in the cellListCondensed.
6351    set num_cells_remain ((item x cellListCondensed) * (1 - proportion))
            ; As float.
6352    let num_cells_remain_int floor num_cells_remain ; As int.
6353    let diff num_cells_remain - num_cells_remain_int ; Calculate
            difference.
6354
6355    if random 1 < diff [set num_cells_remain num_cells_remain + 1] ;
            Determine whether the difference is large enough to remove one
            more cell.
6356
6357    set cellListCondensed replace-item x cellListCondensed
            num_cells_remain_int ; Update the cells in the current
            mite-frequency-class.
6358  ]
6359
6360  ; If too little or too many cells have been removed due to rounding:
          remove or add them again until the number matches with the number
          of drone pupae in this cohort corresponding to the miteOrganiser.
6361  while [sum cellListCondensed != numDronePupae]
```

```
6362  [
6363    ifelse sum cellListCondensed < numDronePupae
6364    [
6365      let index position (max cellListCondensed) cellListCondensed ;
            Local index to get the position of the  largest value in
            cellListCondensed.
6366      set cellListCondensed replace-item index cellListCondensed ((item
            index cellListCondensed) + 1)
6367    ]
6368    ; If still not enough cells have been removed from the miteOrganiser.
6369    ;i.e.: sum cellListCondensed > numDronePupae
6370    [
6371      let index position (max cellListCondensed) cellListCondensed ;
            Local index to get the position of the  largest value in
            cellListCondensed.
6372      set cellListCondensed replace-item index cellListCondensed ((item
            index cellListCondensed) - 1)
6373    ]
6374  ]
6375  report cellListCondensed
6376  end
6377  ; ------ Reduce Mites in MiteOrganiser Reporter END ------
```

13. Code for the varroa treatment with acaricides

13.1 Varroa treatment with formic acid

Within the procedure *BeekeepingProc*.

```
6017   ; Treatment of the varroa mite with organic acaricide.
6018
6019 ; Formid acid (Ameisensaeure) -> possible on two days between July (Day
       182) & September (Day 273) with specified duration.
6020
6021 ; Formid acid treatment 1.
6022 ; If treatment for formid acid is true and the current day is the
       treatment day or within its duration range.
6023 ; 'formidAcidTreatmentDay1-1' means one day before the treatment should
       be applied.
6024 ; For the value of 'formidAcidTreatmentDay1-1' '>' and not '>=" was
       chosen to allow the number of treatment days to be equal to the
       specified treatment duration.
6025 ; E.g. if formidAcidTreatmentDay1-1 = 214 and
       formidAcidTreatmentDuration1 = 3: with '>=' the treatment duration is
       actually 4 days, with '>' the desired 3.
6026 ; -10 is used as the value when the treatment day should be skipped.
6027 ifelse ((formidAcidVarroaTreatment = true) and
       (formidAcidTreatmentDay1-1 != -10) and (Day >
       formidAcidTreatmentDay1-1) and (Day <= formidAcidTreatmentDay1-1 +
       formidAcidTreatmentDuration1))
6028 [
6029   ask signs with [shape = "x" or shape = "varroamite03"] [ show-turtle]
         ; NetLogo interface command ; Copied from Matthias' code.
6030
6031   set current_formidAcidEfficiencyPhoretic1 formidAcidEfficiencyPhoretic1
6032   set current_formidAcidEfficiencyBroodcells1
         formidAcidEfficiencyBroodcells1
6033
6034   ; Kill phoretic mites.
```

```
6035    set PhoreticMites round(PhoreticMites * (1 -
            current_formidAcidEfficiencyPhoretic1))
6036
6037    ; Kill mites in cells.
6038    ; Remove the desired proportion from every cell in every miteOrganiser.
6039    ask miteOrganisers ; Ask all miteOrganisers in Beehave, regardless of
            their age/associated bee life stage.
6040    [
6041      ; Call outsorced procedure to update the worker/drone
              cellListCondensed.
6042      set workerCellListCondensed (UpdateCellListCondensed_FormidAcid
              current_formidAcidEfficiencyBroodcells1 workerCellListCondensed)
              ; For workers.
6043      set droneCellListCondensed (UpdateCellListCondensed_FormidAcid
              current_formidAcidEfficiencyBroodcells1 droneCellListCondensed) ;
              For drones.
6044
6045      ; Update the total number of mites in this miteOrganiser.
6046      set cohortInvadedMitesSum sumMitesinMiteOrganiser
              workerCellListCondensed droneCellListCondensed
6047    ]
6048
6049    ; If true, kill open brood at the end of the treatment -> negative
            effect of formid acid treatment on bee colony.
6050    if (formidAcidKillOpenBrood1 = true and Day =
            formidAcidTreatmentDay1-1 + formidAcidTreatmentDuration1)
6051    [ ; Copied from Matthias' code.
6052      ask (turtle-set eggCohorts larvaeCohorts) with [ age < PUPATION_AGE
              ] [ set number 0 ]
6053      ask (turtle-set droneEggCohorts droneLarvaeCohorts) with [ age <
              DRONE_PUPATION_AGE ] [ set number 0 ]
6054      ask miteOrganisers with [ age <= 10] ; i.e. those mite organisers,
              connected to dying larvae cohorts
6055      [
6056        if age < 10 ; for workers: age 10 brood is already capped, i.e.
                not affected!
6057          [ set workerCellListCondensed n-values
                  (MAX_INVADED_MITES_WORKERCELL + 1) [ 0 ]]
6058        set droneCellListCondensed n-values (MAX_INVADED_MITES_DRONECELL +
              1) [ 0 ]
6059        let memoInvadedW invadedWorkerCohortID
6060        let memoInvadedD invadedDroneCohortID
```

```
6061        if any? turtles with [ who = memoInvadedW ] [ set
                workerCellListCondensed replace-item 0 workerCellListCondensed
                [number] of turtle invadedWorkerCohortID ]
6062        if any? turtles with [ who = memoInvadedD ] [ set
                droneCellListCondensed replace-item 0 droneCellListCondensed
                [number] of turtle invadedDroneCohortID ]
6063      ]
6064    ]
6065  ]
6066  [
6067    ask signs with [shape = "x" or shape = "varroamite03"] [ hide-turtle]
          ; Copied from Matthias' code.
6068  ]
6069
6070
6071  ; Formid acid treatment 2.
6072  ; If treatment for formid acid is true and the current day is the
        treatment day or within its duration range.
6073  ; For the value of 'formidAcidTreatmentDay2-1' '>' and not '>=" was
        chosen for reasons specified above for the first formid acid
        treatment.
6074  ; -10 is used as the value when the treatment day should be skipped ->
        for example if one only wants one treatment with formid acid put -10
        in 'formidAcidTreatmentDay2-1' on the graphical interface.
6075  ifelse ((formidAcidVarroaTreatment = true) and
        (formidAcidTreatmentDay2-1 != -10) and (Day >
        formidAcidTreatmentDay2-1) and (Day <= formidAcidTreatmentDay2-1 +
        formidAcidTreatmentDuration2))
6076  [
6077    ask signs with [shape = "x" or shape = "varroamite03"] [ show-turtle]
          ; NetLogo interface command ; Copied from Matthias' code.
6078
6079    set current_formidAcidEfficiencyPhoretic2 formidAcidEfficiencyPhoretic2
6080    set current_formidAcidEfficiencyBroodcells2
          formidAcidEfficiencyBroodcells2
6081
6082    ; Kill phoretic mites.
6083    set PhoreticMites round(PhoreticMites * (1 -
          current_formidAcidEfficiencyPhoretic2))
6084
6085    ; Kill mites in cells.
6086    ; Remove the desired proportion (given by 'formidAcidEfficiency2')
          from every cell in every miteOrganiser.
```

```
6087    ask miteOrganisers ; Ask all miteOrganisers in Beehave, regardless of
            their age/associated bee life stage.
6088    [
6089      ; Call outsorced procedure to update the worker/drone
            cellListCondensed.
6090      set workerCellListCondensed (UpdateCellListCondensed_FormidAcid
            current_formidAcidEfficiencyBroodcells2 workerCellListCondensed)
            ; For workers.
6091      set droneCellListCondensed (UpdateCellListCondensed_FormidAcid
            current_formidAcidEfficiencyBroodcells2 droneCellListCondensed) ;
            For drones.
6092
6093      ; Update the total number of mites in this miteOrganiser.
6094      set cohortInvadedMitesSum sumMitesinMiteOrganiser
            workerCellListCondensed droneCellListCondensed
6095    ]
6096
6097    ; If true, kill open brood at the end of the treatment -> negative
        effect of formid acid treatment.
6098    if (formidAcidKillOpenBrood2 = true and Day =
            formidAcidTreatmentDay2-1 + formidAcidTreatmentDuration2)
6099    [ ; Copied from Matthias' code.
6100      ask (turtle-set eggCohorts larvaeCohorts) with [ age < PUPATION_AGE
            ] [ set number 0 ]
6101      ask (turtle-set droneEggCohorts droneLarvaeCohorts) with [ age <
            DRONE_PUPATION_AGE ] [ set number 0 ]
6102      ask miteOrganisers with [ age <= 10 ] ; i.e. those mite organisers,
            connected to dying larvae cohorts
6103      [
6104        if age < 10 ; for workers: age 10 brood is already capped, i.e.
              not affected!
6105          [ set workerCellListCondensed n-values
                (MAX_INVADED_MITES_WORKERCELL + 1) [ 0 ]]
6106        set droneCellListCondensed n-values (MAX_INVADED_MITES_DRONECELL +
              1) [ 0 ]
6107        let memoInvadedW invadedWorkerCohortID
6108        let memoInvadedD invadedDroneCohortID
6109        if any? turtles with [ who = memoInvadedW ] [ set
              workerCellListCondensed replace-item 0 workerCellListCondensed
              [number] of turtle invadedWorkerCohortID ]
6110        if any? turtles with [ who = memoInvadedD ] [ set
              droneCellListCondensed replace-item 0 droneCellListCondensed
              [number] of turtle invadedDroneCohortID ]
```

```
    ]
  ]
]
[
  ask signs with [shape = "x" or shape = "varroamite03"] [ hide-turtle]
     ; Copied from Matthias' code.
]
```

Own reporter: *UpdateCellListCondensed_FormidAcid*.

```
; ------- Varroa Treatment Reporter: Formid Acid --------
; Called in BeekeepingProc for formid acid treatment.
to-report UpdateCellListCondensed_FormidAcid [formidAcidEfficiency
   current_cellListCondensed]
  ; It is assumed that once the treatment gets into a brood cell all
     mites in this cell die -> the cell is always shifted to the
     no-mites frequency-class.
  ; Remove the desired proportion from every entry in the current
     cellListCondensed.

  let cellListCondensed current_cellListCondensed ; Local copy of the
     current cell list.

  let num_cells_remain 0 ; Local number of cells that will remain in the
     current mite-frequency-class.
  let index 0 ;

  foreach cellListCondensed ; Loop through every entry in the
     cellListCondensed.
  [
    x -> ; Do the following for all entries in cellListCondensed,  while
       'x' being the current cellListCondensed entry.

    if index > 0 ; Skip the first entry in the cellListCondensed, as
       this corresponds to the number of no-mite-cells.
    [
      set num_cells_remain (x * (1 - formidAcidEfficiency)) ; As float.
      let num_cells_remain_int floor num_cells_remain ; As int.
      let diff num_cells_remain - num_cells_remain_int ; Calculate
         difference.

      if random 1 < diff [set num_cells_remain num_cells_remain + 1] ;
         Determine whether the difference is large enough to remove one
```

```
          more cell.
6403
6404      set cellListCondensed replace-item index cellListCondensed
              num_cells_remain_int ; Update the cells in the current
              mite-frequency-class.
6405      set cellListCondensed replace-item 0 cellListCondensed ((x -
              num_cells_remain_int) + item 0 cellListCondensed)  ; Move the
              removed cells to the first mite-frequency-class, i.e. no mites.
6406    ]
6407    set index index + 1
6408  ]
6409
6410  report cellListCondensed
6411 end
6412 ; ------ Varroa Treatment Reporter END ------
```

13.2 Varroa treatment for oxalic acid

Within the procedure *BeekeepingProc*.

```
6114 ; Oxalic acid -> possible in November (earliest Day 305) and December
       (latest Day 365).
6115 ; For the value of 'oxalicAcidTreatmentDay-1' '>' and not '>=" was
       chosen for reasons specified above for the first formid acid
       treatment.
6116 ; -10 is used as the value when the treatment day should be skipped.
6117 ifelse ((oxalicAcidvarroaTreatment = true) and (oxalicAcidTreatmentDay-1
       != -10) and (Day > oxalicAcidTreatmentDay-1) and (Day <=
       oxalicAcidTreatmentDay-1 + oxalicAcidTreatmentDuration))
6118 [
6119   ask signs with [shape = "x" or shape = "varroamite03"] [ show-turtle]
         ; Copied from Matthias' code.
6120
6121   ; Check that the bee colony is brood free and that the latest
         egglaying was 3 to 4 weeks before the treatment day -> requirement
         for oxalic acid.
6122   let totalBrood (totalEggs + TotalLarvae + TotalPupae + TotalDroneEggs
         + TotalDroneLarvae + TotalDronePupae)
6123   if (totalBrood > 0 )
6124   [
6125     output-show (word ticks)
6126     type "TotalEggs: "
6127     type TotalEggs
6128     type " TotalLarvae: "
6129     type TotalLarvae
6130     type " TotalPupae: "
6131     type TotalPupae
6132     type " TotalDroneEggs: "
6133     type TotalDroneEggs
6134     type " TotalDroneLarvae: "
6135     print TotalDroneLarvae
6136     type " TotalDronePupae: "
6137     print TotalDronePupae
6138     user-message "BUG Alarm: brood still present on oxalic acid
           treatment day!"
6139   ]
6140
6141   ; Kill ONLY phoretic mites.
```

```
6142    set PhoreticMites round(PhoreticMites * (1 -
            current_oxalicAcidEfficiency))
6143
6144  ]
6145  [
6146    ask signs with [shape = "x" or shape = "varroamite03"] [ hide-turtle]
            ; Copied from Matthias' code.
6147  ]
```

14. Code for running a burn-in phase in BEEHAVE

Running a burn-in phase in BEEHAVE with mites and varroa control measures. A text file has then to be created with the parameters according to what is read in in the procedures *ReadBeeMappFileProc* & *BeeMappCorrectionProc*. In the Behaviour-Space experiment the following parameters have to be specified accordingly:
ReadBeeMappFile = true & set *BeeMapp_ FILE*

```
7155 to ReadBeeMappFileProc
7156 ; reads colony data in from file, created by the BeeMapp app
7157
7158   ifelse ( file-exists? BeeMapp_FILE )
7159   [
7160     set AllBeeMappCorrectionsList []
7161     file-open BeeMapp_FILE
7162     let dustbin file-read-line
7163     ; first line of input file with headings is read - but not used for
          anything
7164
7165     while [ not file-at-end? ]
7166     [
7167       set AllBeeMappCorrectionsList sentence AllBeeMappCorrectionsList ;
            x columns in BeeMapp input file:
7168       (list (list ; repeat nColumns [ file-read ]
7169       file-read file-read file-read file-read file-read file-read )) ;;
            Changed by I.S.
7170     ]
7171     set AssessmentNumber 0
7172     ;(list (list file-read-line ))]
7173
7174     file-close
7175   ] ; end "ifelse"
7176   [
7177     user-message "There is no such BeeMapp_FILE in current directory!"
7178   ]
7179 end
```

```
7183 to BeeMappCorrectionProc ; ***NEW FOR BEEHAVE_BEEMAPP2015***
7184
7185   let nextBeeMappCorrectionList item AssessmentNumber
           AllBeeMappCorrectionsList
7186
7187   if ticks = item 0 nextBeeMappCorrectionList  ; if day = date of colony
           next colony assessment ;; Changed by I.S.: from item 1 to item 0.
7188   [
7189
7190     ;############ Added by I.S. ############
7191
7192     ; Change mite numbers.
7193     ; Healthy mites.
7194     set N_INITIAL_MITES_HEALTHY item 1 nextBeeMappCorrectionList
7195     ; Infected mites.
7196     set N_INITIAL_MITES_INFECTED item 2 nextBeeMappCorrectionList
7197
7198     ; Change other mite numbers accordingly.
7199     set PhoreticMites  N_INITIAL_MITES_HEALTHY + N_INITIAL_MITES_INFECTED
7200     set TotalMites  PhoreticMites
7201     if (N_INITIAL_MITES_HEALTHY + N_INITIAL_MITES_INFECTED) > 0
7202     [
7203       set PhoreticMitesHealthyRate N_INITIAL_MITES_HEALTHY /
              (N_INITIAL_MITES_HEALTHY + N_INITIAL_MITES_INFECTED)
7204     ]
7205
7206     ; Drone brood removal (DBR).
7207     ifelse item 3 nextBeeMappCorrectionList = 1
7208     [ ; DBR true
7209       set DroneBroodRemoval true
7210     ]
7211     [ ; DBR false
7212       set DroneBroodRemoval false
7213     ]
7214
7215     ; Mite reinfestation.
7216     ifelse item 4 nextBeeMappCorrectionList = 1
7217     [
7218       set AllowReinfestation true
7219     ]
7220     [
7221       set AllowReinfestation false
7222     ]
```

```
7223
7224    ; Formic acid treatment.
7225    ifelse item 5 nextBeeMappCorrectionList = 1
7226    [
7227      set formicAcidVarroaTreatment true
7228    ]
7229    [
7230      set formicAcidVarroaTreatment false
7231    ]
7232
7233
7234    ;############ Addition END ############
```

Bibliography

Allen, M. D. (1958). "Drone Brood in Honey Bee Colonies". In: *Journal of Economic Entomology* 51 (1), pp. 46–48.

Allen, M. D. (1963). "Drone Production in Honey-Bee Colonies (Apis Mellifera L.)" In: *Nature* 199, pp. 789–790.

Allen, M. D. (1965). "The Effect of a Plentiful Supply of Drone Comb on Colonies of Honeybees". In: *Journal of Apicultural Research* 4 (2), pp. 109–119.

Becher, M. A., V. Grimm, P. Thorbek, J. Horn, P. J. Kennedy, and J. L. Osborne (2014). "BEEHAVE: a systems model of honeybee colony dynamics and foraging to explore multifactorial causes of colony failure". In: *Journal of Applied Ecology* 51 (2). Ed. by Eric Morgan, pp. 470–482.

Büchler, R. (1996). "Erzeugung von Naturbau und mögliche Auswirkungen auf die Volksentwicklung". In: *Allgemeine Deutsche Imkerzeitung (ADZI)* 30 (2), pp. 20–23.

Calderón, R A, R A Ortiz, H G Arce, J W van Veen, and J Quan (2000). "Effectiveness of Formic Acid on Varroa Mortality in Capped Brood Cells of Africanized Honey Bees". In: *Journal of Apicultural Research* 39.3-4, pp. 177–179.

Calis, J. N. M., I. Fries, and S. C. Ryrie (1999). "Population modelling of Varroa jacobsoni Oud". In: *Apidologie* 30 (2-3), pp. 111–124.

Charrière, J.-D., A. Imdorf, B. Bachofen, and A. Tschan (2003). "The removal of capped drone brood: an effective means of reducing the infestation of varroa in honey bee colonies". In: *Bee World* 84 (3), pp. 117–124.

Dadant, C. C. (1975). *The hive and the honey bee.* Dadant & Sons, Hamilton. Ill. Chap. Beekeeping equipment. Pp. 303–328. 740 pp.

DeGrandi-Hoffman, G., S. A. Roth, G. L. Loper, and E. H. Erickson (1989). "BEEPOP: A honeybee population dynamics simulation model". In: *Ecological Modelling* 45 (2), pp. 133–150.

Dettli, M. (2009). *Vergleich Naturbau-Mittelwand. Aus dem Schlussbericht zu den beiden Forschungsprojekten.* M. Dettli Bienenforschung und Wanderimkerei (www.summ-

summ.ch). URL: https://www.summ-summ.ch/app/download/10847886295/naturbaupdf.pdf?t=1483352685. Accessed 26 May 2021.

Dettli, M. (2019). "Naturbauforschung (2. Teil)". In: *Schweizerische Bienen-Zeitung* (1), pp. 14–17.

Free, J. B. (1977). *The Social Organisation of Honeybees*. E. Arnold, London. 68 pp.

Grimm, V., U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. Heinz, and G. Huse (2006). "A standard protocol for describing individual-based and agent-based models". In: *Ecological modelling* 198, pp. 115–126.

Grimm, V., S. Railsback, C. Vincenot, U. Berger, C. Gallagher, D. Deangelis, B. Edmonds, J. Ge, J. Giske, J. Groeneveld, A. Johnston, A. Milles, J. Nabe-Nielsen, J. Polhill, V. Radchuk, M.-S. Rohwäder, R. Stillman, J. Thiele, and D. Ayllón (2020). "The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism". In: *Journal of Artificial Societies and Social Simulation* 23 (2).

Imdorf, A., K. Ruoff, and P. Fluri (2008). "Volksentwicklung bei der Honigbiene." In: *ALP forum* 68. URL: https://ira.agroscope.ch/de-CH/publication/18837. Accessed 31 May 2021.

Jay, S. C. (1974). "Seasonal Development of Honeybee Colonies Started From Package Bees". In: *Journal of Apicultural Research* 13 (2), pp. 149–152.

Liebig, G. (1997). "Volksentwicklung auf Naturwabenbau". In: *Deutsches Bienen Journal* 5 (5-6), pp. 12–13/16.

Mesquida, J. (1976). "Nouvelles observations sur l'élevage des mâles dans les colonies d'abeilles (Apis Mellifica L.)" In: *Apidologie* 7 (4), pp. 307–330.

Pratt, S. C. (1998). "Decentralized control of drone comb construction in honey bee colonies". In: *Behavioral Ecology and Sociobiology* 42 (3), pp. 193–205.

Rowland, C. M. and A. R. McLellan (1987). "Seasonal changes of drone numbers in a colony of the honeybee, Apis mellifera". In: *Ecological Modelling* 37 (3-4), pp. 155–166.

Schmickl, T. and K. Crailsheim (2007). "HoPoMo: A model of honeybee intracolonial population dynamics and resource management". In: *Ecological Modelling* 204 (1-2), pp. 219–245.

Seeley, T. D. (2002). "The effect of drone comb on a honey bee colony's production of honey". In: *Apidologie* 33 (1), pp. 75–86.

Seeley, T. D. and R. Morse (1976). "The nest of the honey bee (Apis mellifera L.)" In: *Insectes Sociaux* 23, pp. 495–512.

Steube, Xenia, Patricia Beinert, and Wolfgang H. Kirchner (2021). "Efficacy and temperature dependence of 60% and 85% formic acid treatment against Varroa destructor". In: *Apidologie* 52.3, pp. 720–729.

Sumpter, D. J. T. and S. J. Martin (2004). "The dynamics of virus epidemics in Varroa-infested honey bee colonies". In: *Journal of Animal Ecology* 73 (1), pp. 51–63.

Torres, D. J. and N. A. Torres (2020). "Modeling the Influence of Mites on Honey Bee Populations". In: *Veterinary Sciences* 7 (3), p. 139.

Wilkinson, D. and G. C. Smith (2002). "A model of the mite parasite, Varroa destructor, on honeybees (Apis mellifera) to investigate parameters important to mite population growth". In: *Ecological Modelling* 148 (3), pp. 263–275.