

MBO: Marriage in Honey Bees Optimization A Haplometrosis Polygynous Swarming Approach

Hussein A. Abbass

School of Computer Science,
University of New South Wales,
University College, ADFA Campus,
Northcott Drive, Canberra ACT, 2600, Australia,
h.abbass@adfa.edu.au

Abstract-

Honey-bees are one of the most well studied social insects. They exhibit many features that distinguish their use as models for intelligent behavior. These features include division of labor, communication on the individual and group level, and cooperative behavior. In this paper, we present a unified model for the marriage in honey-bees within an optimization context. The model simulates the evolution of honey-bees starting with a solitary colony (single queen without a family) to the emergence of an eusocial colony (one or more queens with a family). From optimization point of view, the model is a committee machine approach where we evolve solutions using a committee of heuristics. The model is applied to a fifty propositional satisfiability problems (SAT) with 50 variables and 215 constraints to guarantee that the problems are centered on the phase transition of 3-SAT. Our aim in this paper is to analyze the behavior of the algorithm using biological concepts (number of queens, spermatheca size, and number of broods) rather than trying to improve the performance of the algorithm while losing the underlying biological essence. Notwithstanding, the algorithm outperformed WalkSAT, one of the state-of-the-art algorithms for SAT.

1 Introduction

Swarm intelligence (Bonabeau et al. 1999) is a new area of research concerned with modelling the behavior of social insects, such as ants and bees, and using these models for problem solving and search. Ant colony optimization is one of the most successful models of swarm-based approaches for optimization, where the search algorithm is inspired by the behavior of real ants (Dorigo and Gambardella 1997; Dorigo and Caro 1999; Dorigo et al. 2000). Honey-bees are also used to model agent-based systems (Perez-Urbe and Hirsbrunner 2000). In this paper, we model the marriage behavior of honey-bees and use this model to inspire an optimization

search algorithm, which we call *optimization with marriage in honey-bees* (MBO).

MBO is inspired by the phylogenetic of sociality in Hymenoptera, such as bees, ants, and wasps, and the mating process in honey-bees. Eusocial insects are characterized by three main features: cooperation among adults in brood care and nest construction, overlapping of at least two generations, and reproductive division of labor. Insects without these attributes are termed solitary and those lack one or two of these attributes are termed presocial (Dietz 1986). The evolutionary steps of eusociality were driven by the continuous series of nests, starting with entirely solitary to high eusociality. Two sequences can be distinguished in establishing this series of nests: subsocial (familial or altruistic cooperation) and parasocial (mutualistic cooperation). In the literature of behavioral genetics, the former series is more common than the latter and this will be the assumed model in this paper.

A colony can be founded in two different methods (Dietz 1986). The first method is called *independent founding*, where a colony starts with one or more reproductive females that construct the nest, produce eggs, and feed the larvae. The first brood is reared alone until they emerge and take over the work of the colony. Subsequently, division of labor starts to take place, where the queen specializes in egg laying and the workers in brood care. The second method is called *swarming* where the colony is founded by one or more queens in addition to a group of workers from the original colony. Division of labor commences from the beginning where queens specialize in egg laying and workers in brood care. The colony is *haplometrosis* if it is founded by a single queen; otherwise *pleometrosis*. If the colony contains one queen during its life-cycle, it is called a *monogynous* colony; otherwise a *polygynous* colony.

In this paper, we take a swarming eusocial approach assuming a haplometrosis polygynous colony. The paper is organized as follows: Section 2 introduces the *propositional satisfiability* (SAT) problem along with some background materials. In Section 3, we

discuss the marriage process in honey-bees and formulate a general version of the MBO algorithm. The MBO algorithm applied to SAT is then discussed in Section 4 followed by experimental setup and results in Section 5. The paper is then concluded with Section 6.

2 The propositional satisfiability problem

A general constraint satisfaction problem (CSP) is the problem of finding an assignment to a set of variables that satisfies a set of constraints over those variables. To formally define a CSP, we introduce the following notations. " S_V^v " represents an ordered instantiation of a set of variables V by substituting corresponding values v from their domain $D(V)$, " \exists " for such that, " \neg " for not derive, and finally " \perp " for falsification. We can define a CSP problem as follows: A set of constraints C over the set of variables $V \in D(V)$ - that is $C \subset V \times V$ - is satisfiable iff $\exists v \in D(V), \exists S_V^v \wedge C \not\vdash \perp$.

In SAT, the domain of each variable is either true or false, or equivalently 1 or 0 (i.e. $V \in \{0, 1\}$). Although SAT is a special case of CSP, any CSP can be mapped to SAT (Hoos 1999b; Hoos 1999a). Many problems in planning and scheduling can be represented using SAT; therefore solving SAT is a very attractive research area. However, it is known that SAT is intractable (Papadimitriou and Steiglitz 1982).

In the literature, there are two main streams of techniques for solving SAT: complete and incomplete techniques (Hoos and Stützle 2000). The former use an exhaustive search approach and guarantee a solution if one exists. However, a complete technique (such as Davis-Putnam) can only handle small problems (Zhang and Stickel 1994), after which the time needed to solve the problem gets beyond any computer capabilities. The latter, although they do not guarantee convergence to a solution, they are fast and more suitable for large problems. Therefore, incomplete techniques become more attractive, especially with problems in planning which include thousands of variables (Selman and Kautz 1993b; Selman et al. 1994a).

The easiness/hardness of solving SAT depends on a phenomena known as "phase transition" (Cook and Mitchell 1997). Problems before the phase transition are easy to solve and those after the phase transition are mostly unsatisfiable. Hard SAT problems exist around the phase transition region. A phase transition is defined by the ratio between the number of clauses (constraints), l , and the number of literals (variables), n . A Markov-bound estimates the phase transition of a problem with

k literals per clause as (Gent and Walsh 1994)

$$\frac{l}{n} \approx -\frac{\ln(2)}{\ln(1 - \frac{1}{2^k})}$$

However, for 3-SAT, the phase transition was experimentally found to be 4.3 (Cook and Mitchell 1997).

3 Marriage in honey-bees

The behavior of Honey-bees is the product of their genetic potentiality, ecological and physiological environments, the social conditions of the colony, and various prior and ongoing interactions among these three (Rinderer and Collins 1986). Each bee performs sequences of actions which unfold according to genetic, environmental, and social regulation. The outcome of each action itself becomes a portion of the environment and greatly influences the subsequent actions of both a single bee and her hive mates. The marriage process represents one type of action that was difficult to study because the queens mate during their mating-flight far from the nest. Consequently, the mating process was hard to observe. Before summarizing the mating-flight in honey-bees, we first present the structure of a normal honey-bees colony.

3.1 Colony structure

Each normal honey-bees colony consists of the queen(s), drones, workers, and broods. Queens represent the main reproductive individuals in some types of honey-bees - such as the European *Apis Mellifera* - and specialize in eggs laying (Laidlaw and Page 1986). Drones are the sires or fathers of the colony. They are haploid and act to amplify their mothers' genome without alteration of their genetic composition except through mutation. Therefore, drones are considered as agents that propagate one of their mother's gametes and function to enable females to act genetically as males.

Workers specialize in brood care and sometimes lay eggs. Broods arise either from fertilized or unfertilized eggs. The former represent potential queens or workers, whereas the latter represent prospective drones.

3.2 The mating-flight

A mating-flight starts with a dance performed by the queens who then start a mating flight during which the drones follow the queens and mate with them in the air. In a typical mating-flight, each queen mates with seven to twenty drones (Adams et al. 1972). In each mating, sperm reaches the spermatheca and accumulates there to form the genetic pool of the

colony. Each time a queen lays fertilized eggs, she retrieves at random a mixture of the sperms accumulated in the spermatheca (Page et al. 1984) to fertilize the egg.

3.3 The artificial analogue model

The mating-flight can be visualized as a set of transitions in a state-space (the environment) where the queen moves between the different states in the space in some speed and mate with the drone encountered at each state probabilistically. The queen is initialized with some energy-content at the start of the flight and returns to her nest when the energy is within some threshold from zero or when her spermatheca is full.

In this paper, we will restrict the functionality of a set of workers to brood cares and therefore, each worker will be represented as a heuristic which acts to improve (take care of) a set of broods. A drone mates with a queen probabilistically using the following equation

$$\text{prob}(Q, D) = e^{\frac{-\text{difference}}{\text{speed}}} \quad (1)$$

where, $\text{prob}(Q, D)$ is the probability of adding the sperm of drone D to the spermatheca of queen Q ; that is, the probability of a successful mating, *difference* is the absolute difference between the fitness of D and Q , and *speed* is the speed of the queen Q . It is clear that this function acts as an annealing function, where the probability of mating is high when either the queen is still in the start of her mating-flight and therefore her speed is high, or when the fitness of the drone is as good as the queen's. After each transition in the space, the queen's speed and energy are reduced using the following equations

$$\text{speed}(t+1) = \alpha * \text{speed}(t)$$

$$\text{energy}(t+1) = \text{energy}(t) - \text{step}$$

where α is a factor $\in]0, 1[$ and is taken to be 0.9 in our implementation, and *step* is the amount of energy reduction after each transition.

In Figure 1, a generic MBO algorithm is presented. The algorithm starts with initializing the set of workers with some heuristics. The genotype of each queen is then initialized at random. A set of mating-flights is then undertaken where the energy, speed, and position of each queen are initialized with some value at random. Afterwards, each queen starts moving between states in the space according to her speed and mate with the drone it encounters at each state using the previously discussed function in Equation 1. If a drone is successfully mated with the queen, its sperm is added to the queen's spermatheca (ie a list of partial solutions). After the mating flight is over for all queens, they return

to the nest and start breeding by randomly selecting a sperm from the queen's spermatheca followed by crossover with the queen's genome that complements the chosen sperm and forms a brood. Mutation then acts on the brood; therefore, if the same sperm is used once more to generate a brood, the resultant brood will be different because of mutation. This process is followed by applying workers that are used to raise (improve) the broods. The fitness of each worker is then updated based on the amount of improvement achieved by the worker to the drone. The least-fitted queen is then replaced with the fittest brood until none of the broods is fitter than any of the queens. The remaining broods are then killed and a new mating flight starts. In reality, the female broods become workers or queens and the diploid males are killed. However, since a worker in our algorithm represents a heuristic without a genome, all remaining broods are assumed to be diploid males for simplification. Also, to avoid inbreeding (since the number of queens is usually small), drones are generated at random; therefore they are assumed to be unrelated to the queens.

```

initialize workers
randomly generate the queens
apply local search to get a good queen
for a pre-defined maximum number of mating-flights
  for each queen in the queen list
    initialize energy, speed and position
    the queen moves between states
      and probabilistically chooses drones
    if a drone is selected, then
      add its sperm to the queen's spermatheca
    end if
    update the queen's internal energy and speed
  end for each
generate broods by crossover and mutation
use workers to improve the broods
update workers' fitness
while the best brood is better than the worst queen
  replace the least-fittest queen with the best brood
  remove the best brood from the brood list
end while
end for

```

Figure 1: *Optimization by marriage in honey-bees*

4 MBO for SAT

In this section, the application of the MBO algorithm to the propositional satisfiability problem is presented in two stages. First, the representation of a colony and a solution along with the means of calculating the fitness of the individuals are presented in Section 4.1. Second,

the algorithm that is applied to SAT is introduced in Section 4.2.

4.1 Representation

A genotype of an individual is represented using an array of binary values and length equal to the number of literals in the problem, where each cell corresponds to a literal. If the value assigned to a cell is 1, this indicates that the corresponding literal is true; otherwise it is false.

A drone is represented using a genotype and a genotype marker. Since all drones are haploid, a genotype marker is used to randomly mark half of the genes in the genotype and leaves the other half unmarked; the unmarked genes are the ones that form a sperm.

Each queen has a genotype, speed, energy, and spermatheca (a repository of drones' sperm). A queen's speed and energy are initialized before each mating-flight at random in the range $[0.5, 1]$. When a mating takes place between a queen and one of the drones' sperm stored in her spermatheca, a brood is constructed by copying the unmarked genes in the drones' sperm into the brood and completing the rest of the genes from the queen's genome. A brood has only one genotype. The fitness of the genotype is the ratio between the number of clauses satisfied by the assignment to the total number of clauses in the problem.

Workers represent a set of different heuristics. Each heuristic has a fitness value that represents the amount of improvement in a brood's genotype as a result of applying the heuristic to that brood. Five heuristics are used in this paper; these are WalkSAT, random walk, random flip, random new, and 1-point crossover. For the description of WalkSAT and random walk, the reader may refer to (Selman et al. 1992; Selman and Kautz 1993a; Selman et al. 1994b). The heuristic *random flip* randomly chooses a variable and changes its value to its complement (*ie.* changing false to true or true to false). The heuristic *random new* replaces the brood's genotype with a new randomly generated genotype. The 1-point crossover heuristic, crossovers the brood's genotype with a randomly generated genotype. The crossover point is chosen at random.

4.2 The algorithm

The complete MBO algorithm is presented in Figure 2. The algorithm starts with three user-defined parameters and one pre-defined parameter. The pre-defined parameter is the number of workers, representing the number of heuristics encoded in the program. However,

the pre-defined parameter may be used as a user parameter to alter the number of active heuristics if required; that is, the user may choose the first W heuristics, where W is less than or equal to the total number of heuristics encoded in the program. The three user-defined parameters are the number of queens, the queen's spermatheca size representing the maximum number of matings per queen in a single mating-flight, and the number of broods that will be born by all queens. The energy and speed of each queen at the start of each mating-flight are initialized at random.

A set of queens is then initialized at random. Then a randomly selected heuristic is used to improve the genotype of each queen, therefore preserving the assumption that a queen is usually a good bee. A number of mating-flights is then undertaken. In each mating-flight, all queens fly based on the energy and speed of each, where both energy and speed are generated at random for each queen before each mating flight commences. At the start of a mating-flight, a drone is generated at random and the queen is positioned over that drone. The transition made by the queen in the space is based on her speed which represents the probability of flipping each bit in the drone's genome. Therefore, at the start of a mating-flight, the speed is usually high and the queen makes very large steps in the space. While the energy of the queen decreases, the speed decreases and as a result the neighborhood covered by the queen decreases. At each step made by the queen in the space, the queen is mated with the drone encountered at that step using the probabilistic rule in Equation 1. If the mating is successful (*ie.* the drone passes the probabilistic decision rule), the drone's sperm is stored within the queen's spermatheca. We may notice here that each time a drone is generated, half of his genes are marked at random since each drone is haploid by definition. Therefore, the genes that will be transmitted to the broods are fixed for each drone.

When all queens complete their mating-flight, they start breeding. For a required B broods, a queen is selected in proportion to her fitness and mated with a randomly selected sperm from her spermatheca. A worker is chosen in proportion to its fitness to improve the resultant brood. After all broods are being generated, they are sorted according to their fitness. The best brood replaces the worst queen until there is no brood that is better than any of the queens. All broods are then killed and a new mating-flight is undertaken until all mating-flights are completed or an assignment that satisfies all clauses is encountered.

```

Define  $Q, W$ , and  $B$  to be the number of queens, workers,
and broods respectively
Define  $M$  to be the spermatheca size
Define energy, and speed to be the queen's energy and
speed respectively
Initialize each worker with a unique heuristic
Initialize each queen's genotype at random
select a worker at random and apply it to improve the
queen's genotype
while the stopping criteria are not satisfied
  for queen = 1 to  $Q$ 
    initialize energy, speed and position
    initialize step =  $\frac{0.5 \times \text{energy}}{M}$ 
    generate a drone using position
    while energy > 0
      evaluate the genotype of the drone
      if the drone passes the probabilistic
        condition, then
        if the queen's spermatheca is not
          full, then
            add its spermatozoa to the
              queen's spermatheca
          end if
        end if
      energy = energy - step
      speed =  $0.9 \times \text{speed}$ 
      with a probability of speed, flip each bit in the
        drone's genotype
    end while
  end for
  for brood = 1 to  $B$ 
    select a queen in proportion to her fitness
    select a sperm from the queen's spermatheca
      at random
    generate a brood by crossovering the queen's genome
      with the selected sperm
    mutate the generated brood's genotype
    select a worker in proportion to its fitness
    use the selected worker to improve the
      drone's genotype
    update the worker's fitness based on the amount of
      drone's improvement
  end for
  while the best brood is better than the worst queen
    replace the least-fittest queen with the best brood
    remove the best brood from the brood list
  end while
end while

```

Figure 2: *MBO for SAT*

5 Experiments

5.1 Experimental setup

The goal of this set of experiments is to test the behavior of the algorithm in relation to the number of queens, spermatheca size, and number of mating-flights.

A honey-bees colony usually contains a single queen, although sometimes a couple of queens exist (Rinderer and Collins 1986). Therefore, we experimented with fewer number of queens of 1, 2, 3, 4, and 5. Since as mentioned in Section 3 the number of matings per flight usually ranges between seven and twenty (Adams et al. 1972), we experimented with three values for this parameter of 7, 14, and 21 representing the spermatheca size. The number of flights is somehow flexible and limited with the life of the queens. To have a fair comparison among our experiments, we needed to guarantee that the number of broods all over a single run and under any experimental setup is equal. Therefore, we experimented with 60, 30, 20, 15, and 12 mating flights, where the corresponding number of broods was 20, 40, 60, 80, and 100 respectively. For example, 60 mating flights times 20 broods will result in 1200 trial solutions, where the workers will work to improve each of these solutions. It is taken that the worker will try to improve each brood 100 times. This parameter is assumed fixed, although for future work, we are interested in examining the effect of this parameter on the problem.

Fifty different 3-SAT problems were uniformly generated. Since the phase transition of a 3-SAT problem occurs at a ratio of 4.3 between the number of clauses and the number of literals, each of the fifty problems contained 50 variables and 215 constraints to maintain the ratio of 4.3. Therefore, all fifty problems are hard and there is no guarantee that a solution exists.

5.2 Results and comparisons

In this section, we will present the results of MBO model, followed by a comparison between MBO and WalkSAT in the following section.

In Figure 3, we present the average number of solutions found for each spermatheca size. The scale of the z-axis is different in each figure based on the average number of solutions. The bottom right figure shows the average number of solutions found over the three different spermatheca sizes.

Table 1: *The average number of solutions found in relation to the number of queens and number of broods.*

Broods	Number of Queens				
	1	2	3	4	5
20	2.3 ± 1.5	4.3 ± 2.4	3 ± 1.7	3 ± 2.1	2.3 ± 1.3
40	4 ± 2.2	2.7 ± 1.4	3 ± 1.7	1.7 ± 1.3	2.3 ± 1.7
60	5.3 ± 2.9	4 ± 2.6	3 ± 2.1	0.7 ± 1	1 ± 0.5
80	5 ± 2.6	4.3 ± 2.4	3 ± 2.6	2.3 ± 2.1	3.3 ± 2.1
100	4.3 ± 3.3	3.3 ± 2.1	2.7 ± 1.6	2.3 ± 1.3	0.7 ± 0.6

As we can see from the top left figure in Figure 3, the peak of the graph occurs for two queens and 60 broods. With a spermatheca sizes of 14 and 21, a single queen achieves the best performance when the number

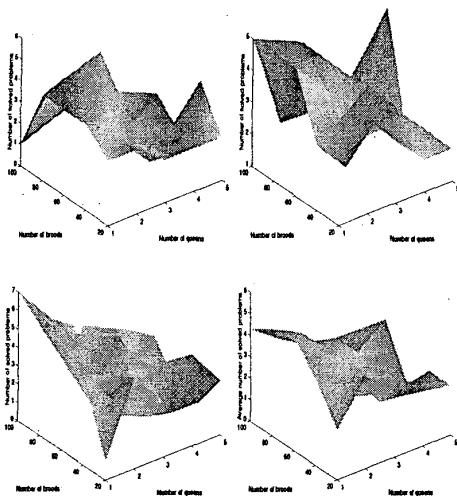


Figure 3: The number of solutions found for the fifty problems for each of the three used spermatheca sizes and the average of the three. The x-axis represents the number of queens, y-axis represents the number of broods, and the z-axis represents the average number of solutions found. Top left: spermatheca size of 7. Top right: spermatheca size of 14. Bottom left: spermatheca size of 21. Bottom right: the average of the three spermatheca sizes.

of broods is greater than 40. A consistent result among the three different spermatheca sizes - which can be also seen from the bottom right graph and Table 1 - is that a single queen is higher on the average than two or more queens when the number of broods is higher than 20.

This final result is very interesting in the sense that the behavior of the model on the propositional satisfiability problem is somehow consistent with the underlying biological model where a honey-bees' colony usually contains a single queen. Also the results suggest that an average number of broods is better than small or large number of broods. What is more interesting is that the best results achieved with the largest spermatheca size of 21. This spermatheca size reflects the number of drones the queen mated with in the mating-flight or equivalently, the number of accepted states in the annealing stage.

In summary, the best results occurred with the largest number of states accepted during the annealing state, the medium size of children or broods, and the smallest number of elitist solutions. Correspondingly, from biological point of view, the best performance occurred with the largest spermatheca size, an average

colony size, and the smallest number of queens.

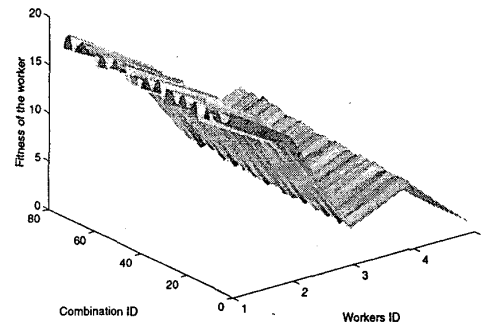


Figure 4: The fitness of each of the five workers over the 75 different configurations. Worker ID 1: WalkSAT, Worker ID 2: Random walk, Worker ID 3: Random new, Worker ID 4: 1-point crossover, and Worker ID 5: Random flip

Table 2: The average fitness of each of the five workers in all runs.

Worker	Average fitness
WalkSAT	18.1 ± 0.7
Random walk	14.3 ± 0.6
Random new	4.4 ± 0.8
1-point crossover	7.1 ± 0.5
Random flip	0.8 ± 0.0

Another interesting set of results that comes out of the MBO model is the workers' fitness. In Figure 4, we present the average fitness of each worker at the end of each run for each of the 75 combinations (number of queens, spermatheca size, and number of broods). Table 2 presents the average performance of each worker in all runs. As we can see from the figure, there is a consistency in the behavior of each worker regardless of the three user-defined parameters. This is valid as the performance of each worker is somehow independent of the number of queens, queen's spermatheca size, and number of mating-flights. The fittest worker is WalkSAT followed by random walk. A question may arise at this stage of whether WalkSAT alone would have achieved the best results overall without the additional overhead of this algorithm. The answer of this question is the context of the next section.

5.3 Comparison with WalkSAT

To find out whether WalkSAT alone would have achieved the best results overall without the additional

overhead of MBO, we solved the fifty problems with the same version of WalkSAT that we used in our implementation. The number of trial solutions in WalkSAT is set to 1200 (number of broods in all generations of our algorithm, which equals to the number of mating-flights times the number of broods per mating-flights). The number of flips in WalkSAT is set to 100 so that it is consistent with the number of trials by each worker to improve the broods in our algorithm.

Surprisingly, WalkSAT failed to find any solution for all problems. The average number of unsatisfied clauses found by WalkSAT in all runs over all problems was 8.7 ± 1.2 . This value is much higher than the corresponding value of 2.13 ± 1.1 representing the average number of unsatisfied clauses found by the best queen in the last generation of each run by MBO with 1 queen, 21 spermatheca size, and 60 broods. These results emphasize the importance of the cooperative behavior of the bees in MBO. The cooperative performance of the workers performed better than the performance of the best worker alone.

To summarize, the annealing stage undertaken by each queen during her mating-flight proved useful in the set of SAT problems solved here. In addition, a single queen deemed better with average-number of broods. More queens were useful when the number of broods was too small. Finally, the cooperative behavior between the different heuristics was more functional than a single heuristic in isolation.

6 Conclusion

In this paper, a new heuristic, MBO, based on the marriage process in honey bees was introduced. The biological motivation and the computational aspects of the algorithm were both discussed. From the analysis of the experimental results, MBO was very successful on a group of fifty hard 3-SAT problems. The main advantage from our perspective was that the algorithm preserved many of the underlying biological concepts and achieved the good performance with parameters taken from real biological concepts. Moreover, it was shown that MBO performed better than WalkSAT alone although WalkSAT was the heuristic with the highest fitness in MBO.

Bibliography

- Adams, J., E. Rothman, W. Kerr, and Z. Paulino (1972). Estimation of the number of sex alleles and queen matings from diploid male frequencies in a population of *apis mellifera*. *Genetics* 86, 583–596.
- Bonabeau, E., M. Dorigo, and G. Theraulaz (1999). *Swarm intelligence: from natural to artificial systems*. Oxford Press.
- Cook, S. and D. Mitchell (1997). Finding hard instances of the satisfiability problem: A survey. In D. S. in Discrete Mathematics and T. C. Science (Eds.), *Satisfiability Problem: Theory and Applications*. American Mathematical Society.
- Dietz, A. (1986). Evolution. In T. Rinderer (Ed.), *Bee genetics and breeding*, pp. 3–22. Academic Press, Inc.
- Dorigo, M., E. Bonabeau, and G. Theraulaz (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems* 16, 851–871.
- Dorigo, M. and G. Caro (1999). The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover (Eds.), *New Ideas in Optimization*, pp. 11–32. McGraw-Hill.
- Dorigo, M. and L. Gambardella (1997). Ant colony system: a cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation* 1, 53–66.
- Gent, I. and T. Walsh (1994). The satisfiability constraint gap. Technical Report 702, University of Edinburgh.
- Hoos, H. (1999a). On the run-time behaviour of stochastic local search algorithms for sat. *Proceedings of AAAI*, 661–666.
- Hoos, H. (1999b). Sat-encodings, search space structure, and local search performance. *Proceedings of IJCAI'99*, 296–302.
- Hoos, H. and T. Stützle (2000). Local search algorithms for sat: An empirical evaluation. *Journal of Automated Reasoning* 24, 421–481.
- Laidlaw, H. and R. Page (1986). Mating designs. In T. Rinderer (Ed.), *Bee Genetics and Breeding*, pp. 323–341. Academic Press, Inc.
- Page, R., R. Kimsey, and H. Laidlaw (1984). Migration and dispersal of spermatozoa in spermathecae of queen honey bees: *Apis mellifera*. *Experientia* 40, 182–184.
- Papadimitriou, C. and K. Steiglitz (1982). *Combinatorial optimization: algorithms and complexity*. Prentice-Hall.
- Perez-Urbe, A. and B. Hirsbrunner (2000). Learning and foraging in robot-bees. *CEC2000*.
- Rinderer, T. and A. Collins (1986). Behavioral genetics. In T. Rinderer (Ed.), *Bee Genetics and Breeding*, pp. 155–176. Academic Press, Inc.
- Selman, B. and H. Kautz (1993a). Domain-independent extensions to gsat: Solving large

- structured satisfiability problems. *Proceedings of IJCAI'93*.
- Selman, B. and H. Kautz (1993b). An empirical study of greedy local search for satisfiability testing. *Proceedings of AAAI*.
- Selman, B., H. Kautz, and B. Cohen (1994a). Local search strategies for satisfiability testing. *Proceedings of 2nd DIMACS Challenge on Cliques, Coloring and Satisfiability*.
- Selman, B., H. Kautz, and B. Cohen (1994b). Noise strategies for improving local search. *AAAI94*, 337–343.
- Selman, B., H. Levesque, and D. Mitchell (1992). A new method for solving hard satisfiability problems. *Proceedings of AAAI*, 440–446.
- Zhang, H. and M. Stickel (1994). Implementing davis-putnam's method by tries. Technical report, The University of Iowa.