Ross Anderson, Devin Kennedy, Ben Rubin
March 14, 2009
rma35@cornell.edu, drk35@cornell.edu, ber24@cornell.edu

Ant Program

For ants, diploid queens produce haploid males by parthenogenesis. Thus males contain only alleles from their parent queens, so examining male genotypes allows us to infer the queens' genotypes directly. While this process can be done by inspection for colonies with a small number of queens, it becomes very difficult as the queen number increases.

To deal with this, we wrote a program to compute the least number of queens that could have produced the observed males. To explain how this program works, we first must create some definitions.

We refer to combinations of males from a colony as *groups*. We define a *group* to be *feasible* if all the males in the *group* could have been produced by a single queen. Finally, we define a *feasible group* to be *maximal* if there are no other *feasible groups* that contain it. Equivalently, we could say that a *feasible group* is *maximal* if we cannot add any of the remaining males to the group without violating the group's feasibility.

We used a two step algorithm. First, we list all of the *feasible, maximal groups* of males for a given colony. Then, we find the least number of these *feasible, maximal groups* of males from this list such that each male is in at least one such *group*. The males in each group specify requirements of the queen's *inferred genotype*, as the queen must be capable of producing all these males.

We used the following algorithm to compute the list of *feasible, maximal groups*, denoted $\mathscr{L}_m$. First, we create a temporary list $\mathscr{L}_f$ of known *feasible groups*. Clearly every pair of males can be produced by a queen, as each male has only one allele for each locus while each queen has two alleles. Thus, for each pair, we create a *feasible group* and add it to the list $\mathscr{L}_f$. For each *feasible group*, we maintain an *inferred genotype* for the queen that could have produced these males, specifically which alleles the queen must have at each locus. Now, starting for $n = 2$, for each *feasible group* G in $\mathscr{L}_f$ of n males, for each of the males i not in the group, we check if we can add i to make a new *feasible group*.[1] If we can, then we add this new group to $\mathscr{L}_f$. If this new group is *infeasible* for every i, then we check $\mathscr{L}_m$ to ensure that no existing maximal group contains this group, and if so, we add the group to $\mathscr{L}_m$. We continue in this manner until there are no *feasible groups* of size n.

Given a *feasible group* G of n males and a male i not in G, adding i to G will maintain *feasibility* if for each locus, the *inferred genotype* of group G already has the allele of the i, or the *inferred genotype* has less than two alleles at this locus.

Notice that if $n + 1$ males can form a *feasible group*, then clearly any n males from this group can also form a *feasible group*, so if we have all possible *feasible groups* of size n and we cannot make a *feasible group* of size $n + 1$, there are no *feasible groups* of size greater than n, so we can stop.

The second step, where we find the least number of *maximal feasible groups* from our list such that each male is in at least one group, is an instance of the classic computer science problem Set Cover. While in general the Set Cover problem is very difficult to solve as the problem becomes large, the problems we encountered were small to be tractable.[2] We wrote our problem as an integer program. If there are s total *maximal feasible groups* in our list, let x_i for $i = 1, \ldots, s$ be the decision variable that has value 1 if group i is chosen and has value 0 otherwise. Additionally, if we have observed t males, then let $m_{i,j}$ for $i = 1, \ldots, s$ and

[1] Note that this process, exactly as described, would cause us to add the same group to $\mathscr{L}_f$ multiple times. However, by using a clever ordering, we can ensure that we need only consider each group once.

[2] In practice, we found that a colonies with around 60 ants and from 2 to 8 queens would produce a Set Cover problem with 60 variables and around 100 constraints, a problem that can be solved quickly. However, if we were not to go through the trouble of considering only sets of *maximal feasible groups*, we would produce an exponentially large integer program that was not tractable.

$j = 1, \ldots, t$ be 1 if male j is in group i and 0 otherwise. Then we can write our integer program as

$$\begin{aligned}
\min_{x} \quad & \sum_{i=1}^{s} x_i & \\
\text{subject to} \quad & \sum_{i=1}^{s} x_i m_{i,j} \geq 1 & j = 1, \ldots, t \\
& x_i \in \{0, 1\} & i = 1, \ldots, s
\end{aligned}$$

There are many software packages for solving integer programs. We used the *Stand-alone LP/MIP Solver* of the open source software *GNU Linear Programming Kit*.

From the minimum number of *maximal feasible groups*, we know the minimum number of queens required to produce the males we observed. Further, from the *inferred genotypes* of each of the chosen *feasible groups* we can infer the genotypes of the colony's queens.