algorithmOne.c

In the first algorithm when there is no control when grabbing their forks, it always ends in deadlock unless they all sleep at separate times. If they all sleep for the same time when eating or thinking they might be all be able to eat at least once before they all deadlock. The number of times a philosopher gets to eat when they don't sleep appears to have a lot of variation. However, it does seem that the first philosopher gets the most and subsequent philosophers that don't share forks tend to eat more.

algorithmTwo.c

By creating a condition that makes sure that the forks value is 1, the program lasts much longer. However, it still eventually deadlocks. Although the more philosphers present at the table the longer the program lasts without deadlock. The philosophers get to eat thousands of times and each at a similar frequency. However, if the philosophers sleep on thinking or eating the program will deadlock almost immediately.

algorithmThree.c

By making it so that only n-1 philosophers can be at the table at a time there appears to be no deadlock since there will always be at least one philosopher that can eat. However, you can cause the program to have starvation by making the philosophers sleep once they start to eat. By sleeping after they put the forks down the philosophers actually have less starvation all around.