Because there can only be one philosopher waiting for chopsticks, the program will not go into deadlock. However, there may be Starvation. For example:
Philosopher A finished thinking, picked up chopsticks a and b, signal mutex.
Philosopher C finished thinking, picked up chopsticks c and d, signal mutex.
Philosopher E finished thinking, picked up chopsticks e, and wait for a. No one can wait for chopsticks before A finish eating.
Thus, there may be starvation when any philosopher spends long time eating.

The solution to this problem is to use a separate lock to represent each chopstick. However, the program goes into deadlock when all five philosophers pick up the right chopstick at the same time. One way to deal with this is to have a philosopher always pick up the left chopstick before picking up the right chopstick (wait(f[i-+1 mod 5]) first).