Documentation

We have tried to reuse as much as possible from the sequential solver and the solution is based on the fork after value. It will fork at the next intersection after the fork after value has been reached. If a thread has checked all it's saved positions and reaches a dead end it will hold and wait for the other threads to finish.
The set of visited cells is shared so that the threads don't pass a cell that has already been passed by one of the other threads. We also use a shared boolean so all threads know when the goal is found by any other thread and then knows to terminate.
If a thread finds the goal we set the shared boolean value to true. The search is running while the goal is not found, so all threads will stop when the goal is found and the path is combined and then returned. If the goal is not found, null is returned.

We have tested the solution on both maps and with different fork after values. We have also tried to move the goal to different positions, both reachable and unreachable, and the solution works as expected.