```
def solve(graph)
      if(graph.rem_food == 0)
            graph.update_optimal()
            return false  ← dead end
     for(Food in graph)
          "apply move to p to get p'"
         " solve (p')"
         "undo move"
          graph.path.current_food = Food
calc energy   if (not graph.path.isValid(g[Food + 1]))
gain                   return false
          net= calculate_net(food, next Food)
          graph.path.update_net(net )
update   g.path.add(food)
net energy  solve(graph)
          graph.path.current_food = previous Food
update    graph.path.update_net(-net)
path      g.path.remove(food)

     return g.optimal_path
```