

COURSE OUTLINE

- **Overview of FPGAs and ASICs**
- **Using Synthesis**
- **HDL Examples**
- **Simulation and Testing**
- **Physical Place and Route**
- **Testing ASICs**
- **Component Reuse**

PLACEMENT AND ROUTING

- The goal of placement and routing is to map *optimally* the structural interconnection of components expressed in a schematic (net-list) onto the target physical architecture.
- Thus, placement and routing seeks to minimize the *cost* of the chip by packing the logic and wires into the smallest possible *area*.
- The software also tries to minimize the time *delays* of critical paths so the chip will run fast.
- A combination of area and delay can be expressed as *total wire length*.
- Since hundreds of components and nets are involved, the task is *formidable* for humans and very difficult even for clever software.
- The task is highly *order dependent* so it can be directed by the designer assigning weights to critical nets. Multiple solutions should be obtained and evaluated since they may vary by 30%-50%.

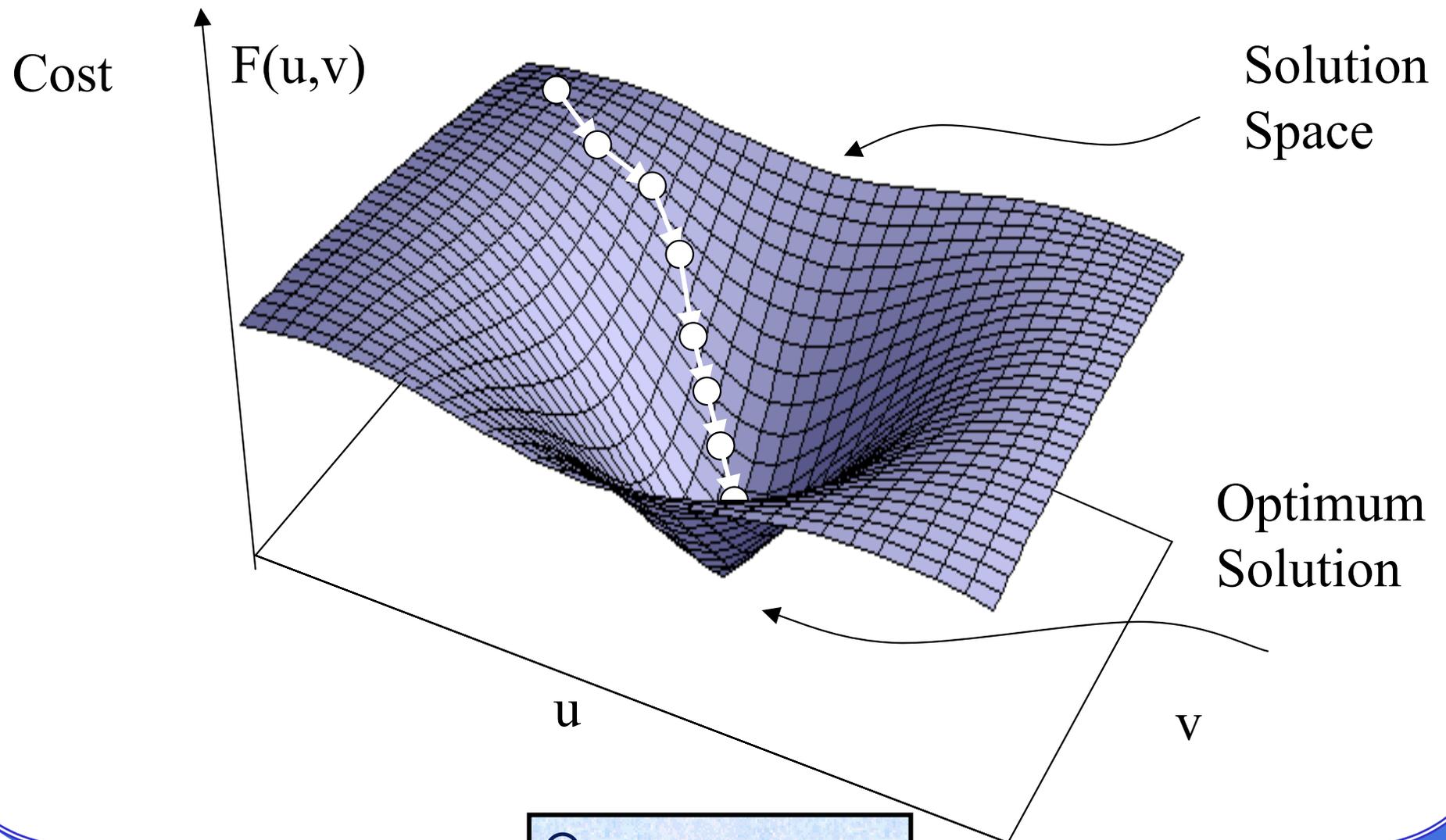
EFFECT OF PLACE AND ROUTE ON TIMING

- Simulations performed prior to layout can verify functionality and some relative timing using *assumed* values for interconnection delay.
- Once placement and routing has been performed, the *actual* timing delays can be back-annotated to the logic simulator for timing verification.
- Since FPGAs have programming elements which introduce RC delays that are much slower than metal vias, the number of these *puddles* that are used to interconnect components should be minimized. Most FPGAs use *tracks or segments* to help with this problem.
- The accepted layout is a *tradeoff* between desired quality and available time and computing resources.

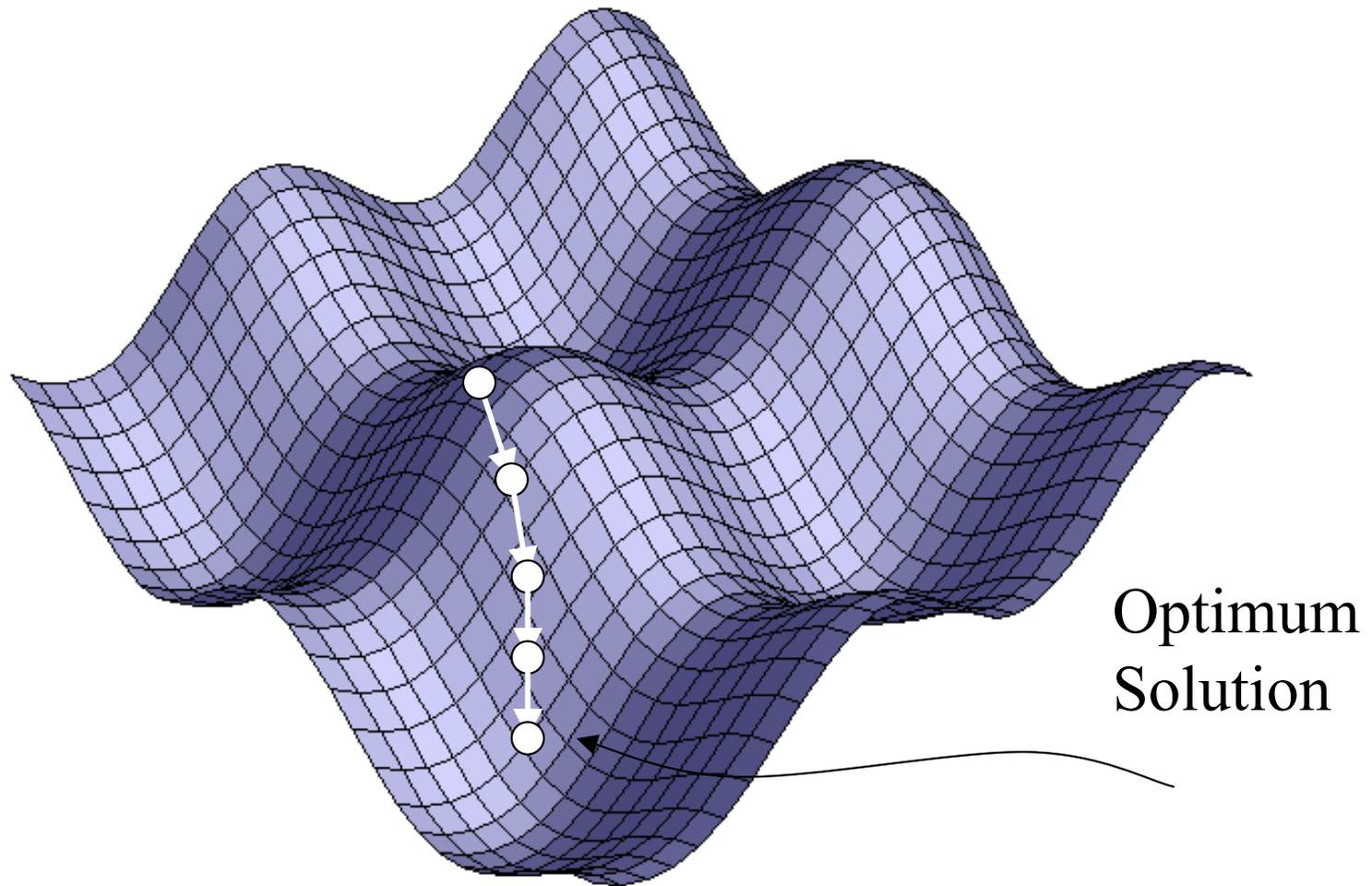
FINDING THE OPTIMAL SOLUTION

- **To find the optimal solution, we must first model the problem as a graphical space.**
- **Next we select an initial temporary solution.**
- **Then we look for improved solutions to find our way “downhill” which is closer to the optimum.**
- **One approach is to compute a gradient to the surface, which is essentially a line tangent to the surface, pointing downwards.**
- **Then follow this gradient downwards and stop at the bottom of a valley.**

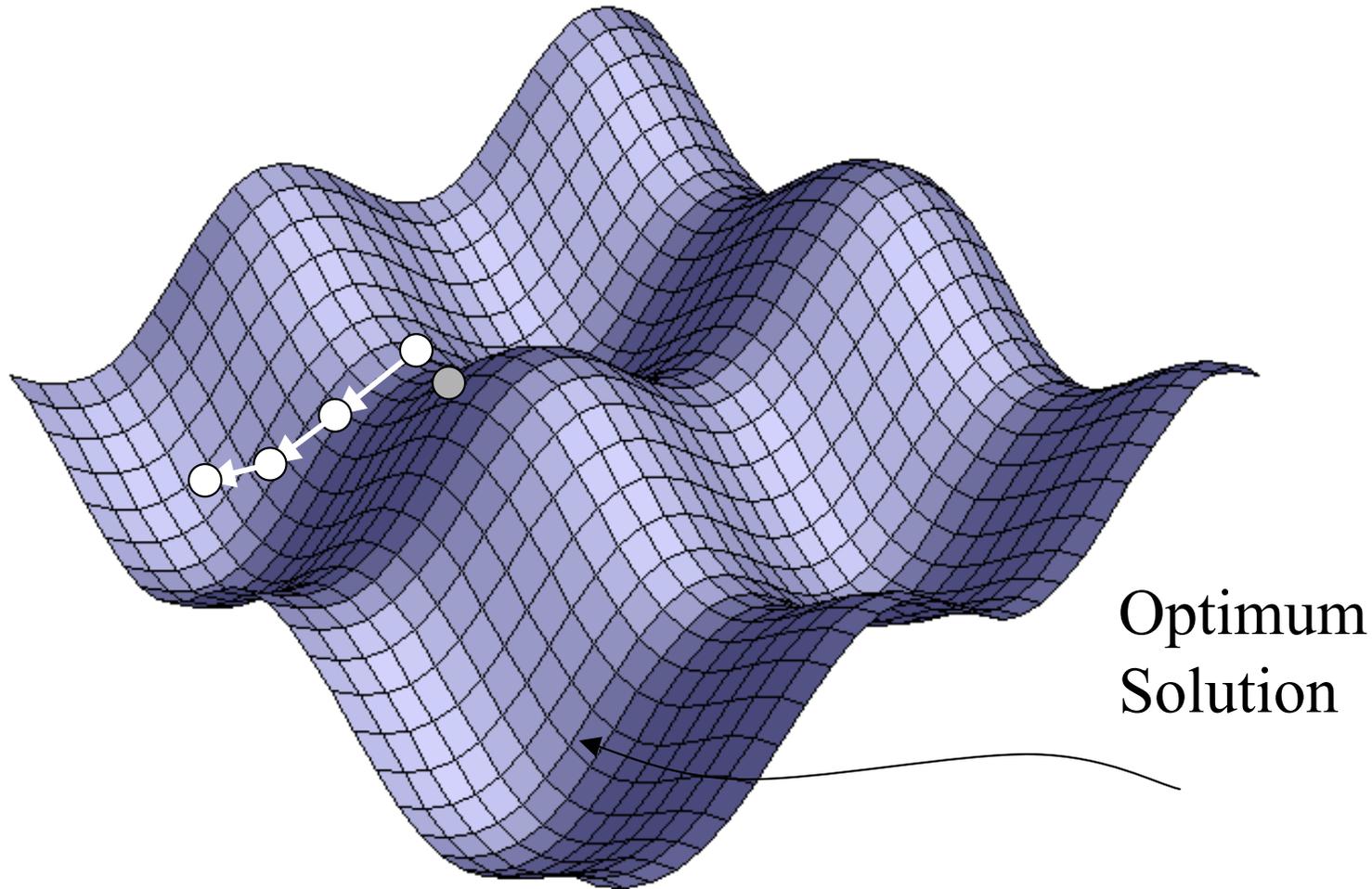
USING THE GRADIENT DESCENT METHOD



WE MAY FIND THE OPTIMUM



BUT WE MAY GET TRAPPED IN A LOCAL MINIMUM



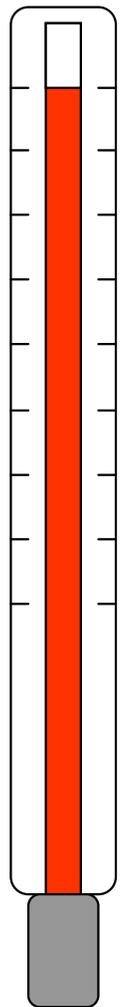
CLEVER HILL CLIMBING IS MORE LIKELY TO FIND THE GLOBAL OPTIMUM

- **To avoid getting trapped in a local minimum, we need to move uphill sometimes.**
- **Ideally, we should move uphill less as we get closer to the global minimum.**
- **If we assume that we get closer to the global minimum every step (on the average), we can implement this by moving uphill less often with each step taken.**

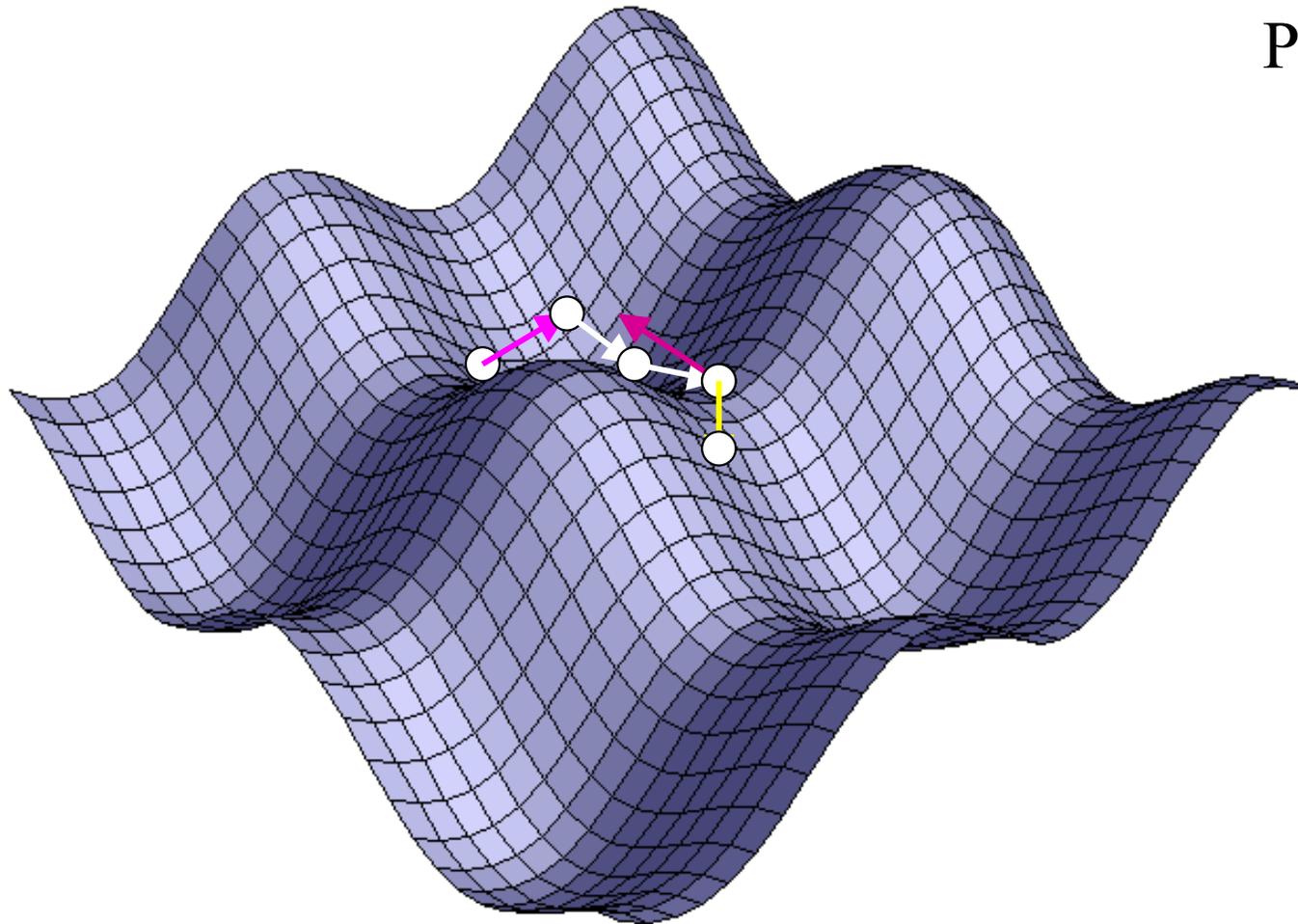
SIMULATED ANNEALING CAN BE USED TO SOLVE OPTIMIZATION PROBLEMS

- **When molten metals are cooled slowly, at a controlled rate of temperature decrease, they can form a single crystal of metal, which is the minimum energy configuration.**
- **If they are not cooled slowly enough, they will instead form amorphous or polycrystalline solids, which are higher energy configurations.**
- **We can apply this technique to our optimization problems.**

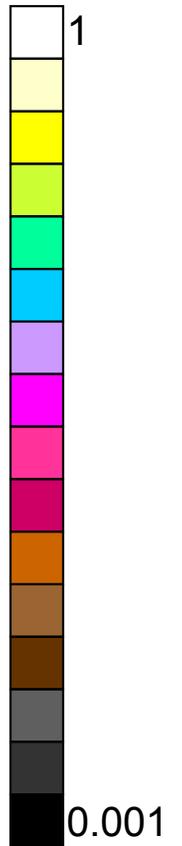
SIMULATED ANNEALING CAN CLIMB HILLS



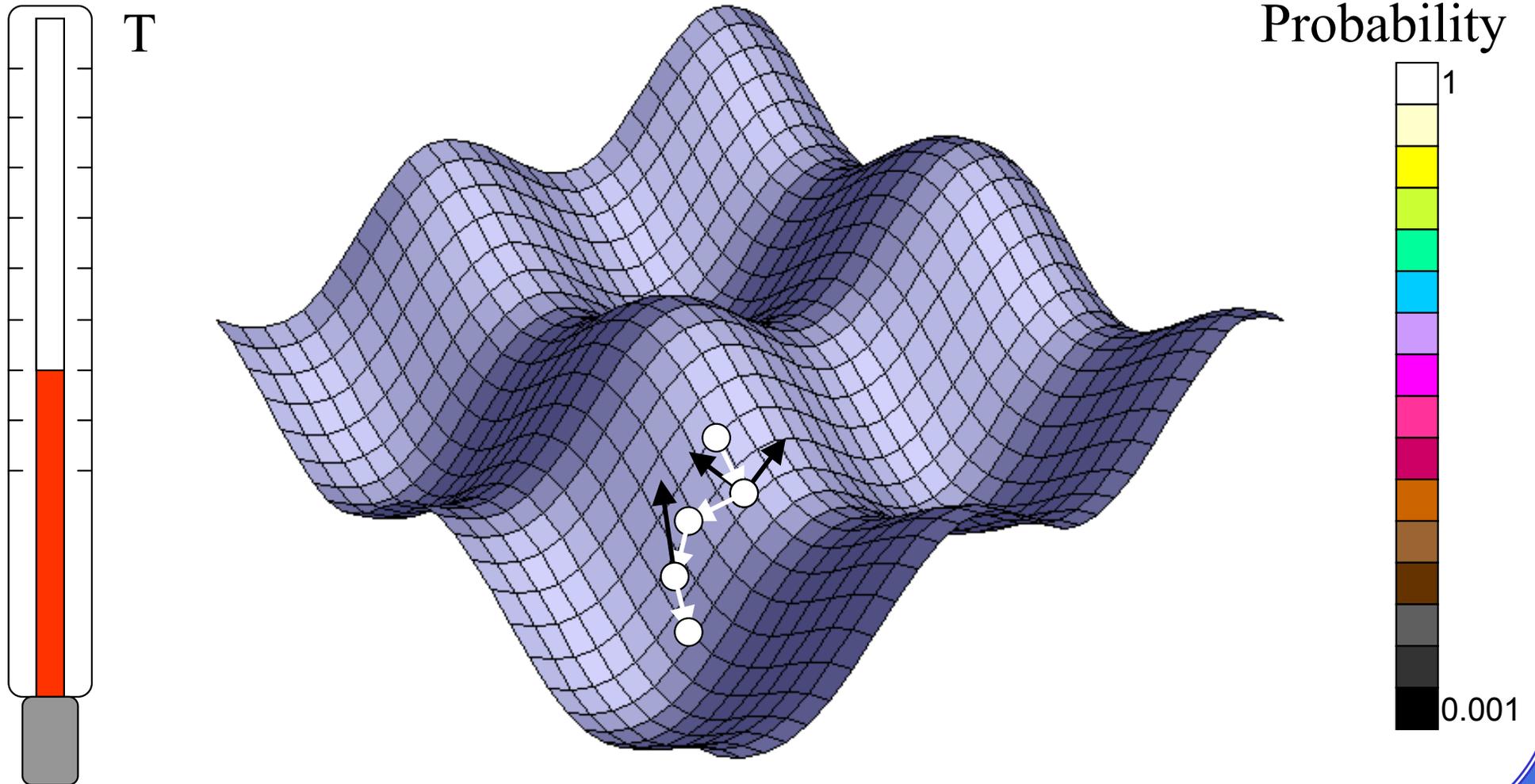
T



Probability



SIMULATED ANNEALING IS USUALLY THE BEST METHOD

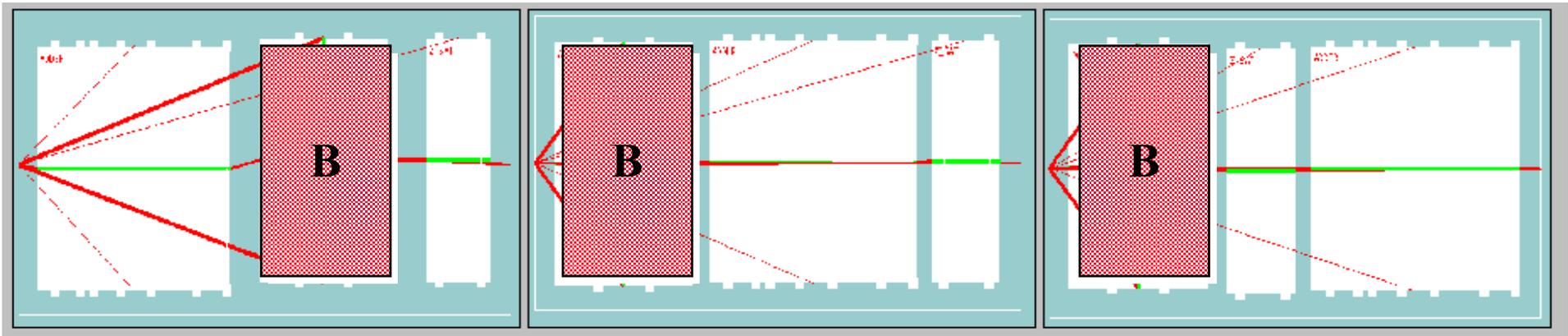


THE ORDER OF PLACEMENT AFFECTS THE FINAL RESULT

Order = ABC

Order = ABC

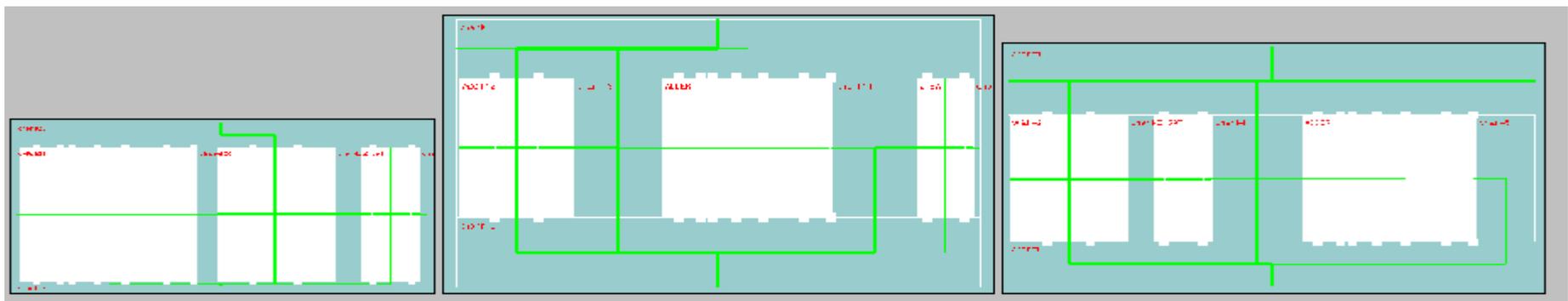
Order = ABC



Area = 125

Area = 260

Area = 250



PLACEMENT AND ROUTING SUMMARY

- The goal of placement and routing is to map *optimally* the structural interconnection of components expressed in a schematic onto the target physical architecture.
- A combination of area and delay (price and performance) can be expressed as *total wire length*.
- Since hundreds of components and nets are involved, the task is *formidable* for humans and very difficult even for clever software.
- The task is highly *order dependent* so the accepted layout is a tradeoff between desired quality and available time for computing resources.