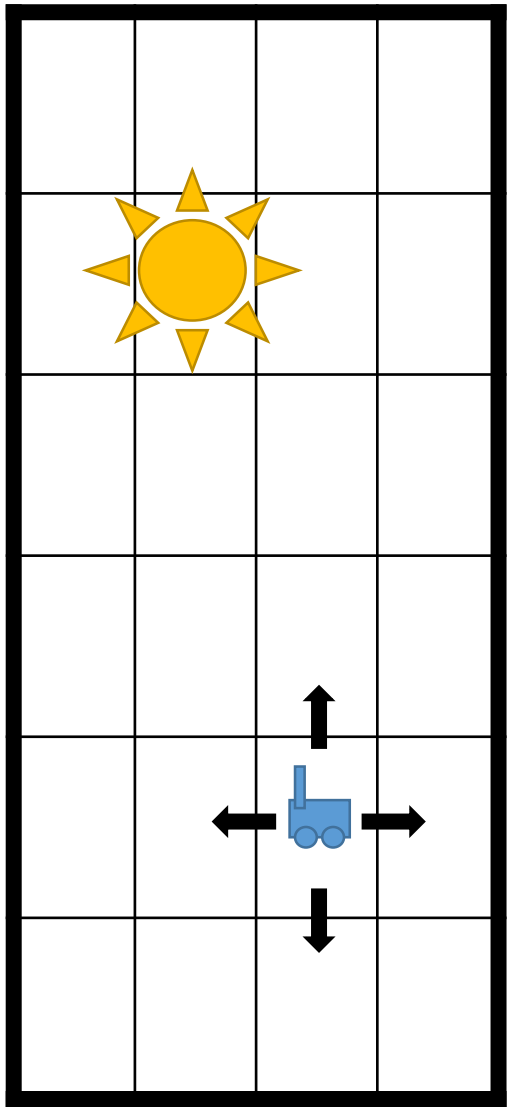
Q1-1: A robot in a large 2D room needs to find the location with the most sunlight so it can recharge. There is 1 skylight letting in light. What optimization strategy should you use?

1. Random search

2. Hill climbing without random restarts

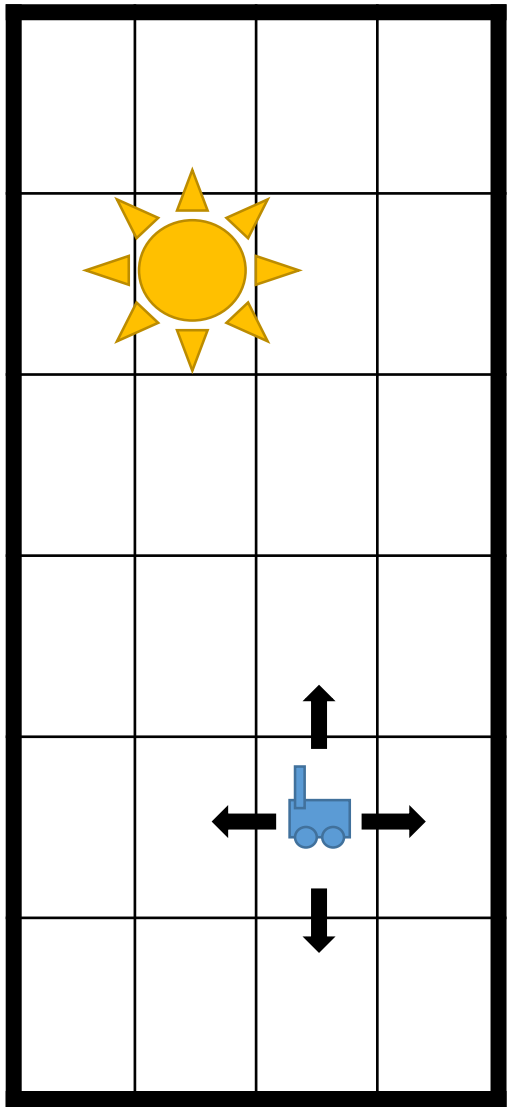
3. Stochastic hill climbing without random restarts

4. First-choice hill climbing without random restarts



Q1-1: A robot in a large 2D room needs to find the location with the most sunlight so it can recharge. There is 1 skylight letting in light. What optimization strategy should you use?

1. Random search
2. Hill climbing without random restarts
3. Stochastic hill climbing without random restarts
4. First-choice hill climbing without random restarts



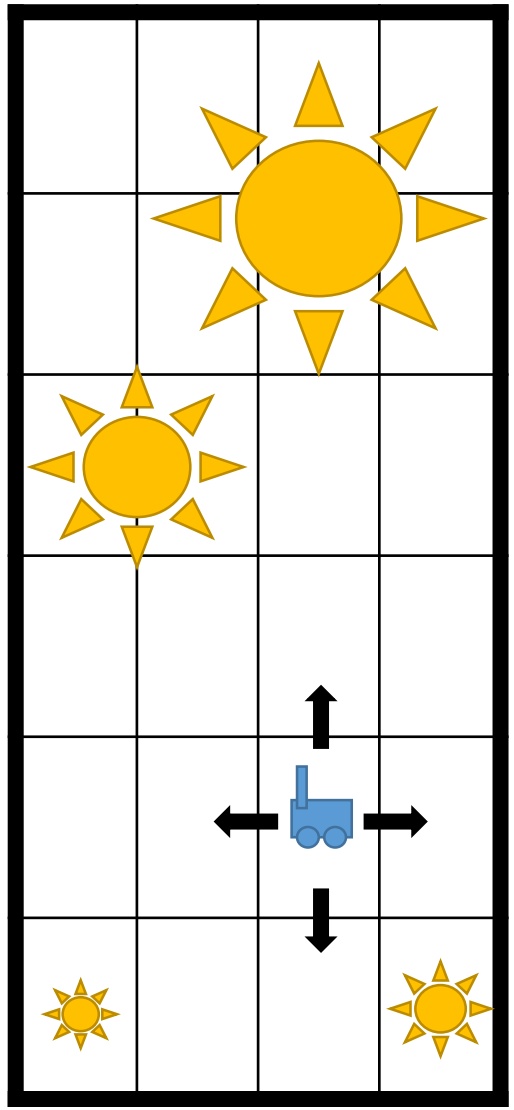
Q1-2: A robot in a large 2D room needs to find the location with the most sunlight so it can recharge. There are many skylights letting in different amounts of light. What optimization strategy should you use?

1. Random search

2. Hill climbing without random restarts

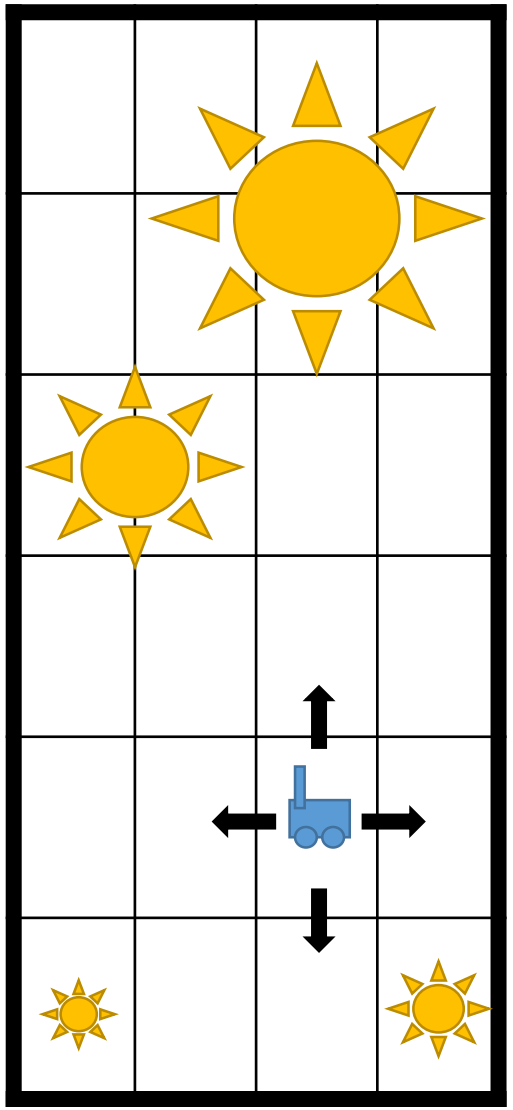
3. Stochastic hill climbing without random restarts

4. First-choice hill climbing without random restarts



Q1-2: A robot in a large 2D room needs to find the location with the most sunlight so it can recharge. There are many skylights letting in different amounts of light. What optimization strategy should you use?

1. Random search
2. Hill climbing without random restarts
3. Stochastic hill climbing without random restarts
4. First-choice hill climbing without random restarts



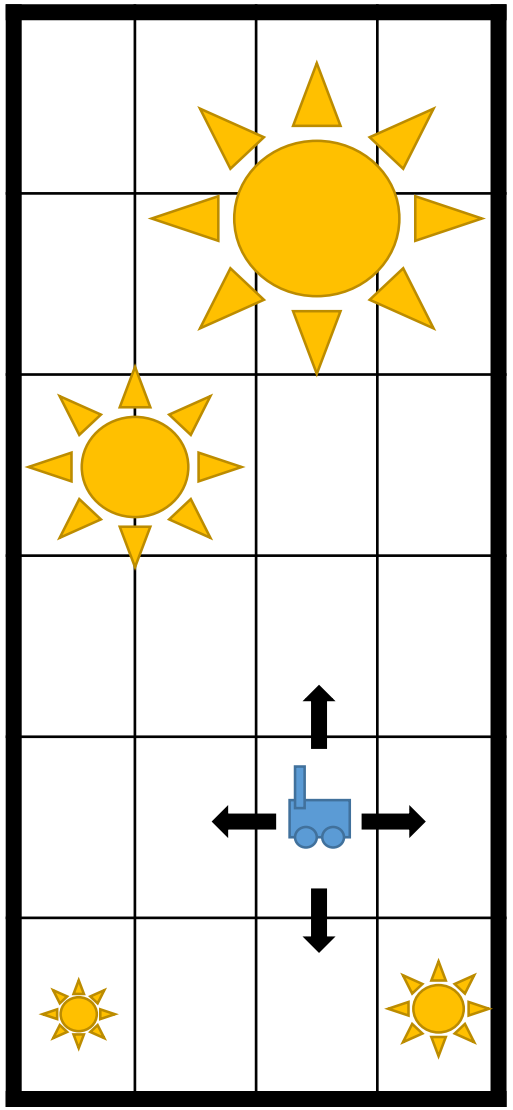
Q1-3: A robot in a large 2D room needs to find the location with the most sunlight so it can recharge. There are many skylights letting in different amounts of light. What optimization strategy should you use to guarantee you find the global maximum?

1. Random search

2. Hill climbing without random restarts

3. Stochastic hill climbing without random restarts

4. First-choice hill climbing without random restarts



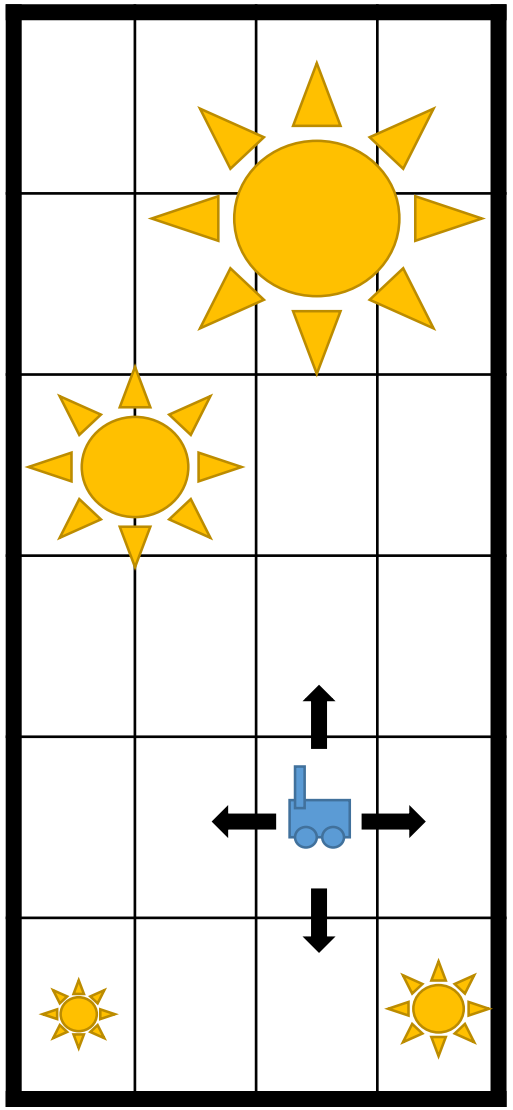
Q1-3: A robot in a large 2D room needs to find the location with the most sunlight so it can recharge. There are many skylights letting in different amounts of light. What optimization strategy should you use to guarantee you find the global maximum?

1. Random search

2. Hill climbing without random restarts

3. Stochastic hill climbing without random restarts

4. First-choice hill climbing without random restarts



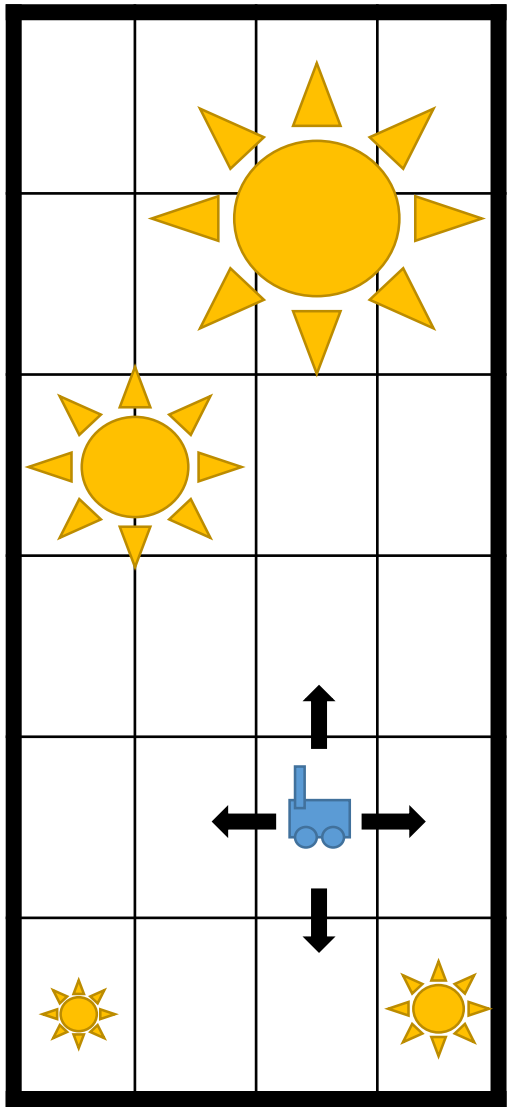
Q1-4: A robot in a large 2^100D room needs to find the location with the most sunlight so it can recharge. There are many skylights letting in different amounts of light. What optimization strategy should you use?

1. Random search

2. Hill climbing without random restarts

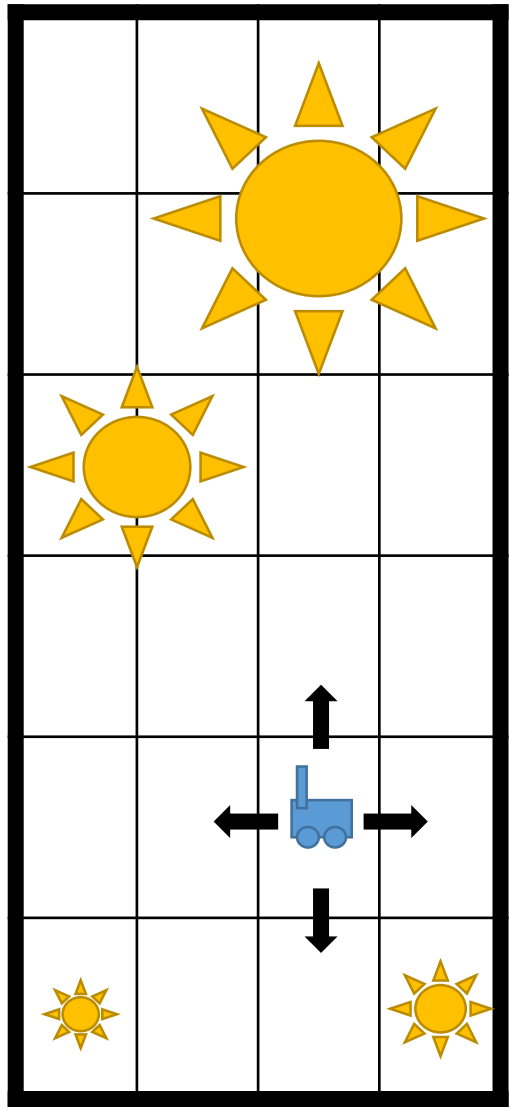
3. Stochastic hill climbing without random restarts

4. First-choice hill climbing without random restarts



Q1-4: A robot in a large 2^100D room needs to find the location with the most sunlight so it can recharge. There are many skylights letting in different amounts of light. What optimization strategy should you use?

1. Random search
2. Hill climbing without random restarts
3. Stochastic hill climbing without random restarts
4. First-choice hill climbing without random restarts



Q2-1: What is an advantage of simulated annealing over hill climbing, stochastic hill climbing, and first-choice hill climbing?

1. It is guaranteed to find the global optimum
2. Algorithms inspired by real world processes work better
3. It is less vulnerable to getting stuck in local optima
4. It terminates more quickly

Q2-1: What is an advantage of simulated annealing over hill climbing, stochastic hill climbing, and first-choice hill climbing?

1. It is guaranteed to find the global optimum
2. Algorithms inspired by real world processes work better
3. It is less vulnerable to getting stuck in local optima
4. It terminates more quickly

Q2-2: What is a not an advantage of simulated annealing over random search?

1. It is less vulnerable to getting stuck in local optima

2. It can make use of a good neighbor function

3. It can climb a local optimum

Q2-2: What is a not an advantage of simulated annealing over random search?

1. It is less vulnerable to getting stuck in local optima

2. It can make use of a good neighbor function

3. It can climb a local optimum

Q2-3: What setting will give the largest probability of moving to state *t*?

1. *f*(*t*) is close to *f*(*s*) and *Temp* is large

2. *f*(*t*) is close to *f*(*s*) and *Temp* is small

3. *f*(*t*) is much less than *f*(*s*) and *Temp* is large

4. *f*(*t*) is much less than *f*(*s*) and *Temp* is small

$$\exp\left(-\frac{|f(s)-f(t)|}{Temp}\right)$$

Q2-3: What setting will give the largest probability of moving to state *t*?

1. *f*(*t*) is close to *f*(*s*) and *Temp* is large

$$\exp\left(-\frac{|f(s)-f(t)|}{Temp}\right)$$

2. *f*(*t*) is close to *f*(*s*) and *Temp* is small

3. *f*(*t*) is much less than *f*(*s*) and *Temp* is large

4. *f*(*t*) is much less than *f*(*s*) and *Temp* is small

Q3-1: Recall the optimization problem of buying *k* computer parts from *k* stores, one part per store, at the least cost. What would be a good encoding for a genetic algorithm?

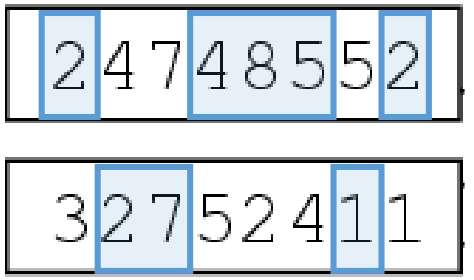
1. A graph in which edges connect a store with the part purchased there
2. A list of strings <*part*>-<*store*> sorted by the price of that part
3. A string with a fixed order for the parts and integer index for each store

Q3-1: Recall the optimization problem of buying *k* computer parts from *k* stores, one part per store, at the least cost. What would be a good encoding for a genetic algorithm?

1. A graph in which edges connect a store with the part purchased there
2. A list of strings <*part*>-<*store*> sorted by the price of that part
3. A string with a fixed order for the parts and integer index for each store

Q3-2: Consider an alternative crossover operation for a genetic algorithm run on N-queens that randomly selects several columns from one parent and the remaining columns from the other parent. Will this work as well as the standard crossover? Why?

1. No, it is impossible to select the indices in this manner
2. No, it will not preserve partial solutions on one side of the board
3. Yes, it still helps make large random moves in the state space
4. Yes, it is more closely related to real biology



Q3-2: Consider an alternative crossover operation for a genetic algorithm run on N-queens that randomly selects several columns from one parent and the remaining columns from the other parent. Will this work as well as the standard crossover? Why?

1. No, it is impossible to select the indices in this manner
2. No, it will not preserve partial solutions on one side of the board
3. Yes, it still helps make large random moves in the state space
4. Yes, it is more closely related to real biology