Search-Based Procedural Content Generation (SBPCG)

Matthew Guzdial

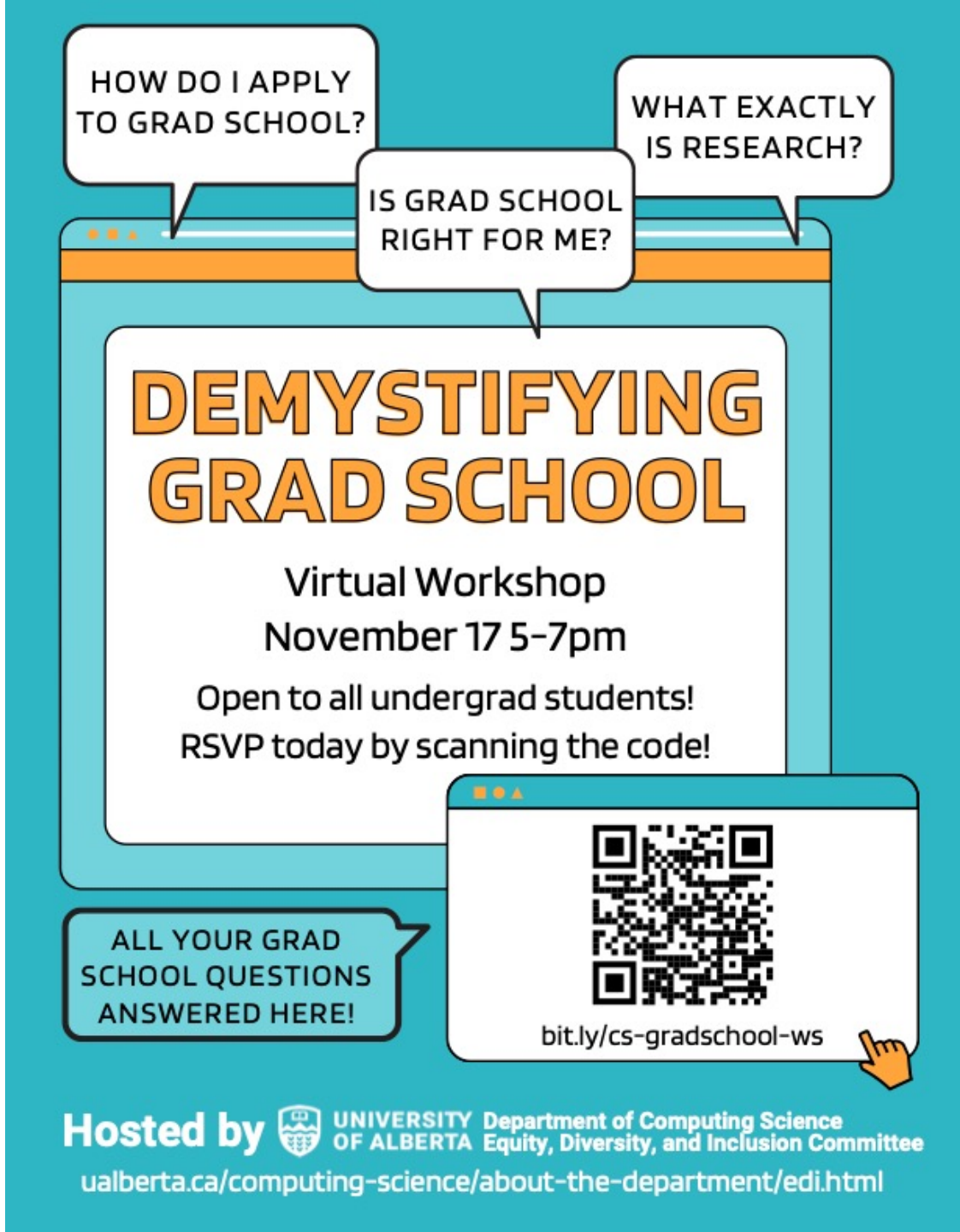
guzdial@ualberta.ca

UNIVERSITY OF ALBERTA

Announcements

- Hope you had a nice reading week!
- Review Quiz 4 answers on Wednesday (no time today!)
- Quiz 4 marks out by next Monday
- Assignment 4 due tonight at 11:55pm (23 hour grace period)
- Assignment 5 released!

Demystifying Grad School
Nov 17 5-7pm
bit.ly/cs-gradschool-ws



Game AI Top 8 (76/86)

1. Reinforcement Learning in Games (53.9%)
2. Balancing Game AI (42.1%)
3. AI for Game Design (39.5%)
4. Automated Playtesting (36.8%)
5. Automated Game Playing (30.3%)
6. AI-based Game Design (28.9%)
6. PCG via Machine Learning (28.9%)
8. Generated Dialogue and story (27.6%)

…

9. Game AI in Academia (25%)
10. Mixed-initiative PCG and More Player Modelling (22.4%)

Review

Constructive PCG

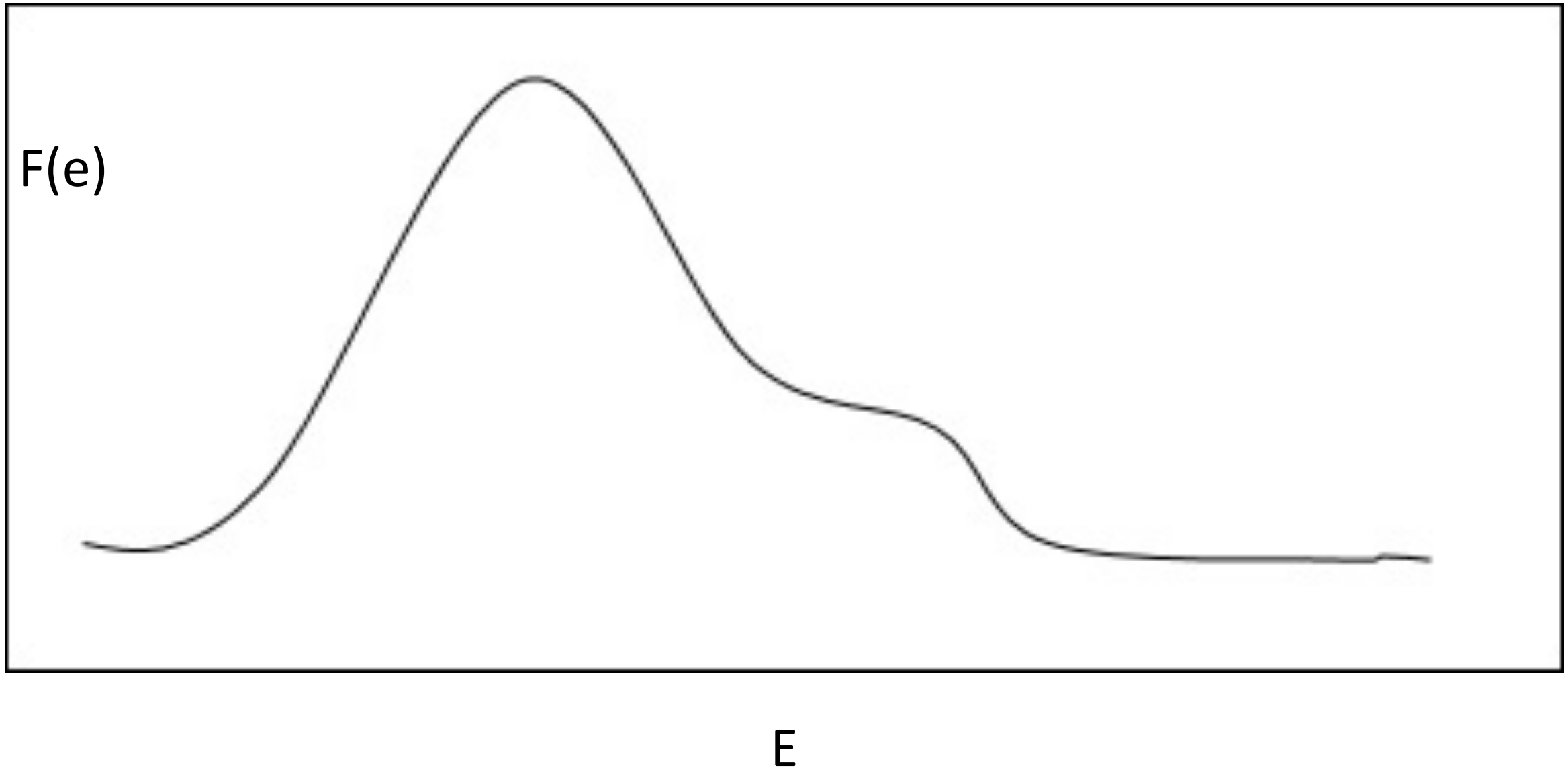
Various approaches for putting together existing content piece-by-piece.

Search-based PCG

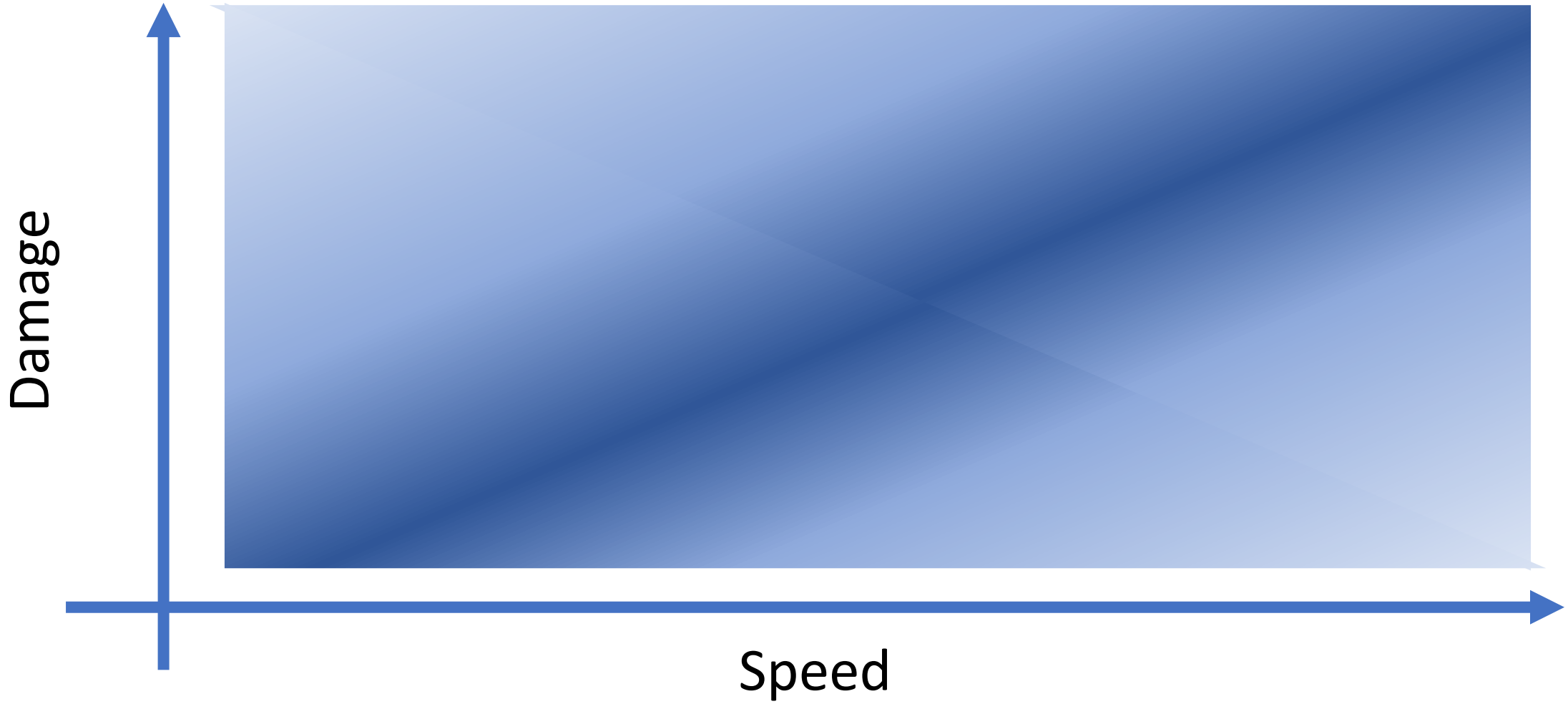
Search across complete pieces of content for the best according to some fitness function.

Search-based PCG Vocabulary

- **Search Space (*S*)**: A space with (generally) fixed dimensions.
- Each point in this space is an **entity (*e*)**, which is a complete piece of content (a song, a story, a level, a character, etc.)
- **Fitness Function (*F*)**: A function that maps each entity F(e)-> to some numeric value.
- **Neighbors (N(e))**: Entities “next” to some entity e.


F(e)
E

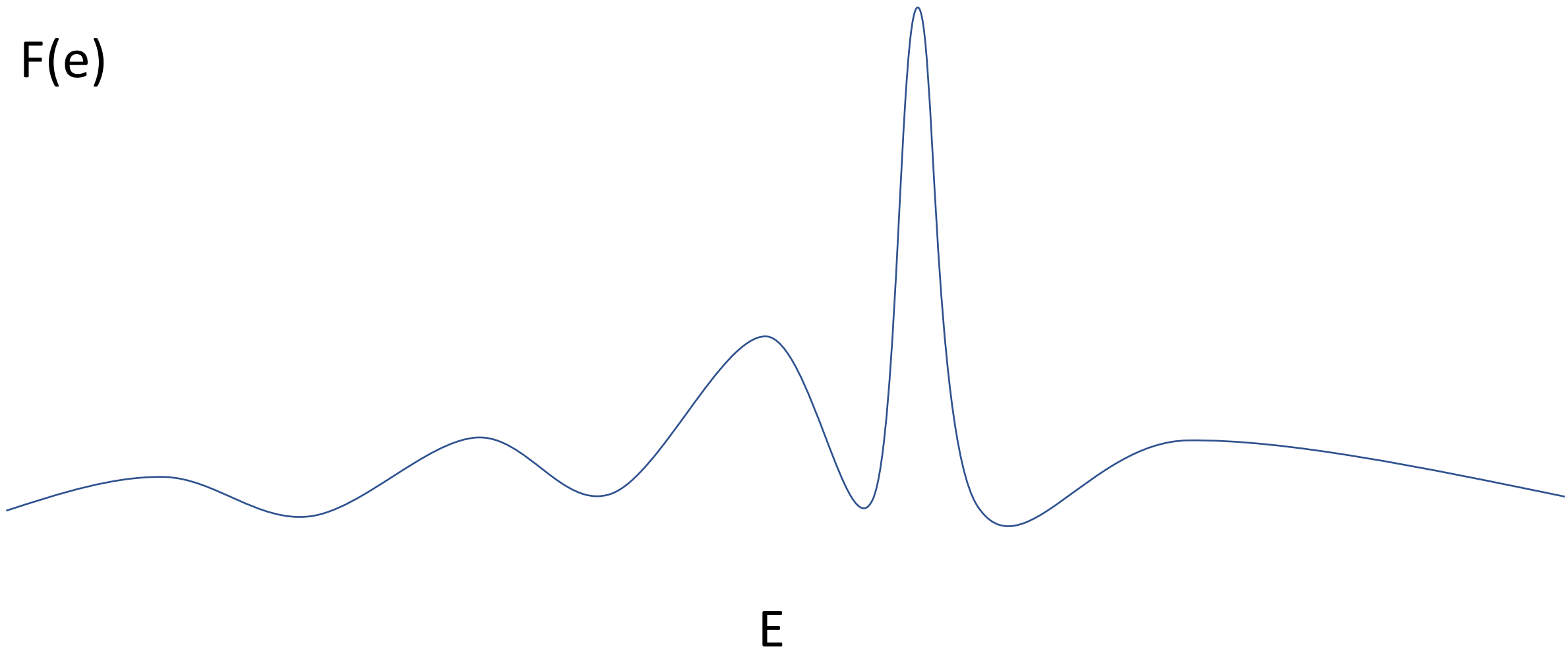
Example: Weapon Generation (Damage and Speed)



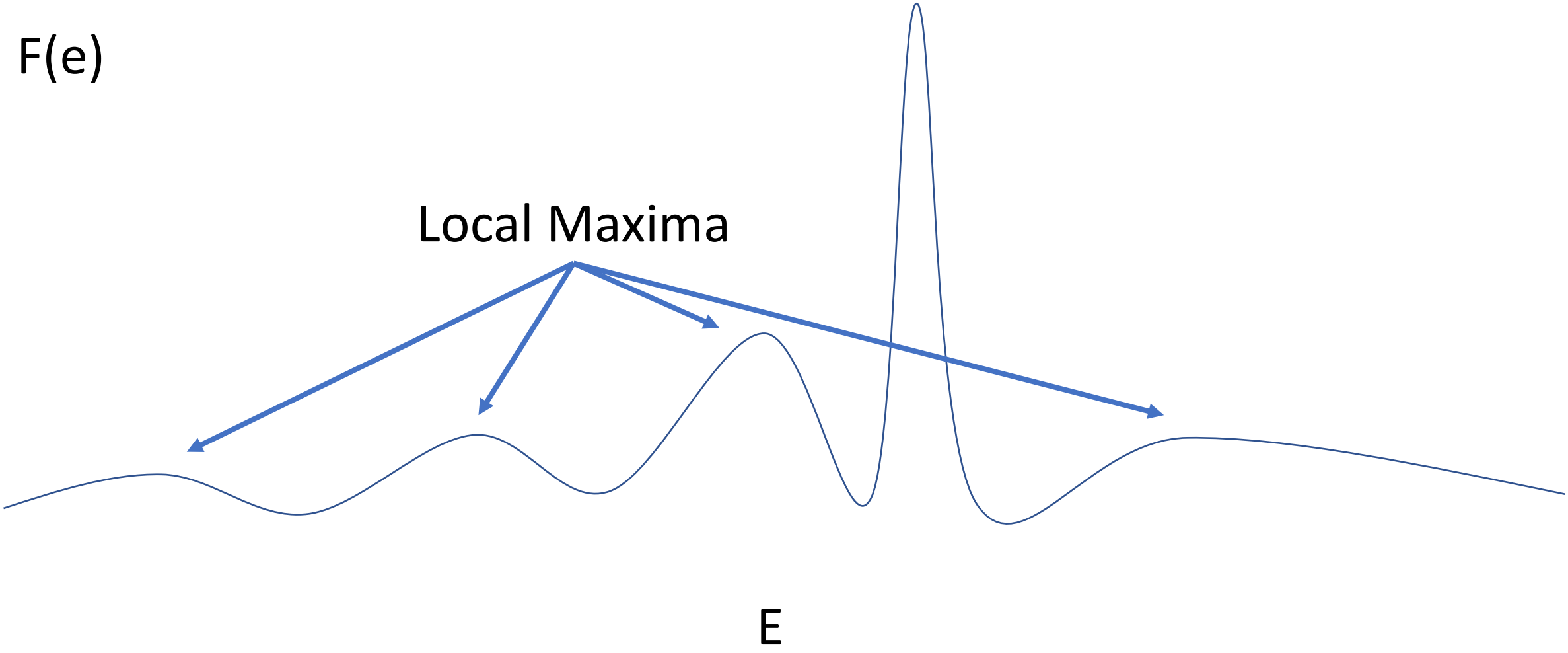
Simplest Method: Greedy/Hill Climbing Search

```
currEntity = Pick A Random Starting Position
while F(currEntity)< threshold:
    neighbors = GetNeighbors(currEntity)
    for neighbor in neighbors:
        if F(neighbor)>F(currEntity):
            currEntity = neighbor
return currEntity
```

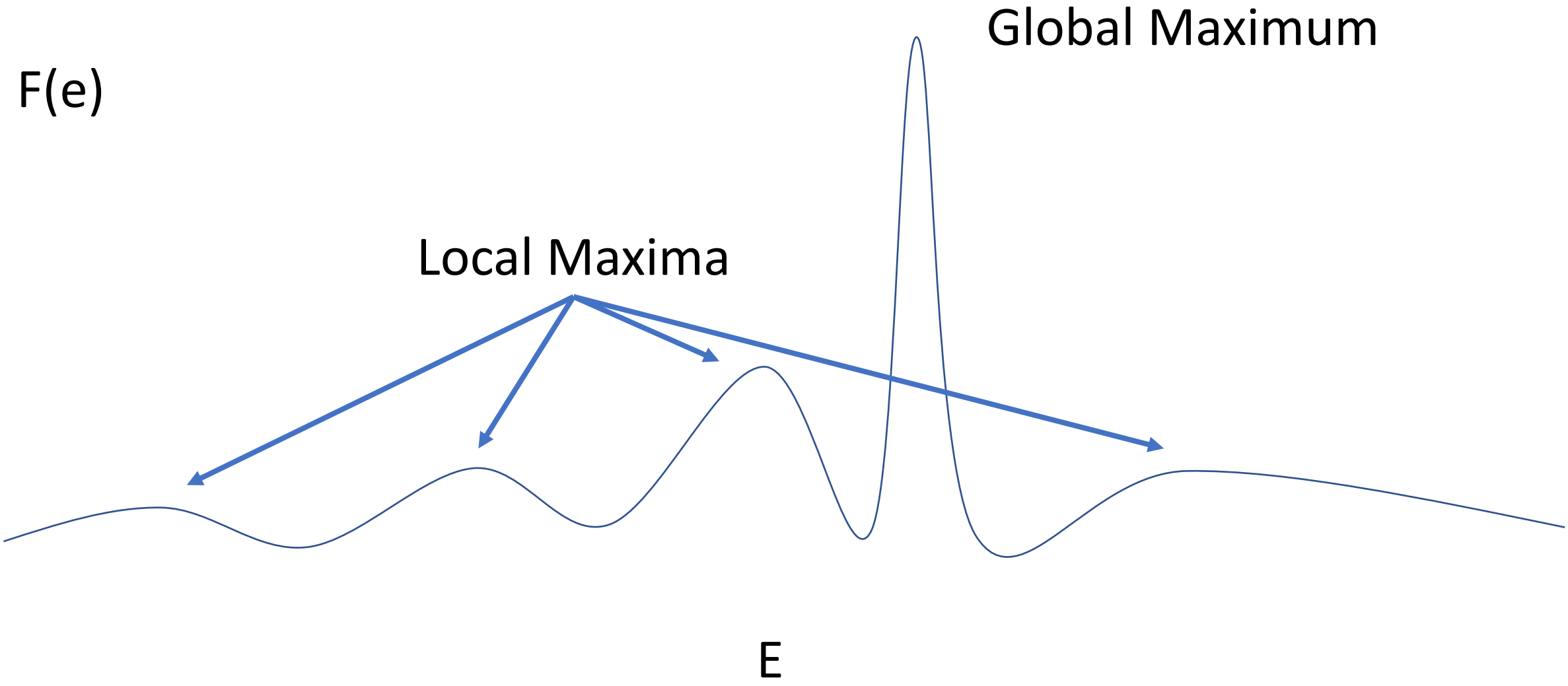
Issue 1: What if the search-space looks “spiky”?



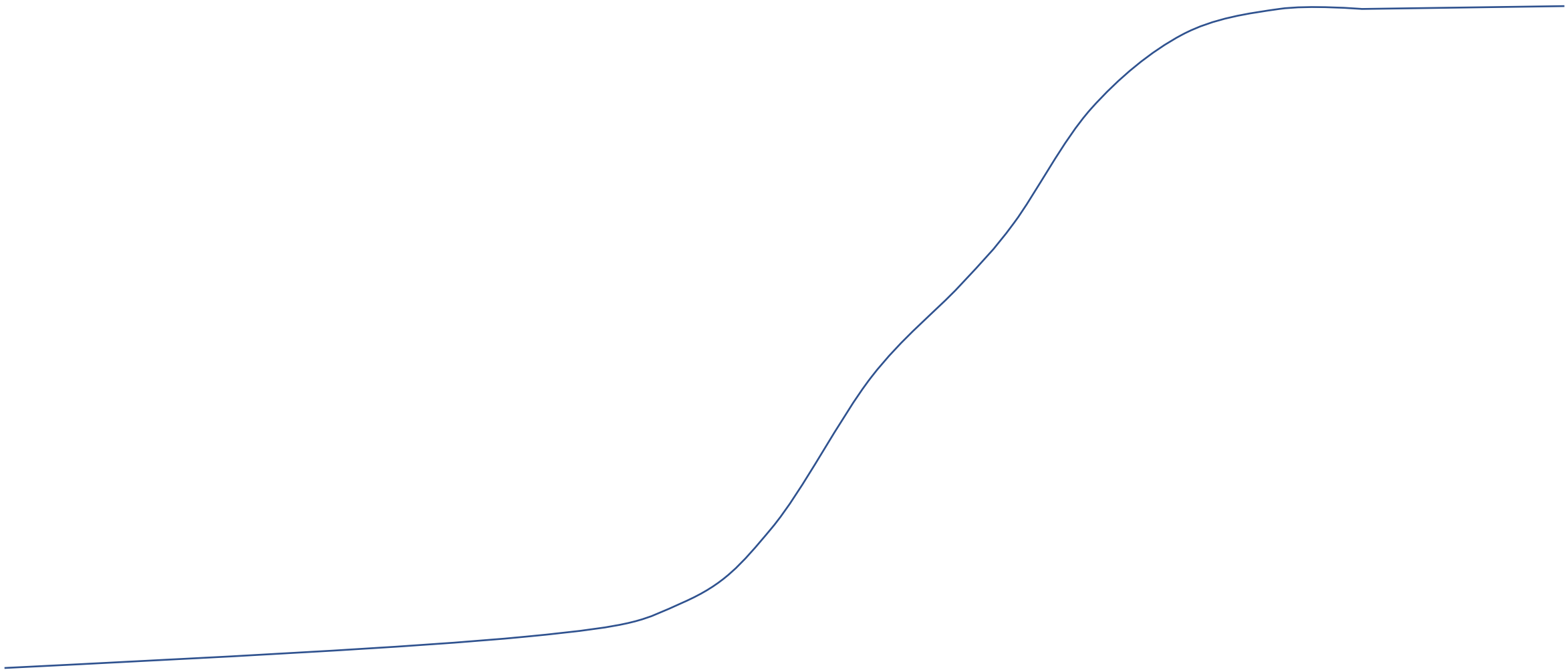
Issue 1: What if the search-space looks “spiky”?



Issue 1: What if the search-space looks “spiky”?



Issue 2: What if the search-space looks like this?



Greedy Search Pros and Cons

Pros:

- Simple
- (Depending on space) gets *A* result quickly.

Cons:

- (Depending on the space) will be unable to find a global maxima.
- If you're in a space simple enough for greedy search, why not do constructive PCG?

PQ1 https://tinyurl.com/guz-pq27a
https://forms.gle/k1VsoufQPAFSixxw8

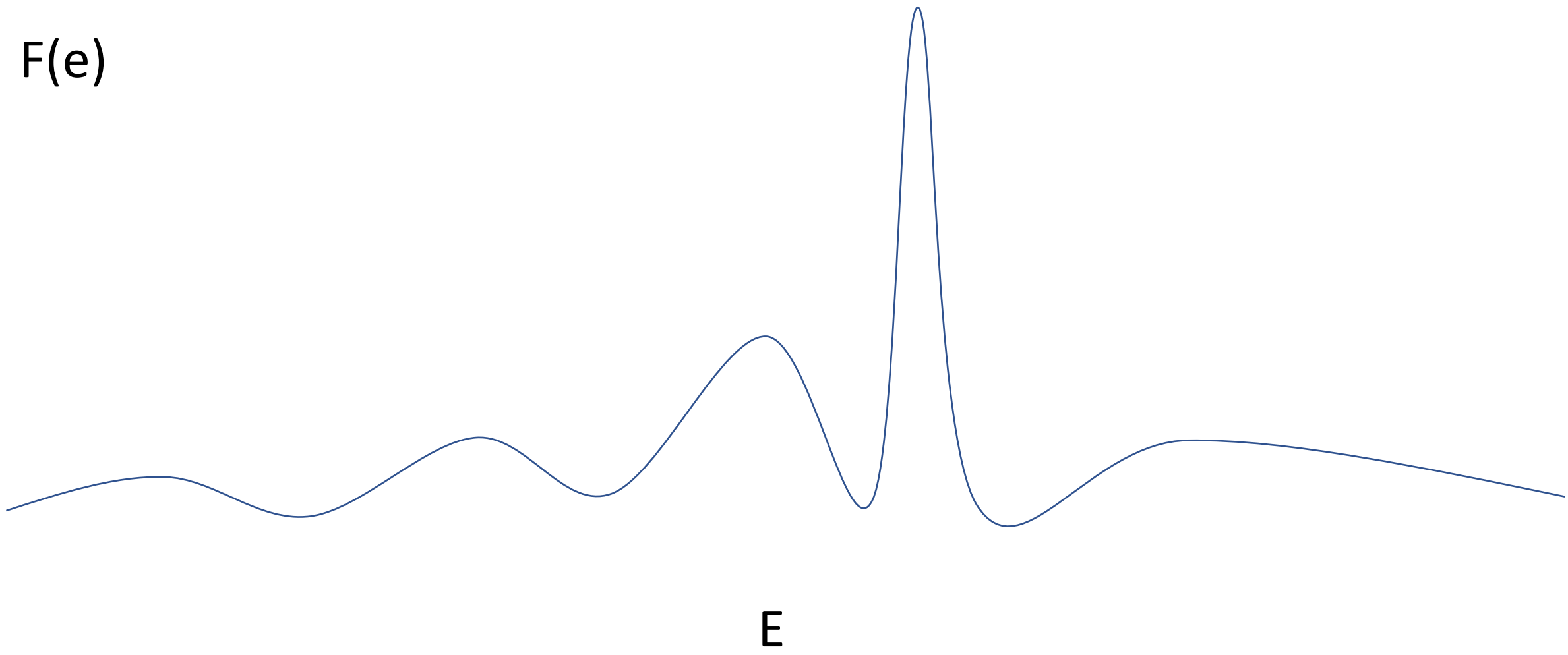
Name a kind of content (level, enemy, song, decoration, etc.). What could the Neighbor and Fitness functions look like?

My answer

- New heroes in a hero-based game
- Neighbors: modify ability, swap ability, modify ultimate, swap ultimate
- Fitness function: How different the hero is from other heroes, how balanced the hero is (determined by bot play), and to what extent the hero synergizes with existing heroes.

Unity Example

Question (for the chat): How could we get out of a local maxima?



Random Walk

- Instead of taking the next best from the neighbors, take a random neighbor.
- At the end, pick the best point we ever saw.
- Can just as easily get stuck going back and forth forever.

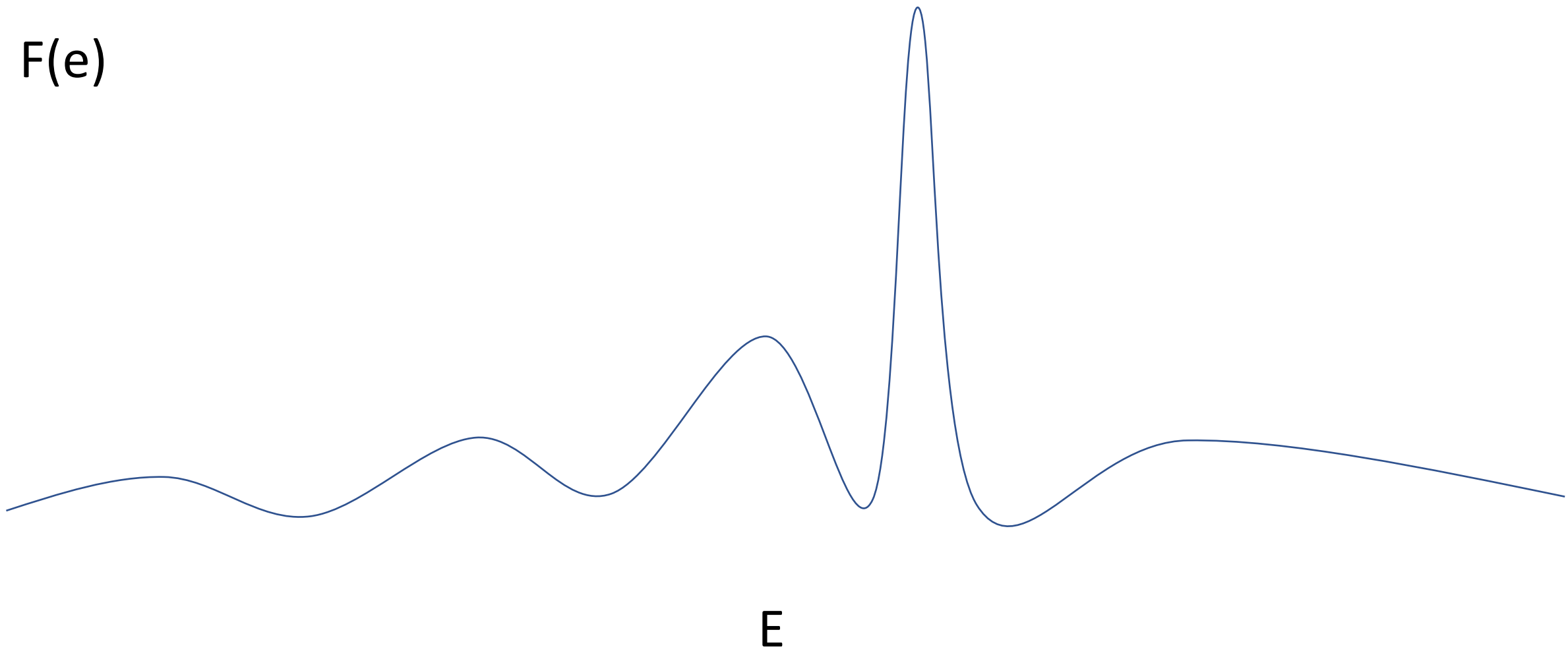
Can we combine Random Walk and Greedy Search?

We sure can! Simplest example: Simulated Annealing.

Simulated Annealing

```
currEntity = Pick A Random Starting Position
T = initialValue
while F(currEntity)< threshold:
        neighbors = GetNeighbors(currEntity)
        T-=rateOfDecay
        for neighbor in neighbors:
                if P(F(neighbor),F(currEntity), T)> Random(0,1)
                        currEntity = neighbor
return currEntity
```

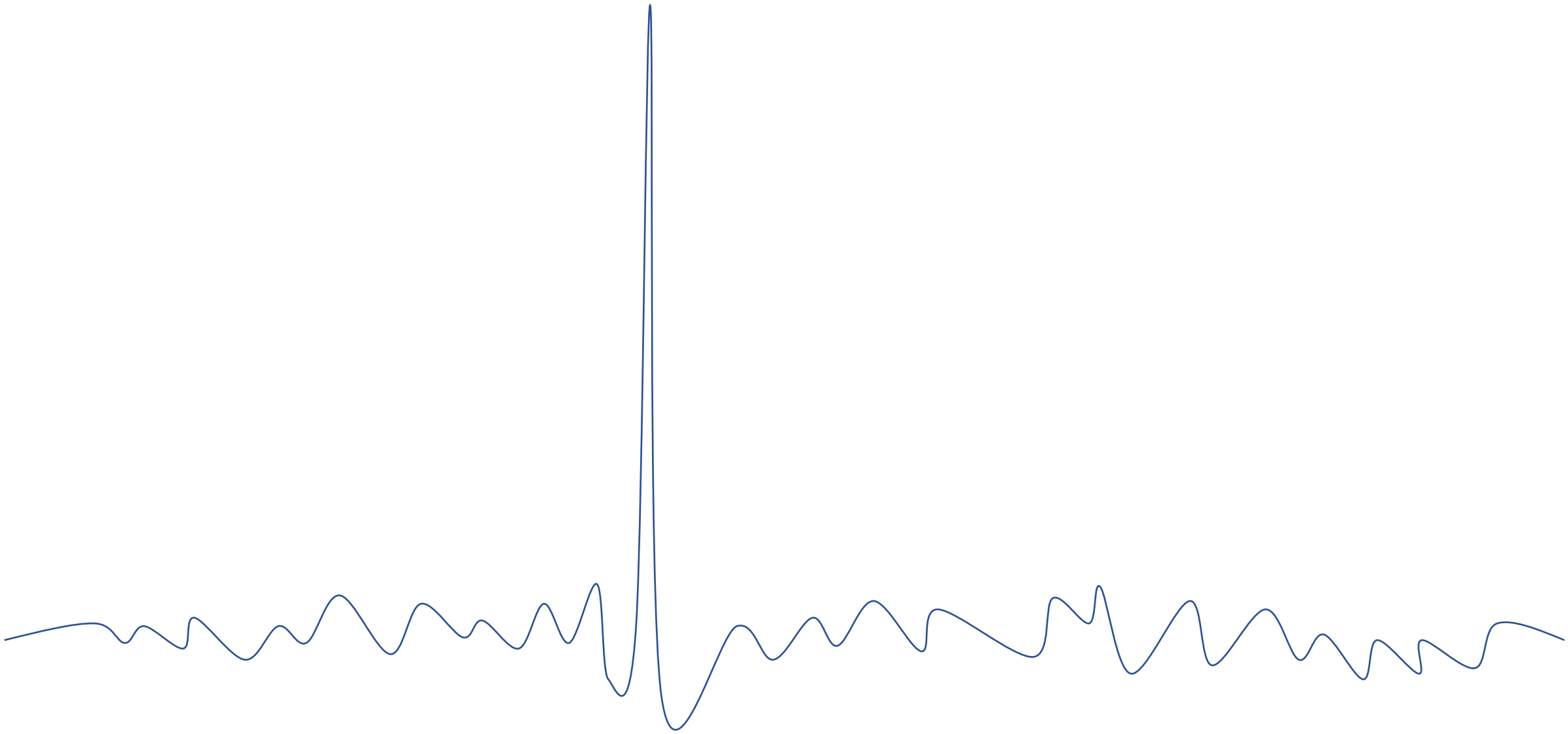
Impact



Has Simulated Annealing solved all our problems?

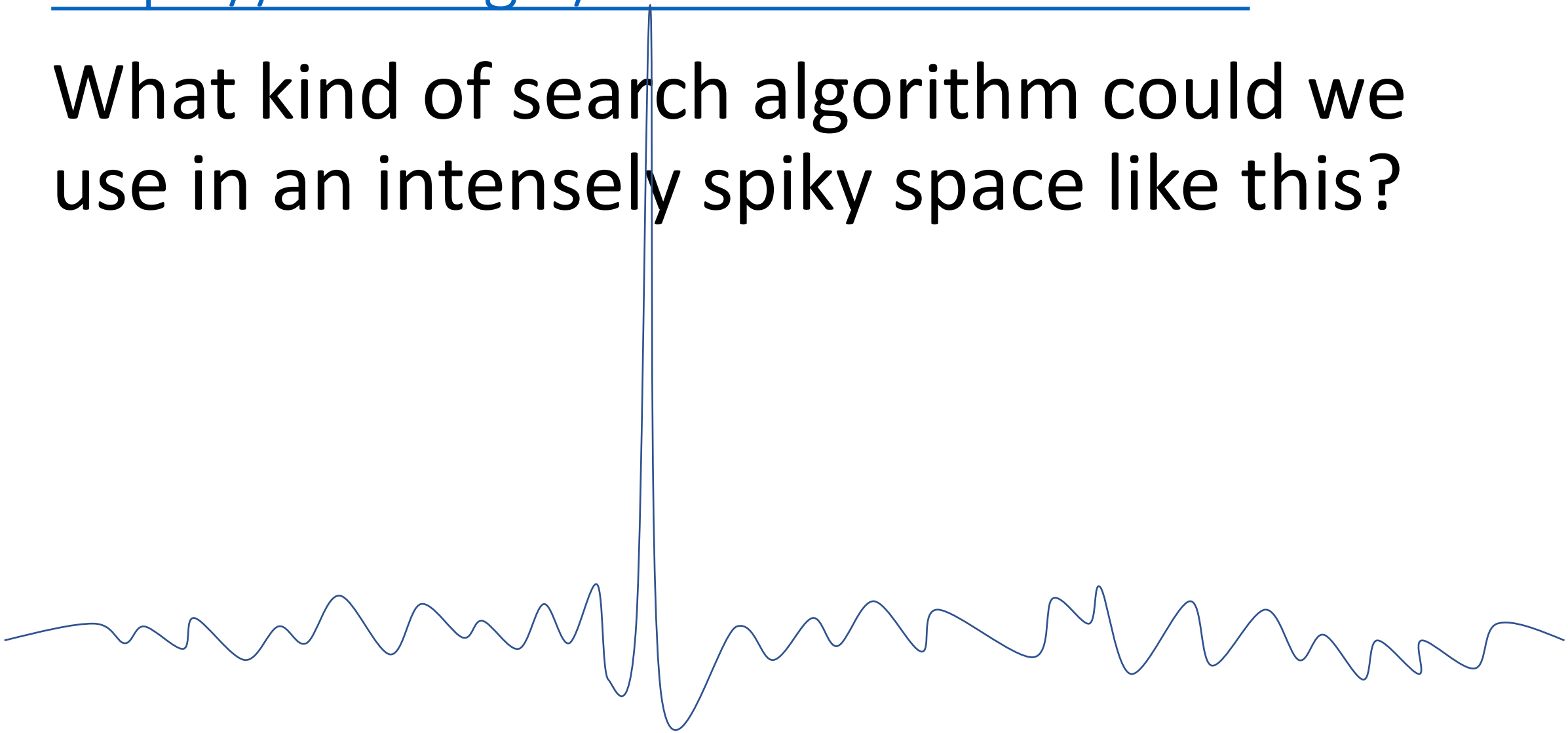
No.

Bad Space for Simulated Annealing



PQ2 https://tinyurl.com/guz-pq27b
https://forms.gle/7X9aGR3Xxnss8Bue8

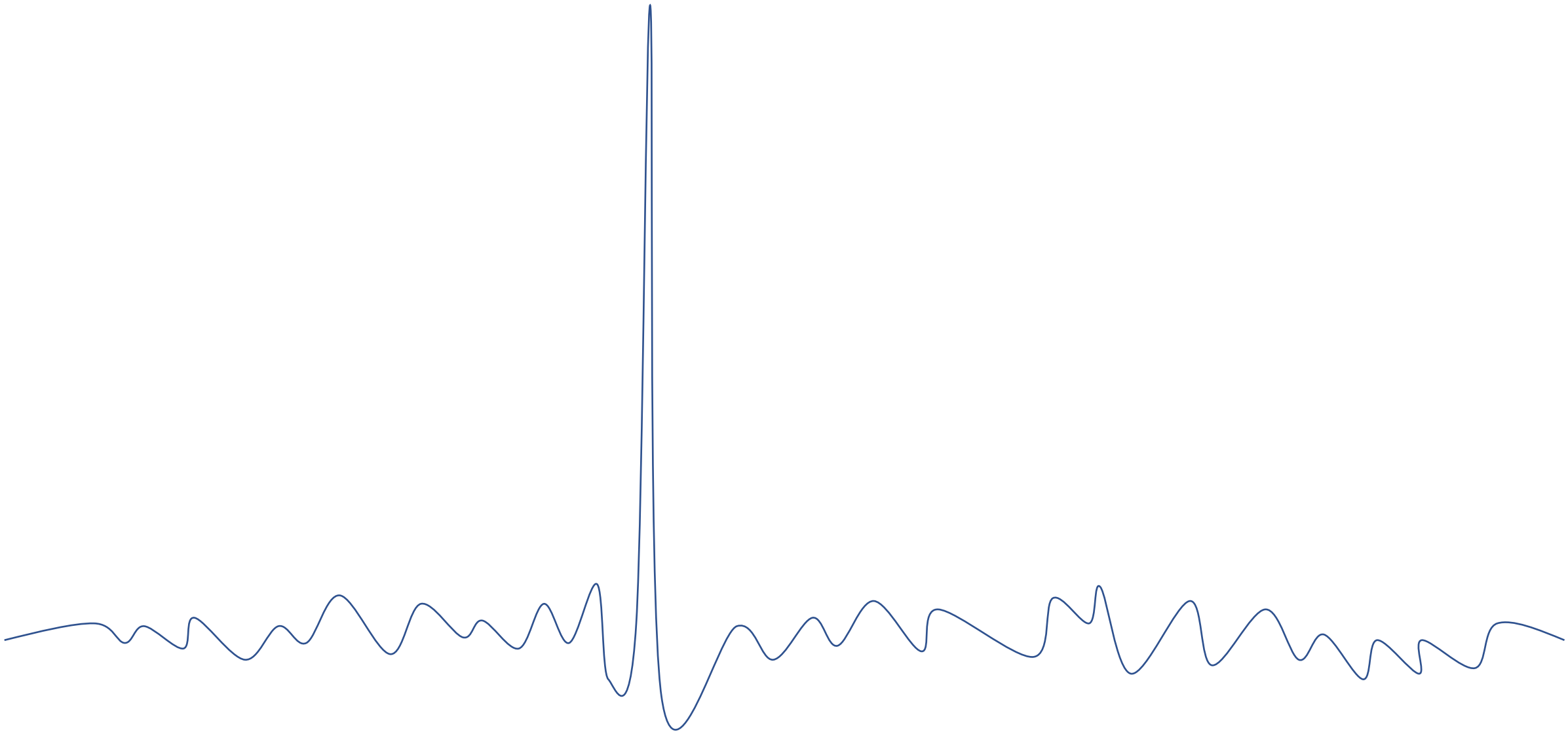
What kind of search algorithm could we use in an intensely spiky space like this?



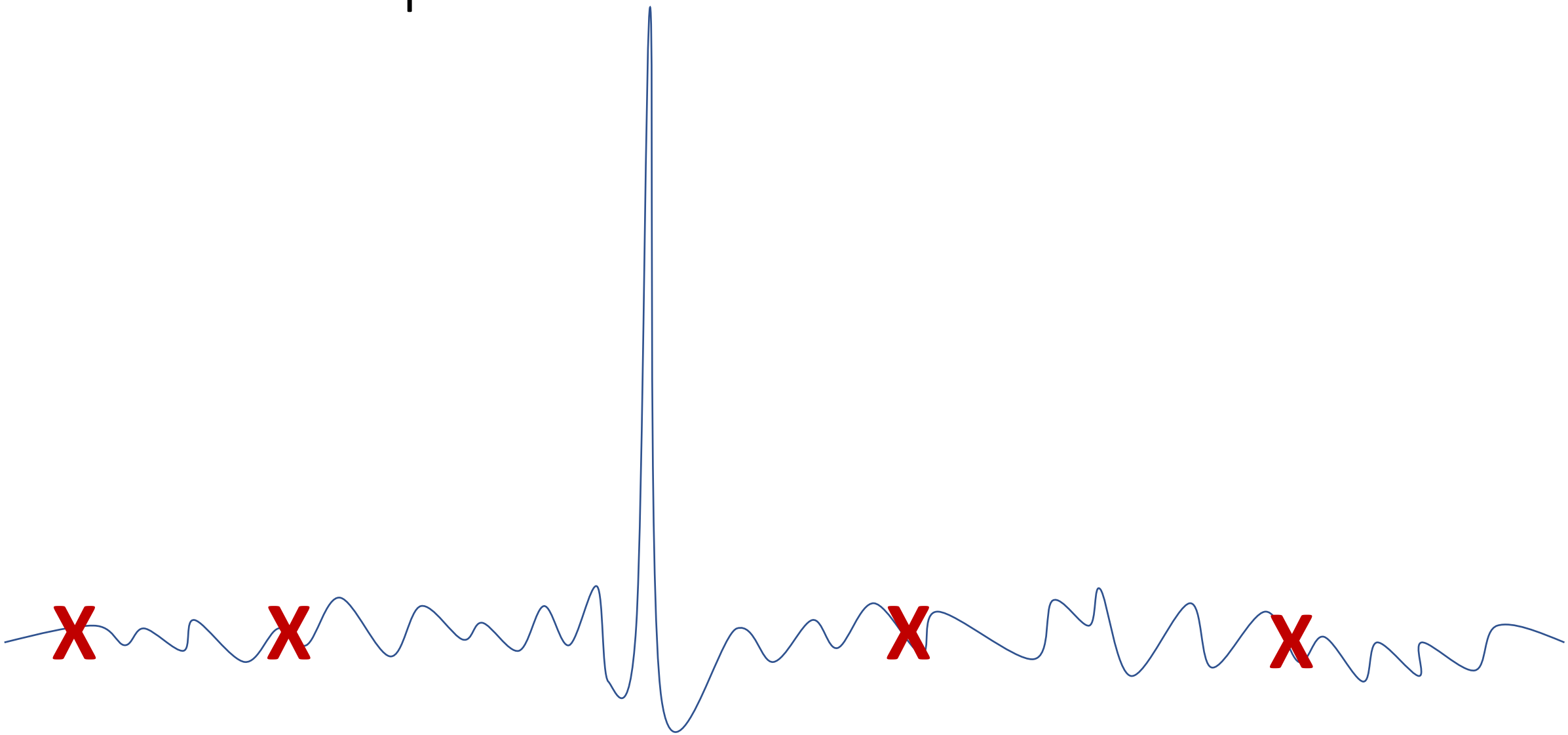
What if we didn’t just search one point at a time? What if we had a population of points?

We could take inspiration from evolution! Maybe call it an evolutionary or genetic algorithm.

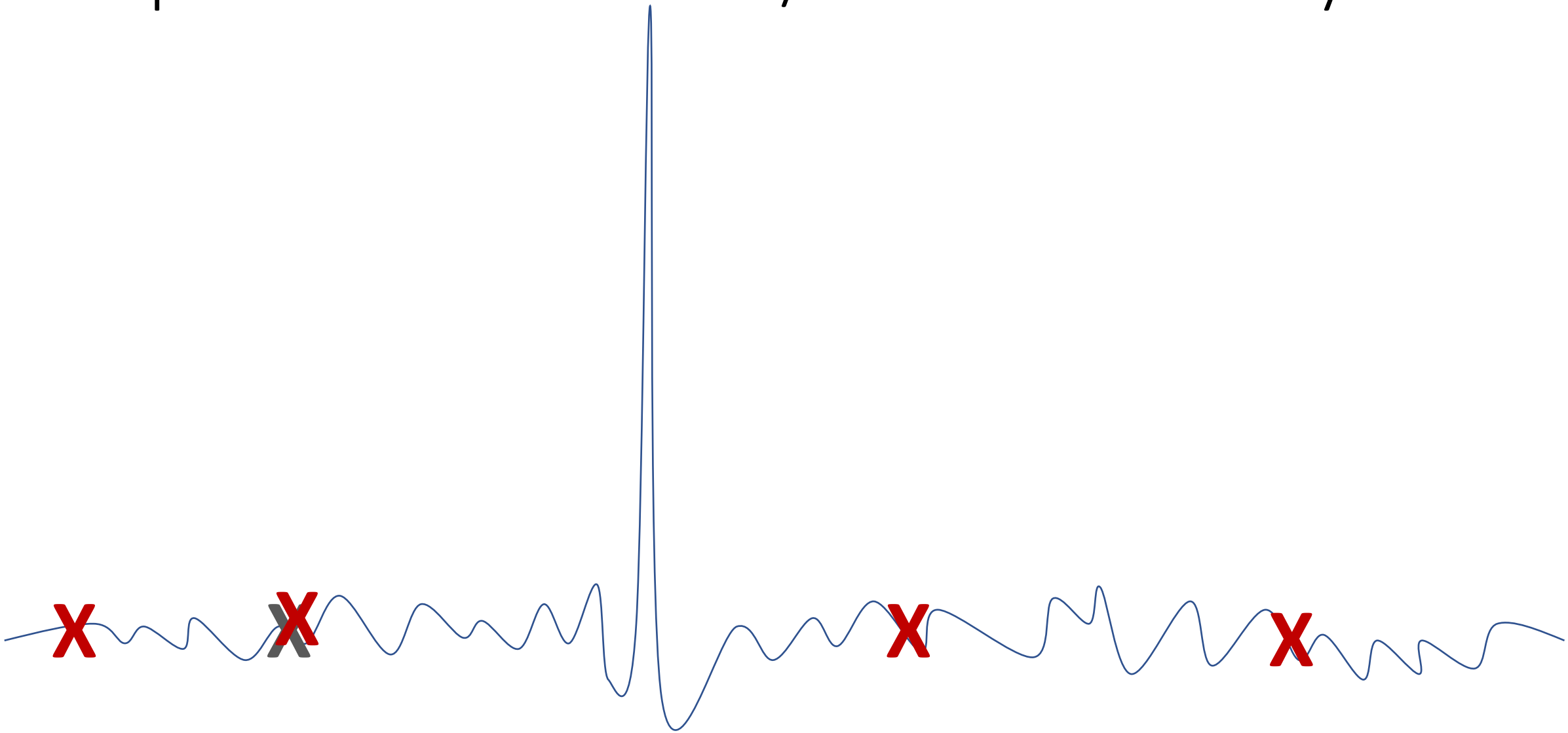
Genetic Algorithm Intuition



Genetic Algorithm Intuition: Initialize Population

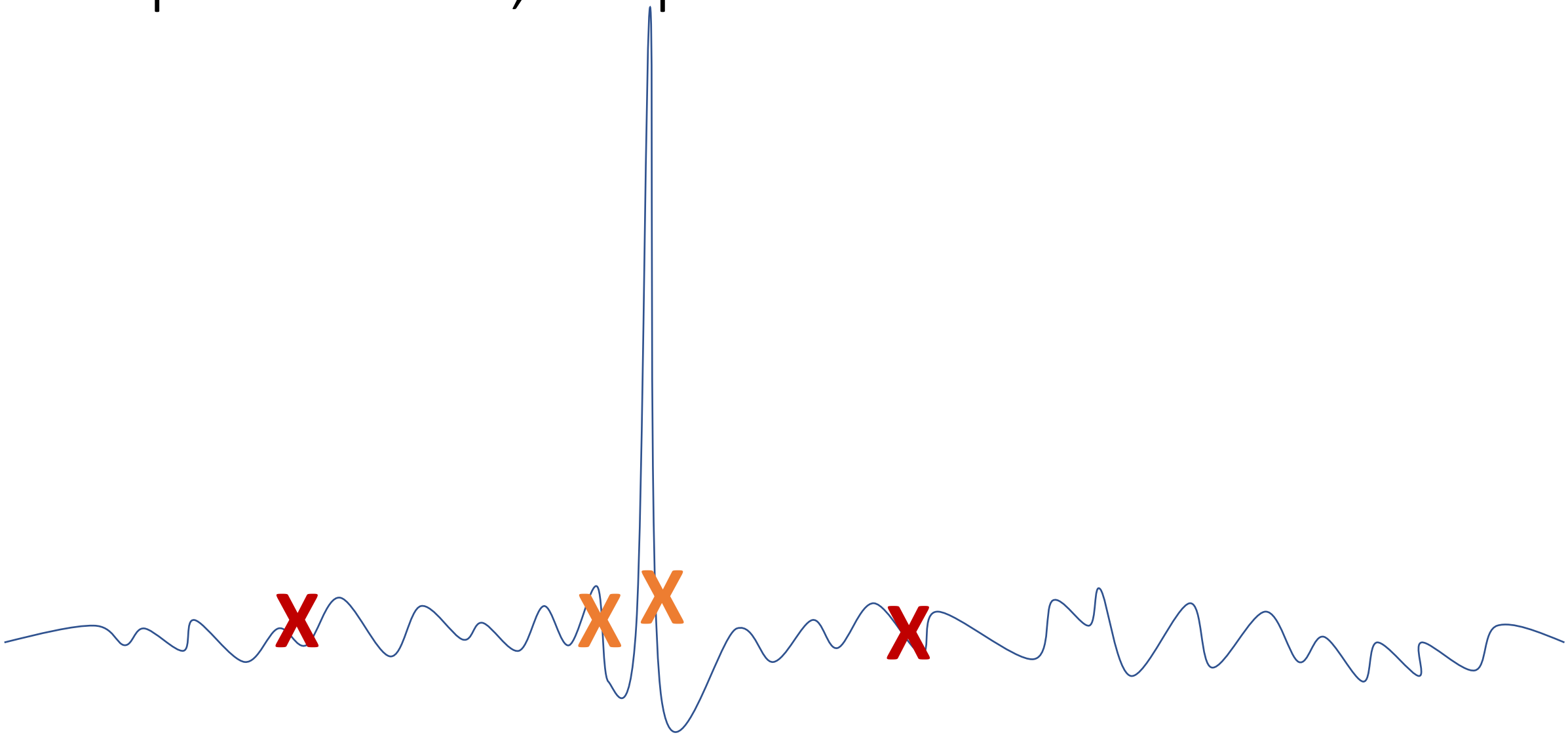


Genetic Algorithm Intuition
Step 1: Random Walk w/ Some Probability

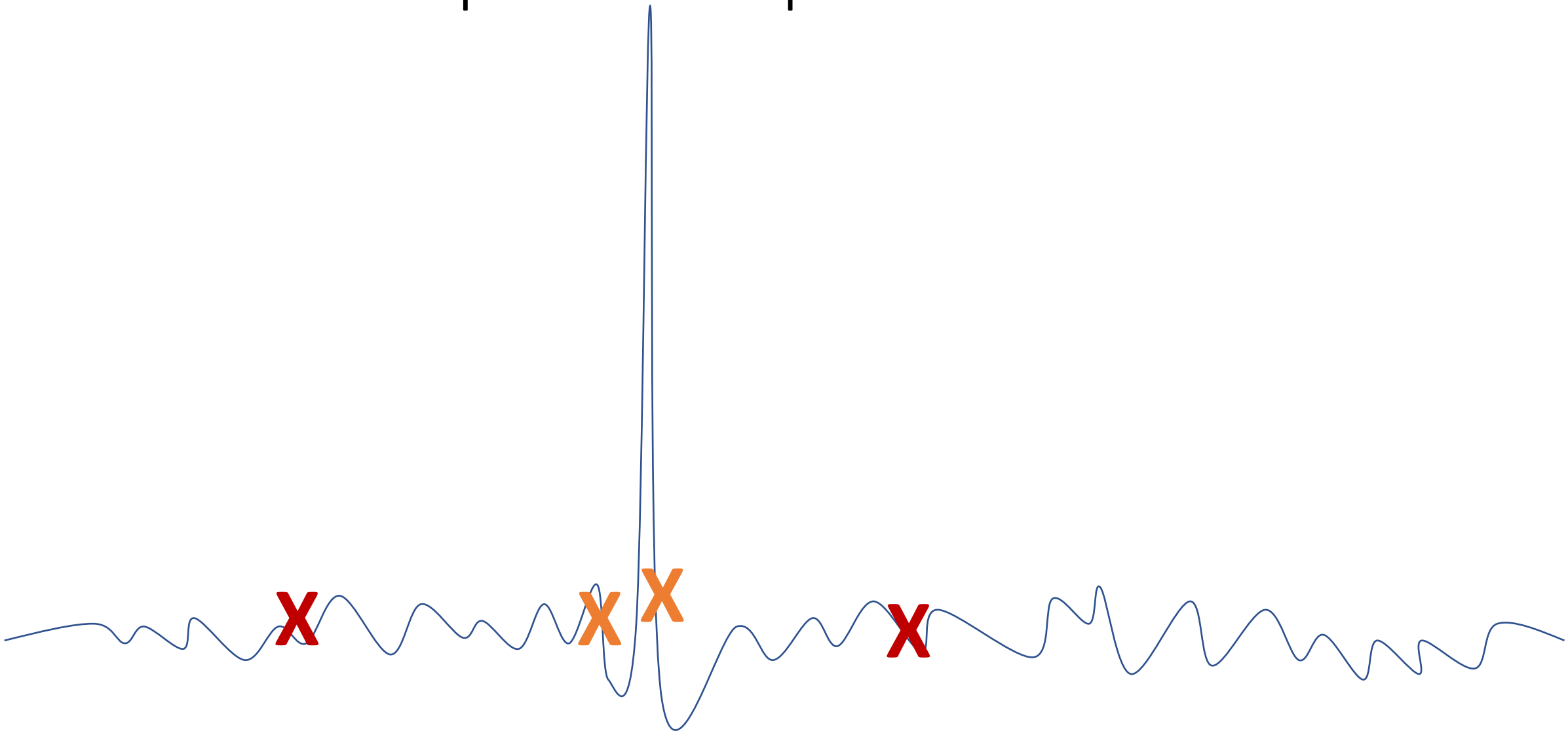


Genetic Algorithm Intuition
Step 2: Crossover (take good parents, hopefully make good big jumps in the space

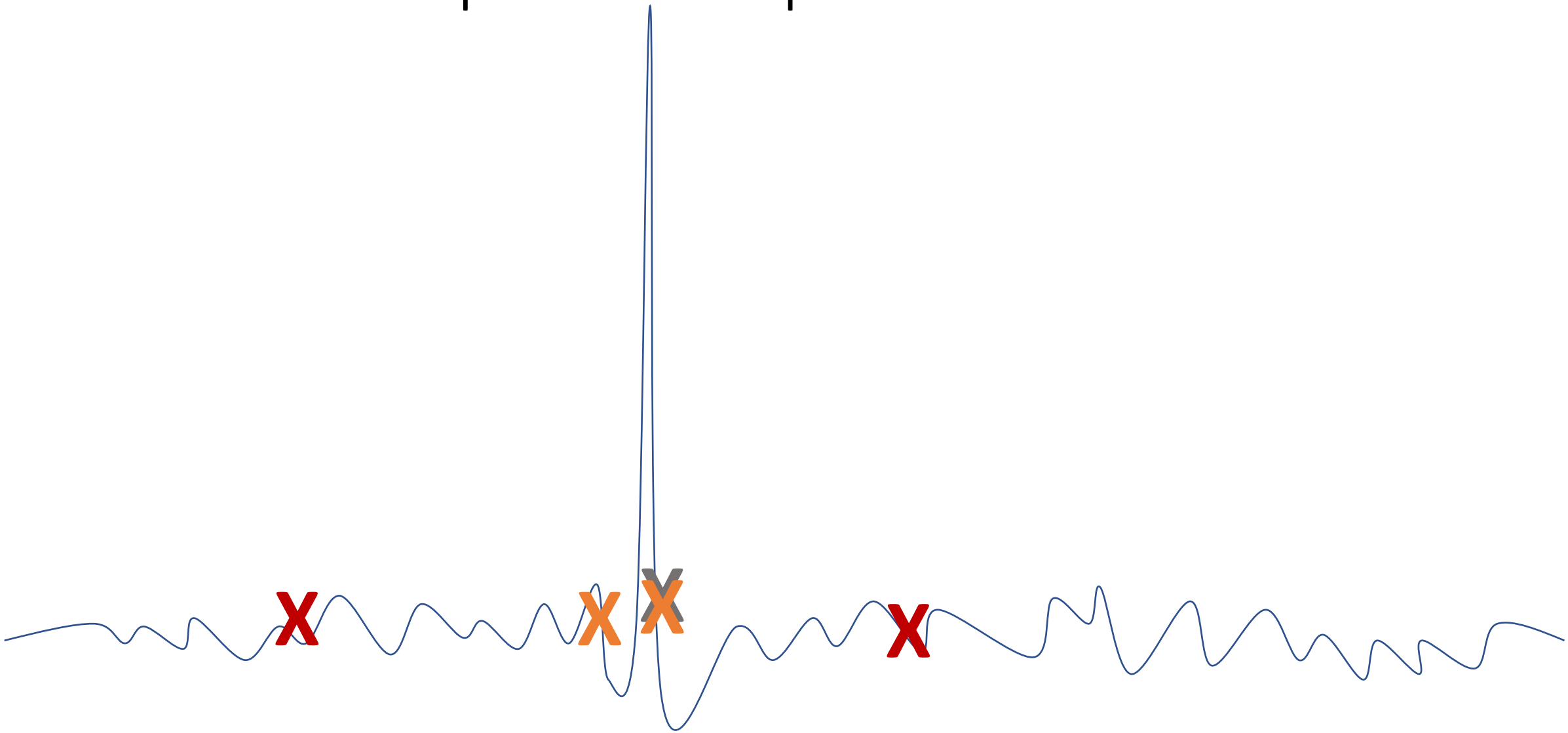
Genetic Algorithm Intuition
Step 3: Reduce, keep the best



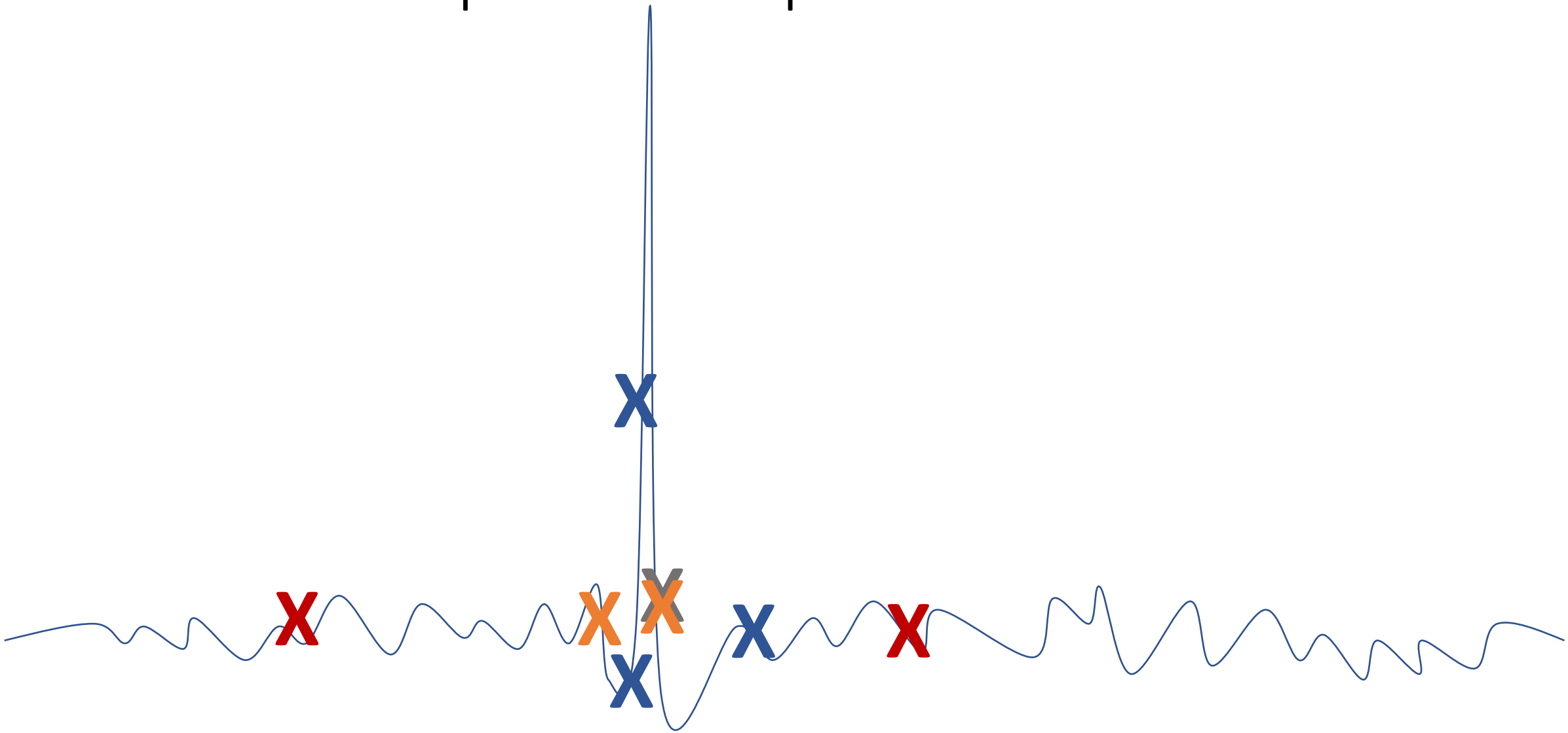
Genetic Algorithm Intuition
Return to Step 1 and Repeat Until…



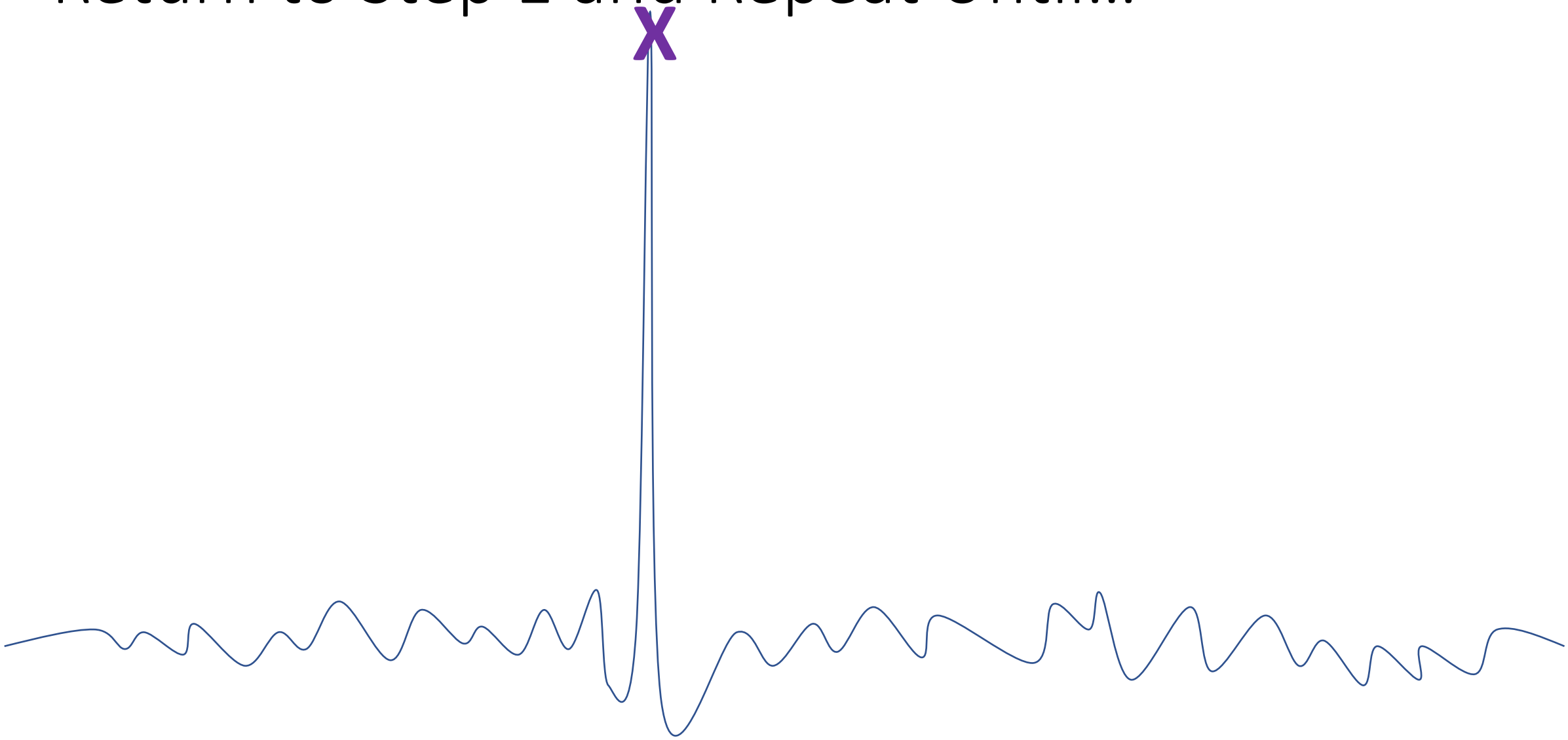
Genetic Algorithm Intuition
Return to Step 1 and Repeat Until…



Genetic Algorithm Intuition
Return to Step 1 and Repeat Until…



Genetic Algorithm Intuition
Return to Step 1 and Repeat Until…



Genetic Algorithm Pseudocode

```
population = initialize populationSize random points
time = 0
while avg or best (F(population))<threshold and time<maxTime:
        time++
        MutatePopulation(population)
        population += CrossoverPopulation(population)
        population = Reduce(population, populationSize)
return best (F(population))
```

Mutate Population

- With some probability (mutation rate), randomly swap out a member with a neighbor

- Has all the benefits of a random walk.

Crossover Population

- Inspired by reproduction
- Sample pairs from our population based on fitness and “mix” their traits.
- Intuitively allows us to make big “jumps” in the space, that we hope are good.

Reduce Population

- Reduce our population back down to the initial size, taking only the best populationSize members
- (Some approaches only take children)
- Has all the benefits of Greedy search

Genetic Algorithms: The Most Popular Search-based PCG Approach, …but still not great!

Galactic Arms Race

Darwin's Demons

Petalz

http://ccg.doc.gold.ac.uk/wp-content/uploads/2016/03/togelius_tciaig12.pdf

Wednesday: More Constructive PCG! + Voting on Future of Game AI Topics