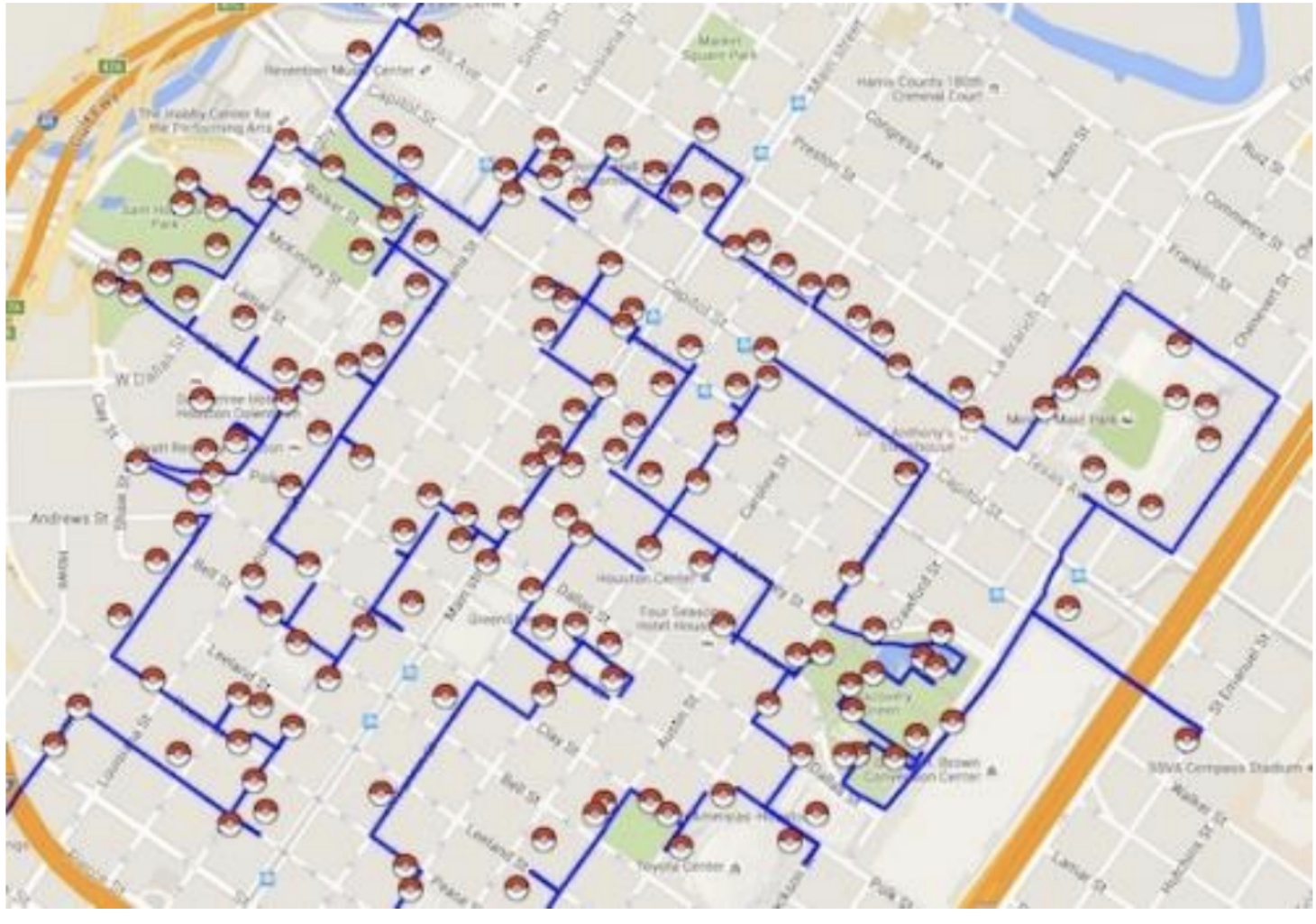
Non-differentiable Optimization

Traveling Salesman Problem (TSP, **задача коммивояжера**)

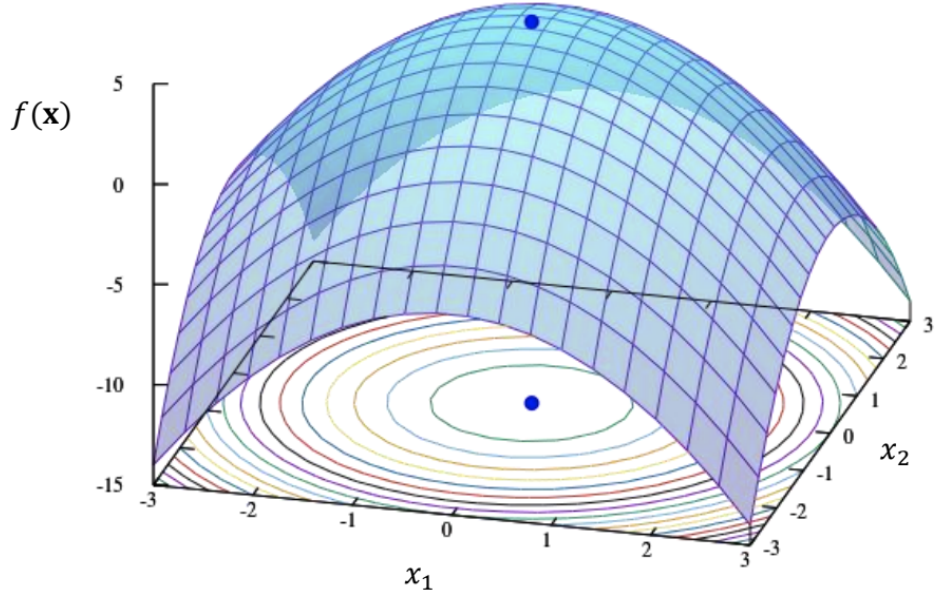


Stochastic optimization

$$\mathbf{x}^* = \text{argmax}(f(\mathbf{x}))$$

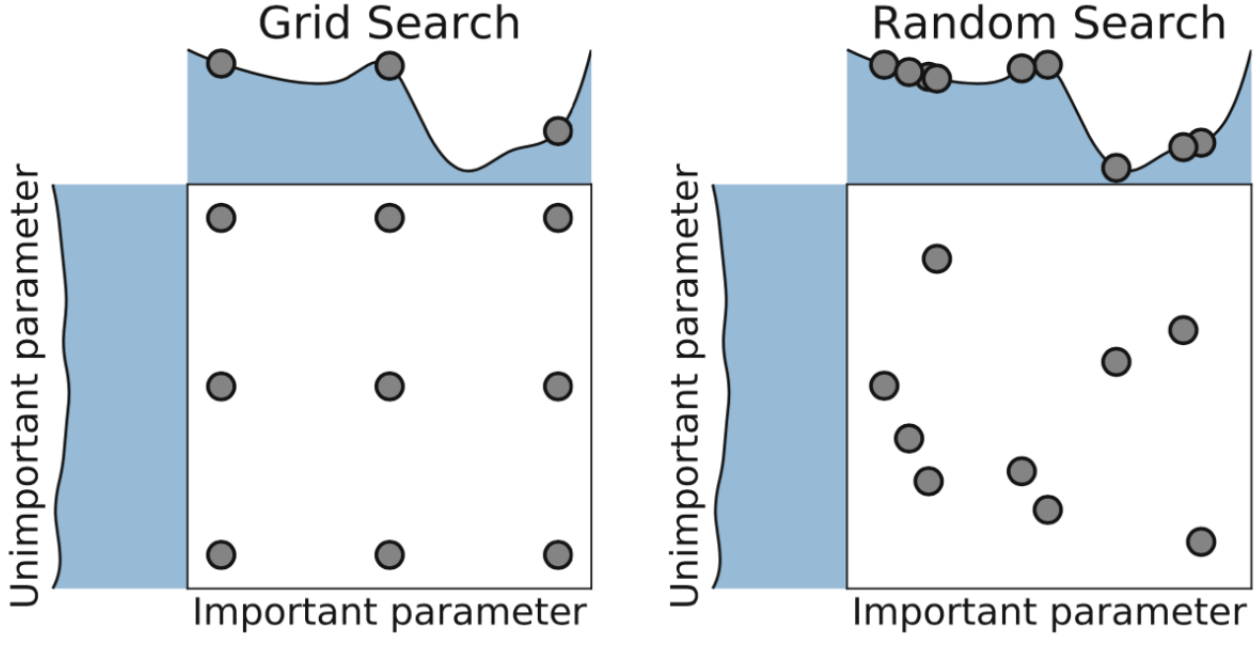
For differentiable loss functions – gradient descent.

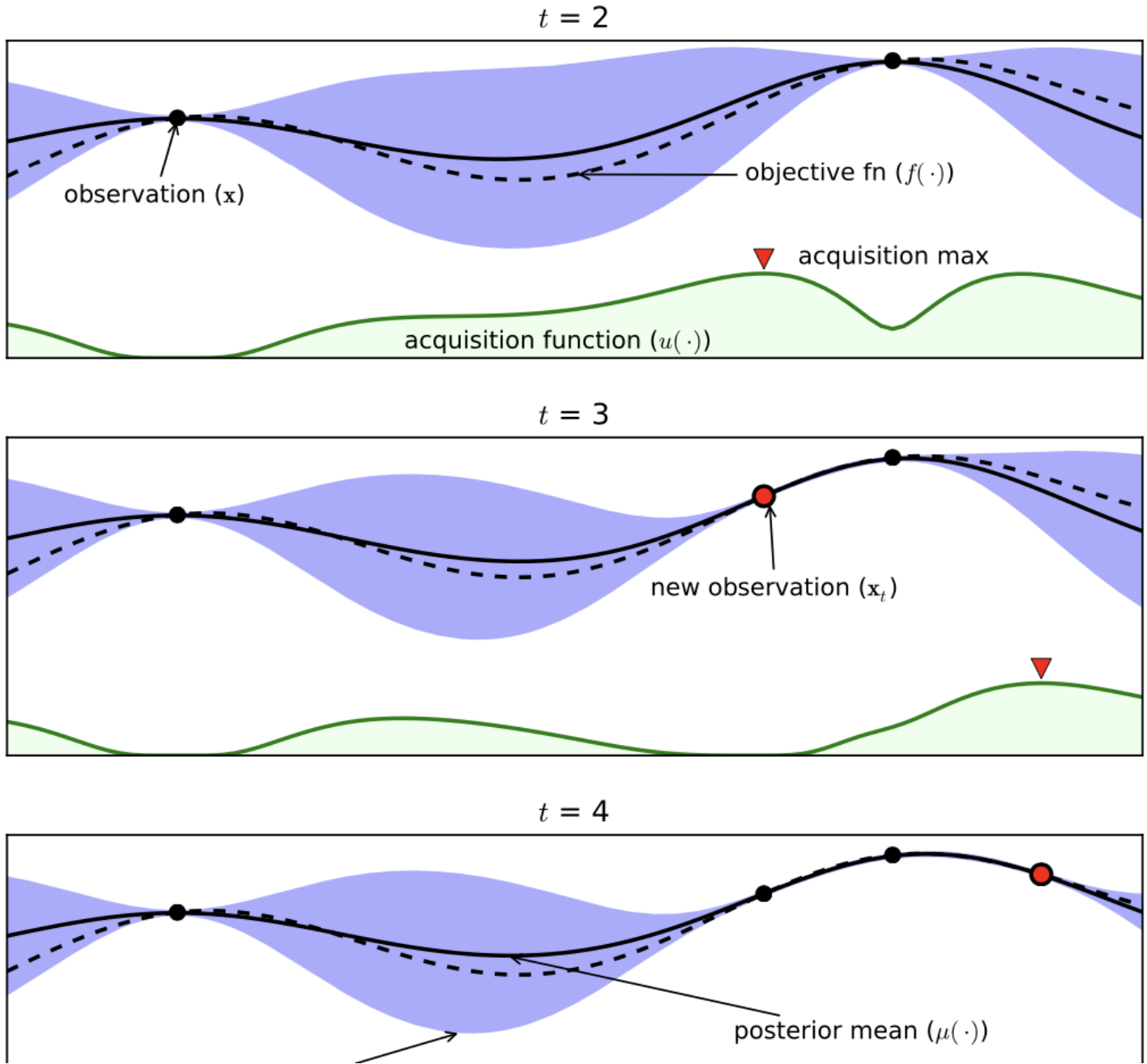
For not differentiable of unknown loss functions – local and global search.



Hyperparameter optimization

- Random Search
- Grid Search
- Bayesin Optimization


Grid Search
Unimportant parameter
Important parameter
Random Search
Unimportant parameter
Important parameter


$t = 2$
observation ($\mathbf{x}$)
objective fn ($f(\cdot)$)
acquisition max
acquisition function ($u(\cdot)$)
$t = 3$
new observation ($\mathbf{x}_t$)
$t = 4$
posterior mean ($\mu(\cdot)$)

Cross-entropy method

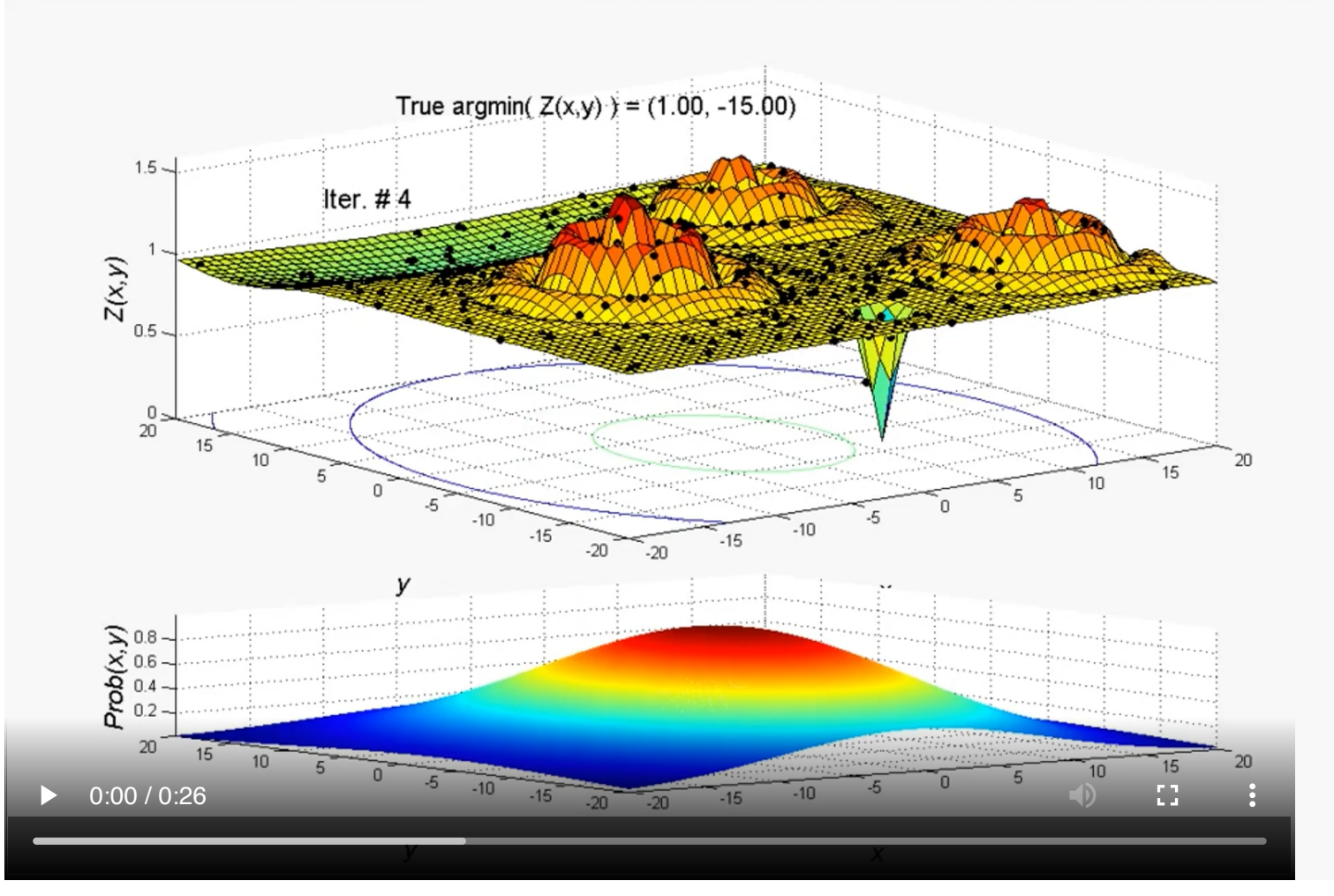
- At step t=1 choose random parameter-vector θ_0
- Sample points $x_1, \ldots, x_n$ from $p(x, \theta_{t-1})$
- $\theta_t = \arg\max_u \frac{1}{k} \sum_{x_i \in top-k} \frac{p(x_i,u)}{p(x_i,\theta_{t-1})} \log$

  $p(x_i, \theta_{t-1})$

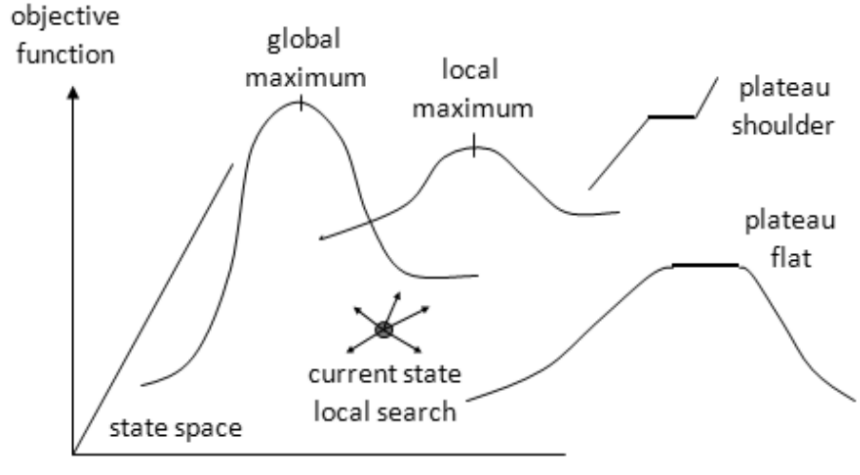
In [2]:

```
Video("imgs/op1.mp4")
```

Out[2]:



Hill climb



Similar to **Gradient Descent** method.

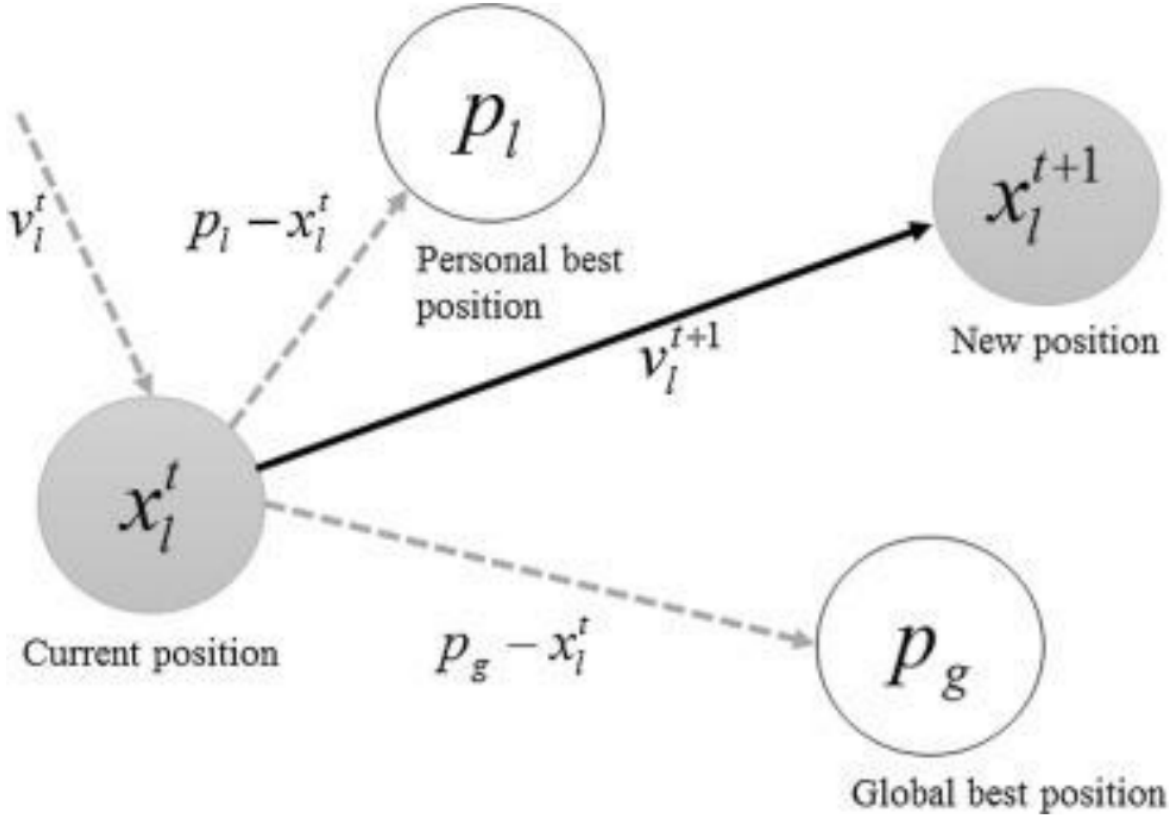
Calculate metric for possible shifts and step in a direction of the best value.

Stochastic Hill Climbing:
Step with probability proportional to increase in the metric (for example, with softmax).

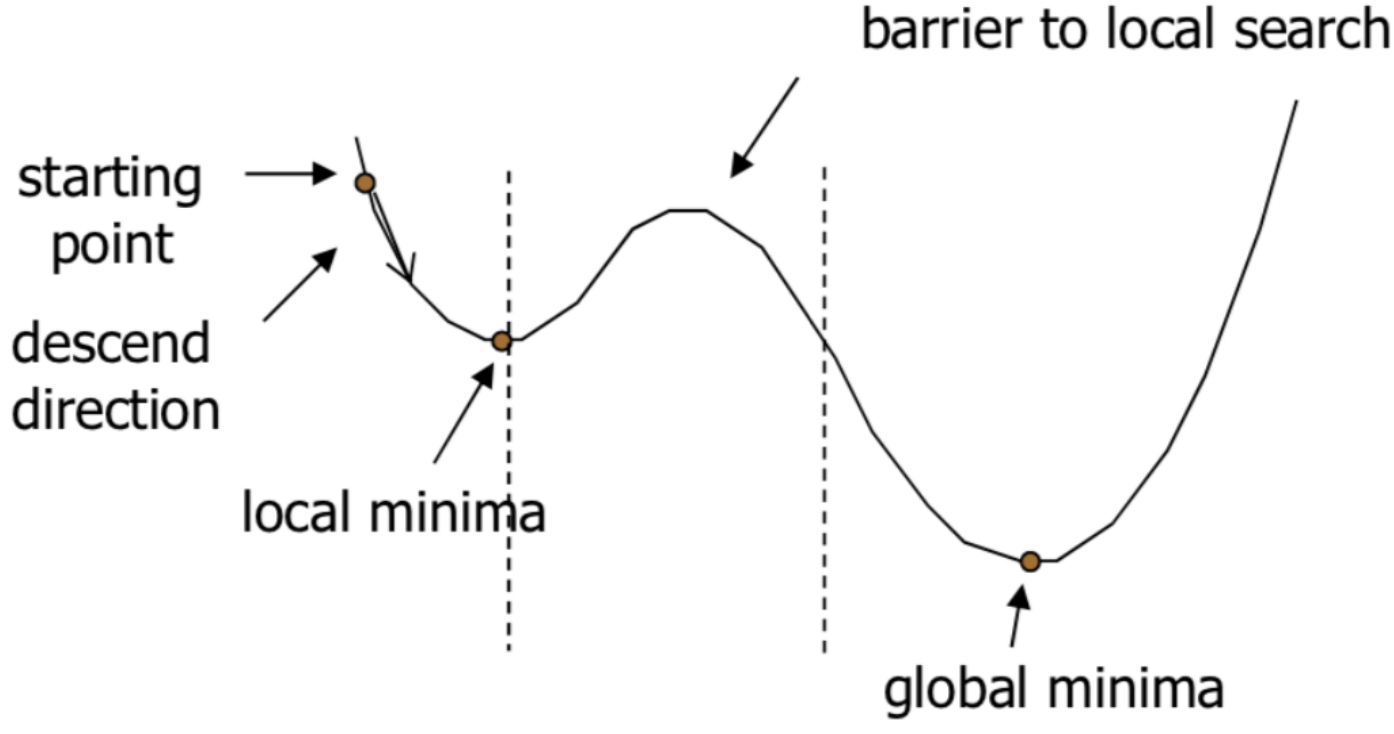
Taboo Search:
Do not return to visited states.

Particle swarm optimization:
Multiple “climbers” that pass information.

Particle swarm optimization



Simulated Annealing



Simulated Annealing

Introduce temperature to stochastic hill climb.

For Softmax:

$$P(s_i) = \frac{e^{\frac{\Delta E(s_i)}{T}}}{\sum e^{\frac{\Delta E(s_j)}{T}}}$$

	T - temperature			
	10000	10	1	0.1
ΔE	P(ΔE)	P(ΔE)	P(ΔE)	P(ΔE)
10	0.25016	0.43944	0.99325	1
5	0.25004	0.26653	0.00669	1.9287E-22
0	0.24991	0.16166	4.51E-05	3.7200E-44
-2	0.24986	0.13235	6.1E-06	7.6676E-53

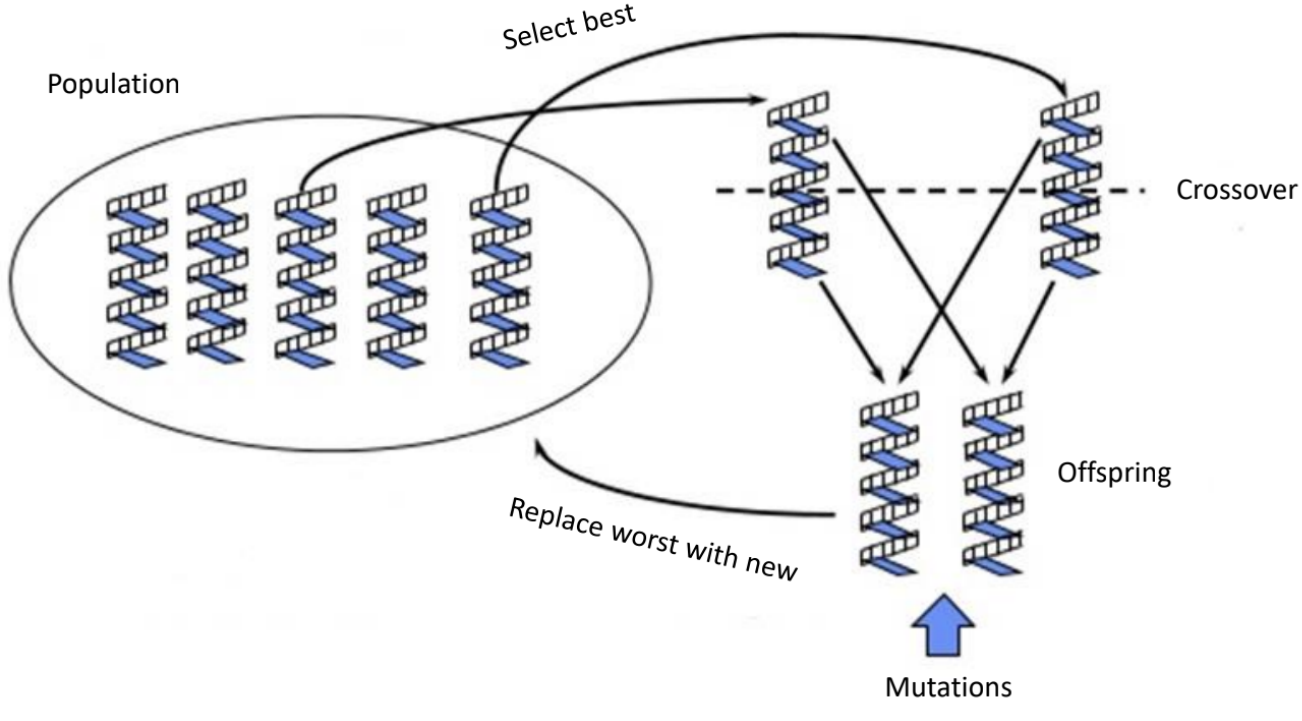
In [7]:

```
Image(filename='imgs/op2.gif',  width = 1200, height = 600)
```

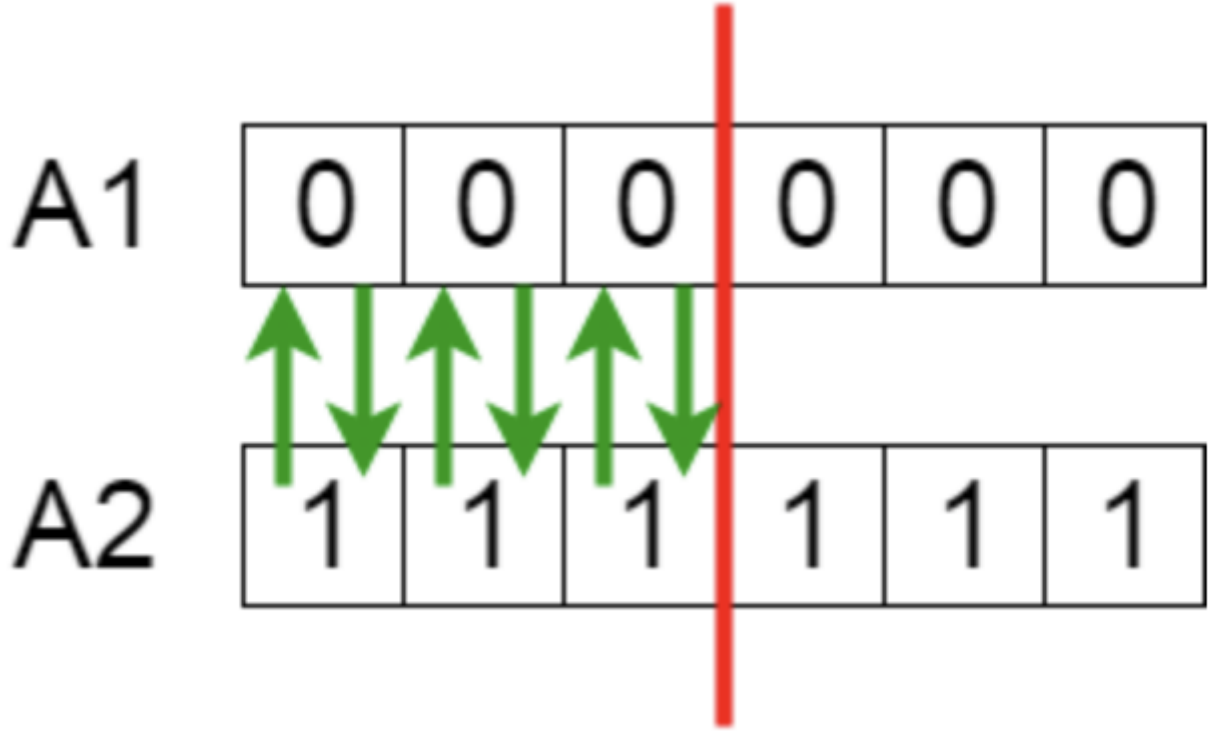
Out[7]:

```
<IPython.core.display.Image object>
```

Genetic Algorithm



- Initial population
- Fitness function
- Survival of the fittest (selection)
- Breeding (crossover)
- Mutation


A1
0 0 0 0 0 0
A2
1 1 1 1 1 1