Solving Optimization Problems with MATLAB

Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization

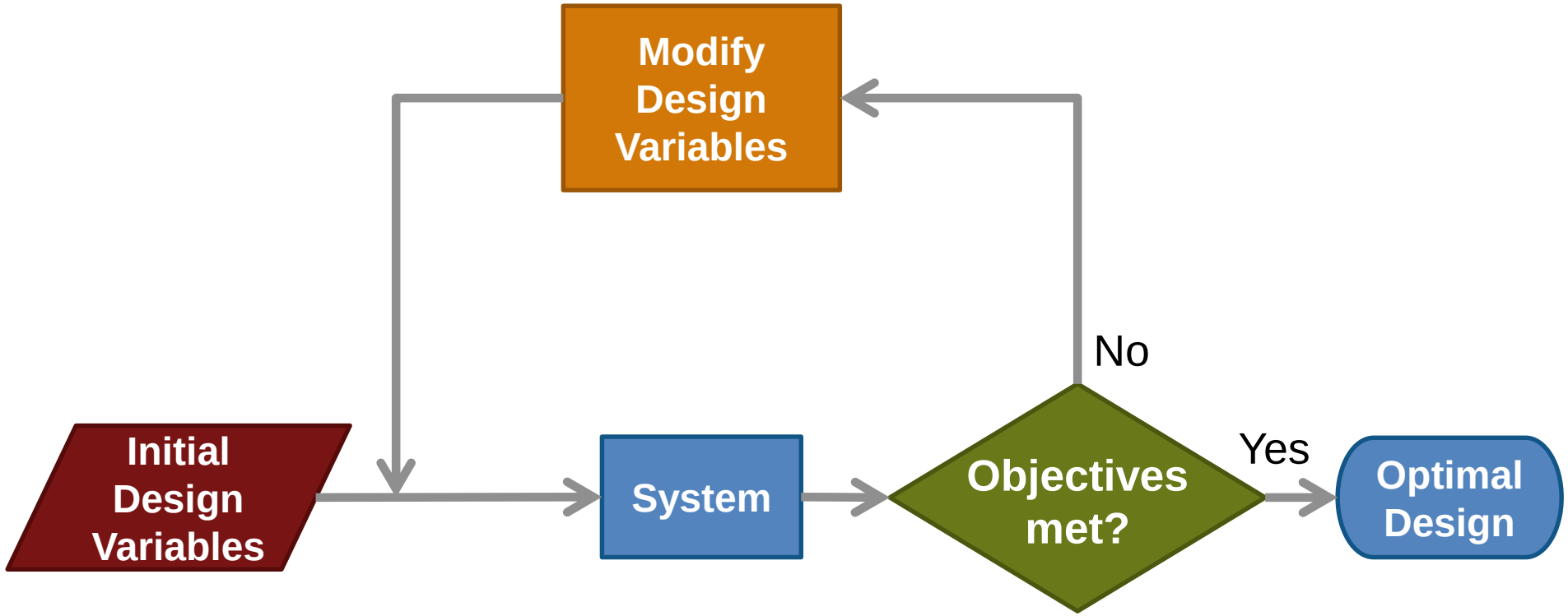
Optimization Problems

Maximize Fuel Efficiency

Minimize Risk

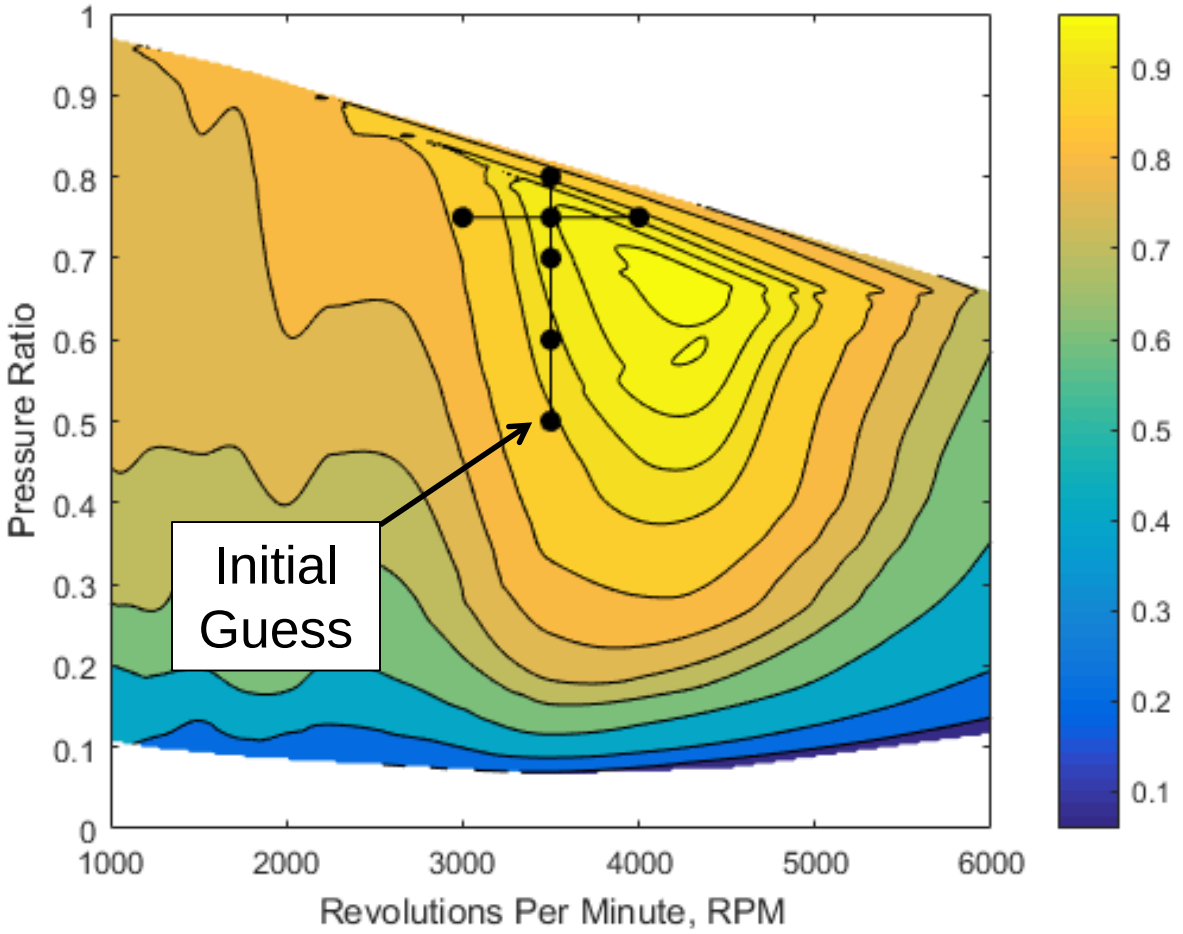
Maximize Profits

Design Process



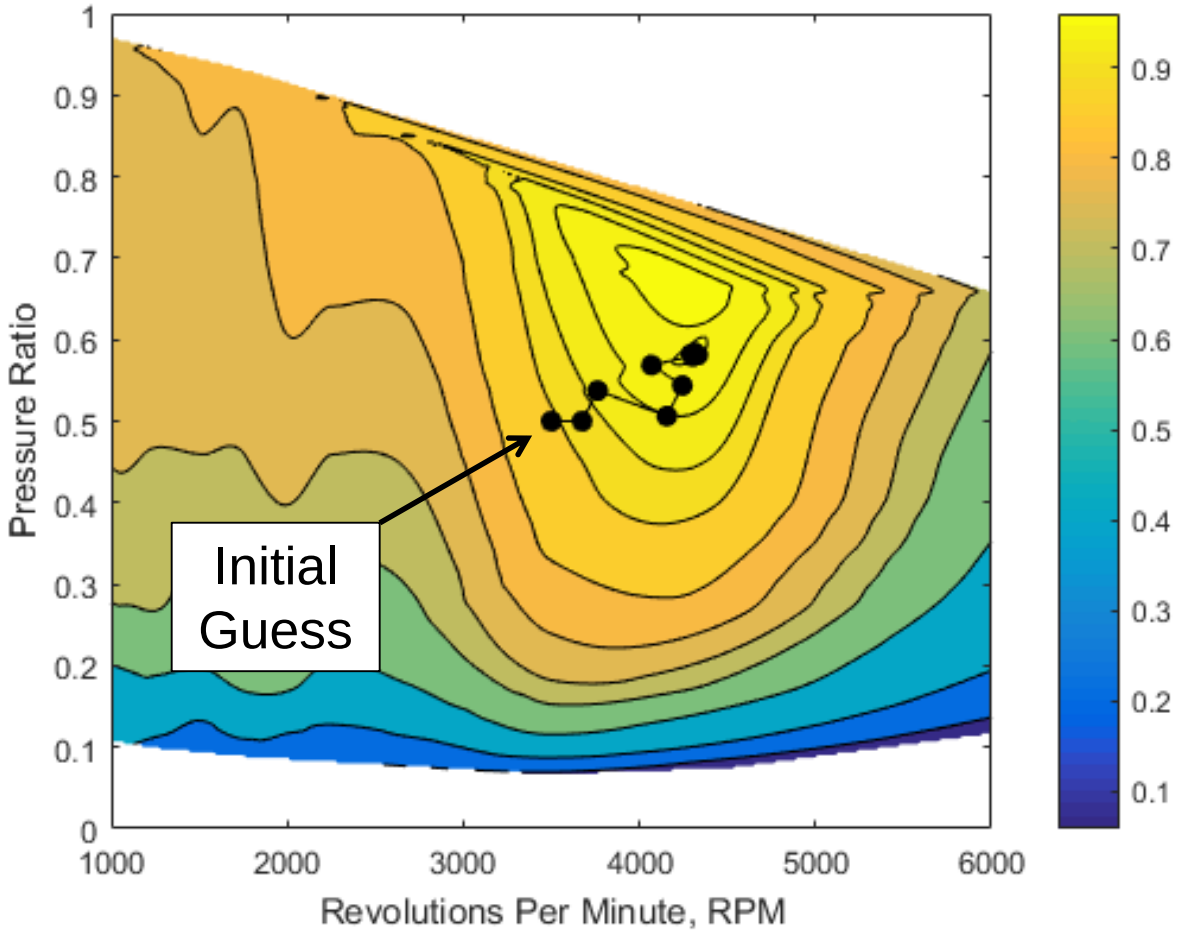
Why use Optimization?

Manually (trial-and-error or iteratively)



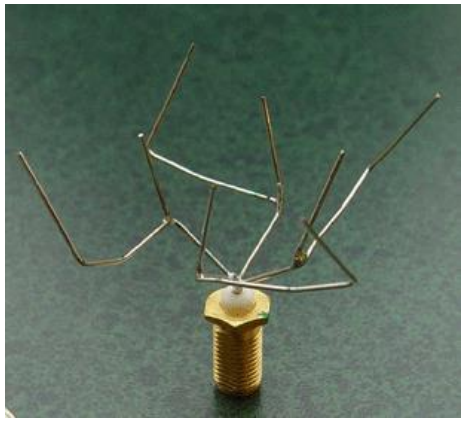
Why use Optimization?

Automatically (using optimization techniques)



Why use Optimization?

- Finding better (optimal) designs and decisions
- Faster design and decision evaluations
- Automate routine decisions
- Useful for trade-off analysis
- Non-intuitive designs may be found



Antenna Design Using Genetic Algorithm
http://ic.arc.nasa.gov/projects/esg/research/antenna.htm

Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization

Curve Fitting Demo

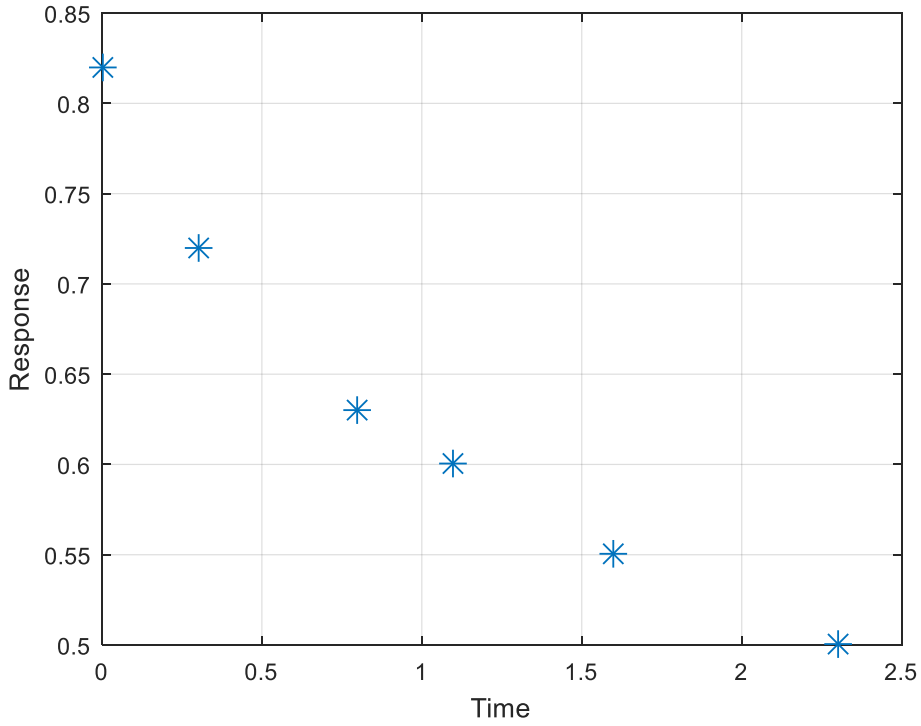
Given some data:

```
t = [0   .3   .8   1.1   1.6   2.3];

y = [.82   .72   .63   .60   .55   .50];
```

Fit a curve of the form:

$$y(t) = c_1 + c_2 e^{-t}$$



How to solve?

As a linear system of equations:

$$y(t) = c_1 + c_2 e^{-t}$$

$$y = \begin{bmatrix} 1 & e^{-t} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = Ec$$

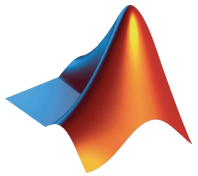
$$\begin{bmatrix} 0.82 \\ 0.72 \\ 0.63 \\ 0.60 \\ 0.55 \\ 0.50 \end{bmatrix} = \begin{bmatrix} 1 & e^{-0} \\ 1 & e^{-0.3} \\ 1 & e^{-0.8} \\ 1 & e^{-1.1} \\ 1 & e^{-1.6} \\ 1 & e^{-2.3} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Can't solve this exactly
(6 eqns, 2 unknowns)

$$y = Ec$$

$$\min_c \|Ec - y\|_2^2$$

An optimization problem!

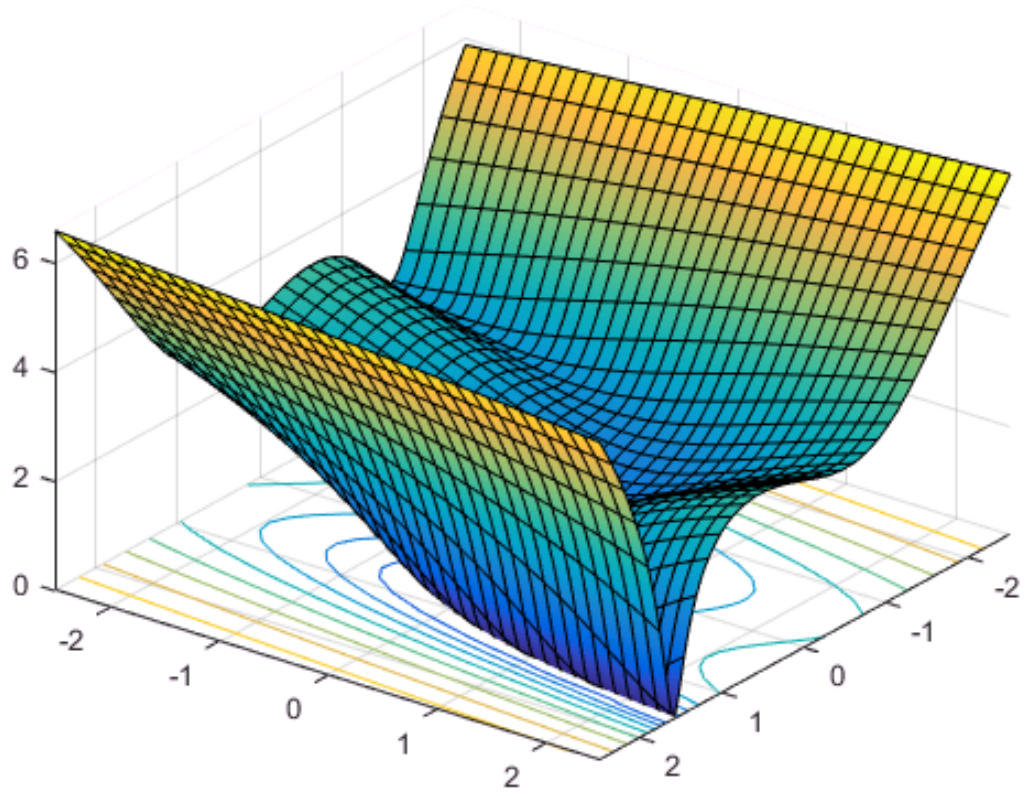


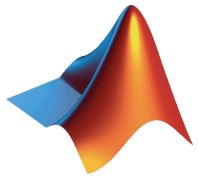
Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization

Nonlinear Optimization

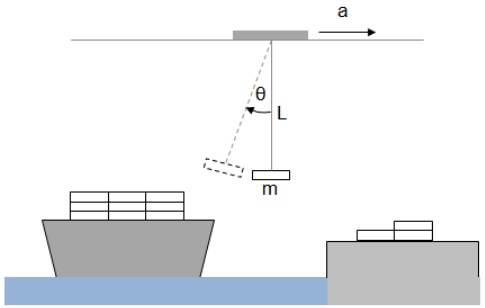
$$\min_x \quad \log\left(1 + \left(x_1 - \frac{4}{3}\right)^2 + 3\left(x_1 + x_2 - {x_1}^3\right)^2\right)$$

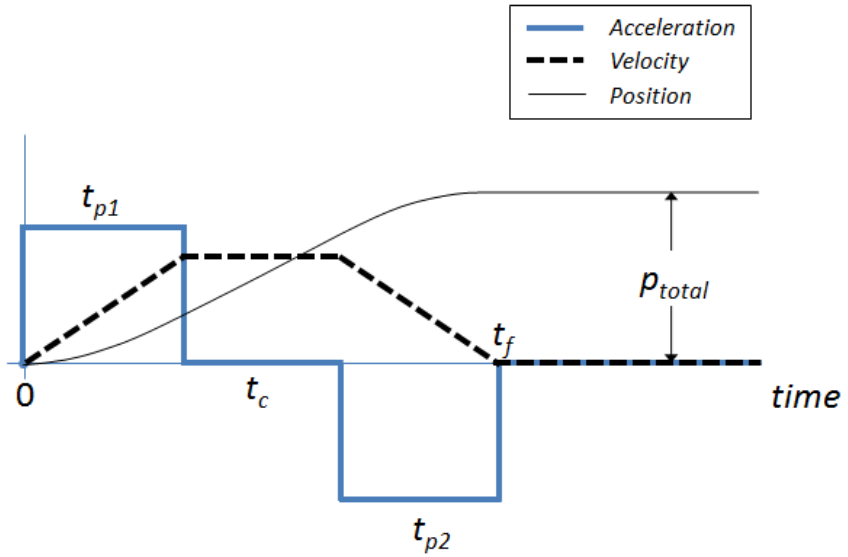




Nonlinear Optimization - Modeling Gantry Crane

- Determine acceleration profile that minimizes payload swing





Constraints :

- $t_f \geq t_{p1} + t_{p2}$
- $1\text{s} \leq t_{p1} \leq 20\text{s}$
- $1\text{s} \leq t_{p2} \leq 20\text{s}$
- $4\text{s} \leq t_f \leq 25\text{s}$

Symbolic Math Toolbox

- Perform exact computations using familiar MATLAB syntax in MATLAB
  - Integration
  - Differentiation
  - Equation solving
  - Transformations
  - Simplification
  - Unit conversion
  - Variable precision arithmetic
- Results in typeset math in Live Editor
- Integrates with MATLAB, Simulink, Simscape

$$\begin{pmatrix} \frac{6\left(3x_1^2-1\right)^2-36x_1\left(-x_1^3+x_1+x_2\right)+2}{\sigma_2}-\frac{\sigma_3^2}{\sigma_2^2} & \sigma_1 \\ \sigma_1 & \frac{6}{\sigma_2}-\frac{\left(-6x_1^3+6x_1+6x_2\right)^2}{\sigma_2^2} \end{pmatrix}$$

where

$$\sigma_1=\frac{\left(-6x_1^3+6x_1+6x_2\right)\sigma_3}{\sigma_2^2}-\frac{18x_1^2-6}{\sigma_2}$$

$$\sigma_2=\left(x_1-\frac{4}{3}\right)^2+3\left(-x_1^3+x_1+x_2\right)^2+1$$

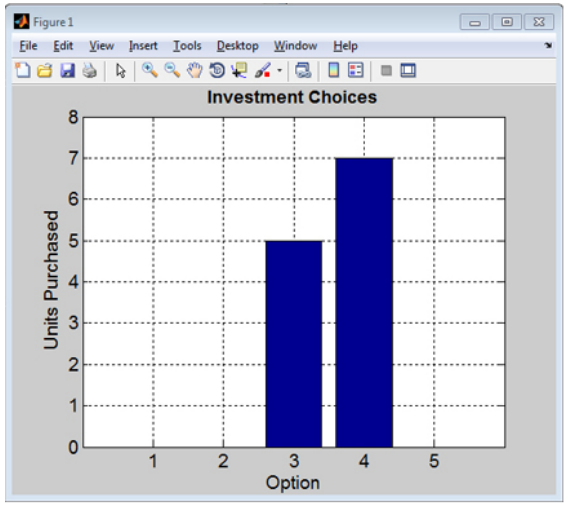
$$\sigma_3=6\left(3x_1^2-1\right)\left(-x_1^3+x_1+x_2\right)-2x_1+\frac{8}{3}$$

Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization

Mixed-Integer Programming

- Many things exist in discrete amounts:
  - Shares of stock
  - Number of cars a factory produces
  - Number of cows on a farm
- Often have binary decisions:
  - On/off
  - Buy/don't buy
- Mixed-integer linear programming:
  - Solve optimization problem while enforcing that certain variables need to be integer



Mixed-Integer Linear Programming

Continuous and integer variables

$$x_1 \in [0, 100] \qquad x_2 \in \{1,2,3,4,5\}$$

Linear objective and constraints

$$\min_x \; -x_1 - 2x_2$$

$$\text{such that} \quad \begin{cases} x_1 + 4x_2 \leq 20 \\ x_1 + x_2 = 10 \end{cases}$$

Optimize Gift Card Spending

Problem:

- Given gift cards to different stores and a shopping list of desired purchases, decide how to spend the gift cards to use as much of the gift card money as possible.

Constraints:

- You cannot overspend the gift card.
- You can purchase one of any item, and must purchase one of a specific item.

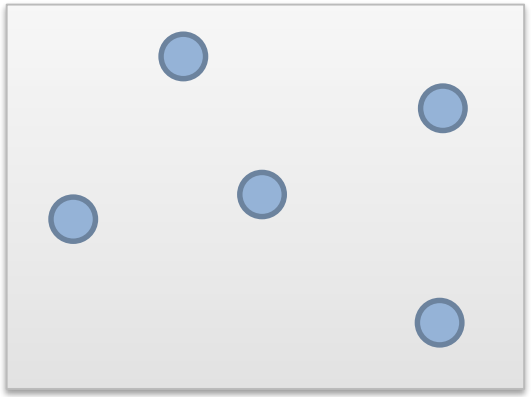
Traveling Salesman Problem

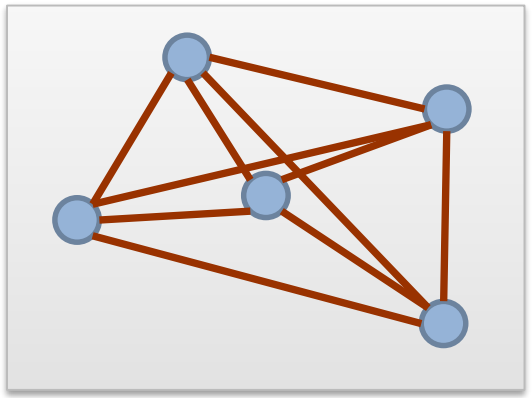
Problem

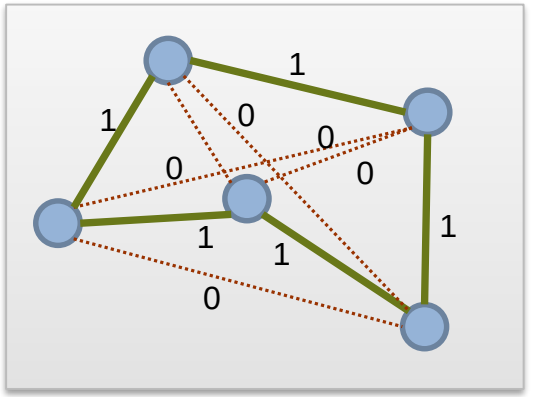
- How to find the shortest path through a series of points?

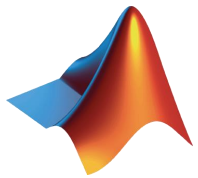
Solution

- Calculate distances between all combinations of points
- Solve an optimization problem where variables correspond to trips between two points







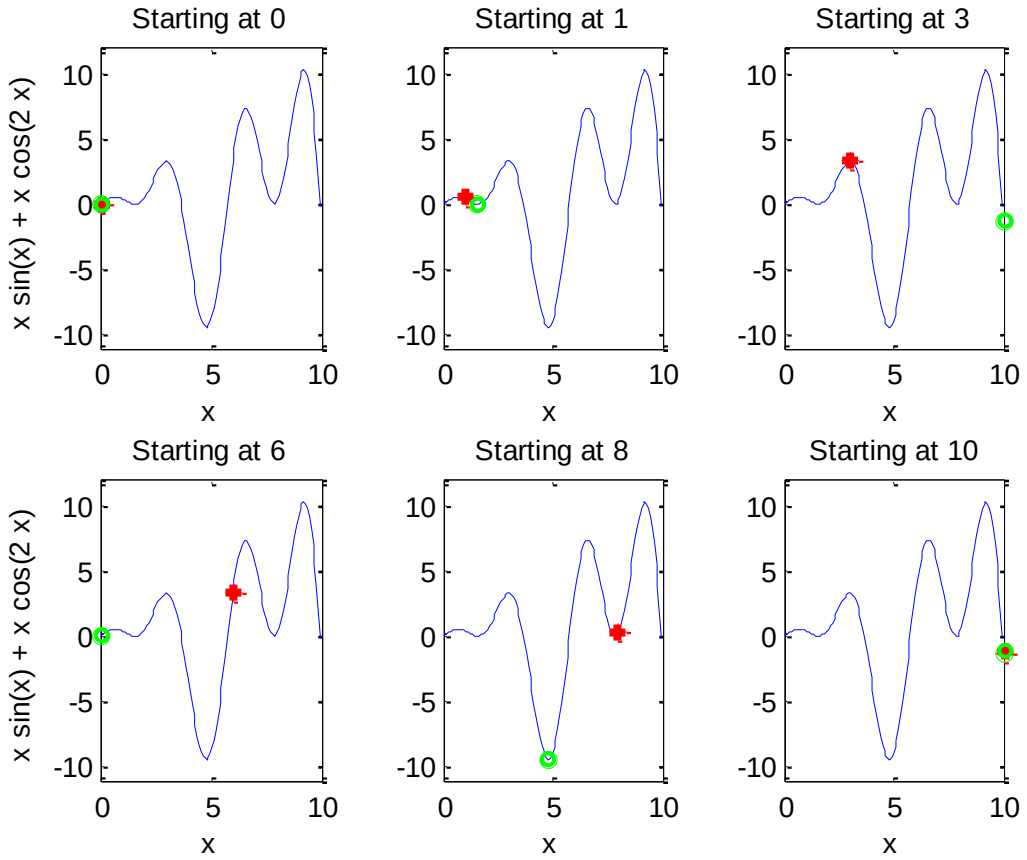


Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization

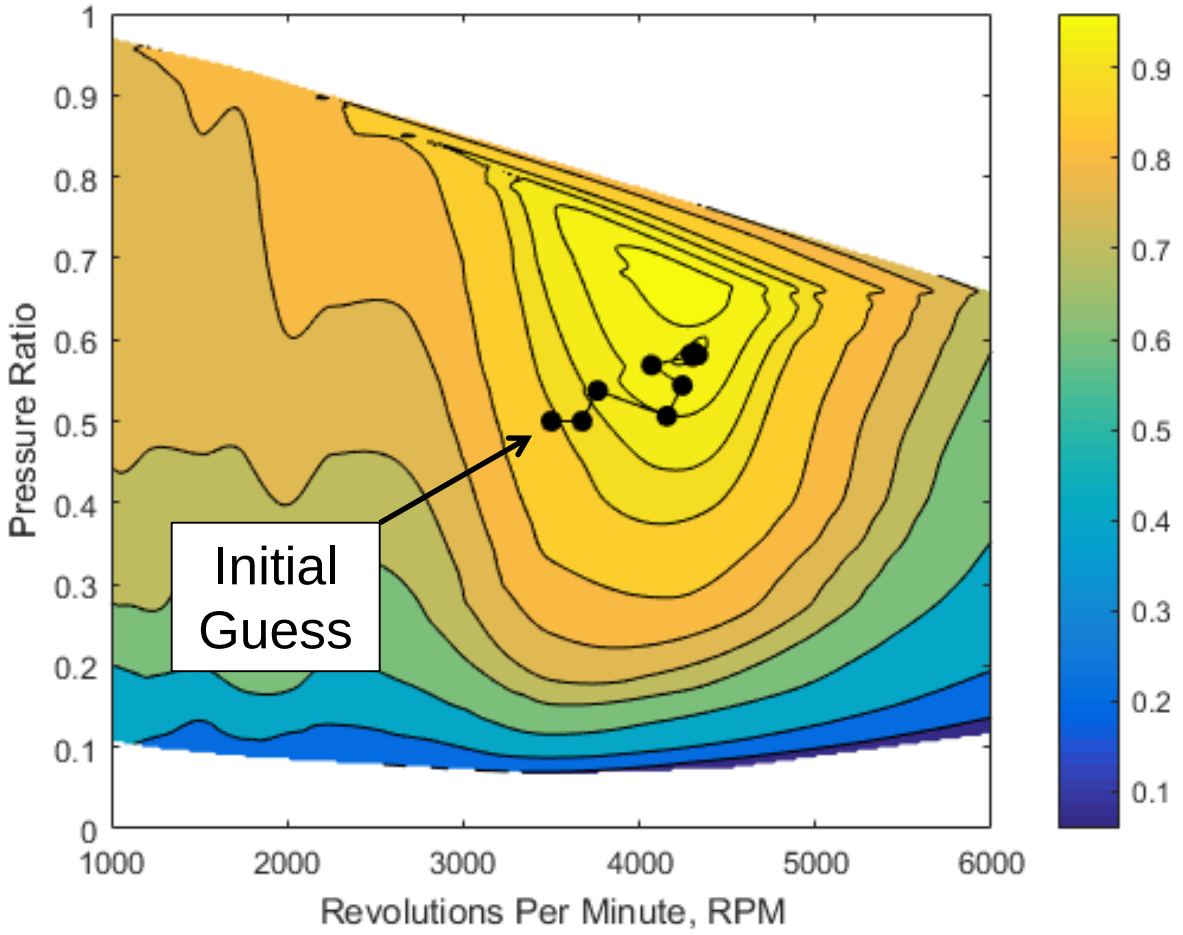
Example Global Optimization Problems

Why does `fmincon` have a hard time finding the function minimum?



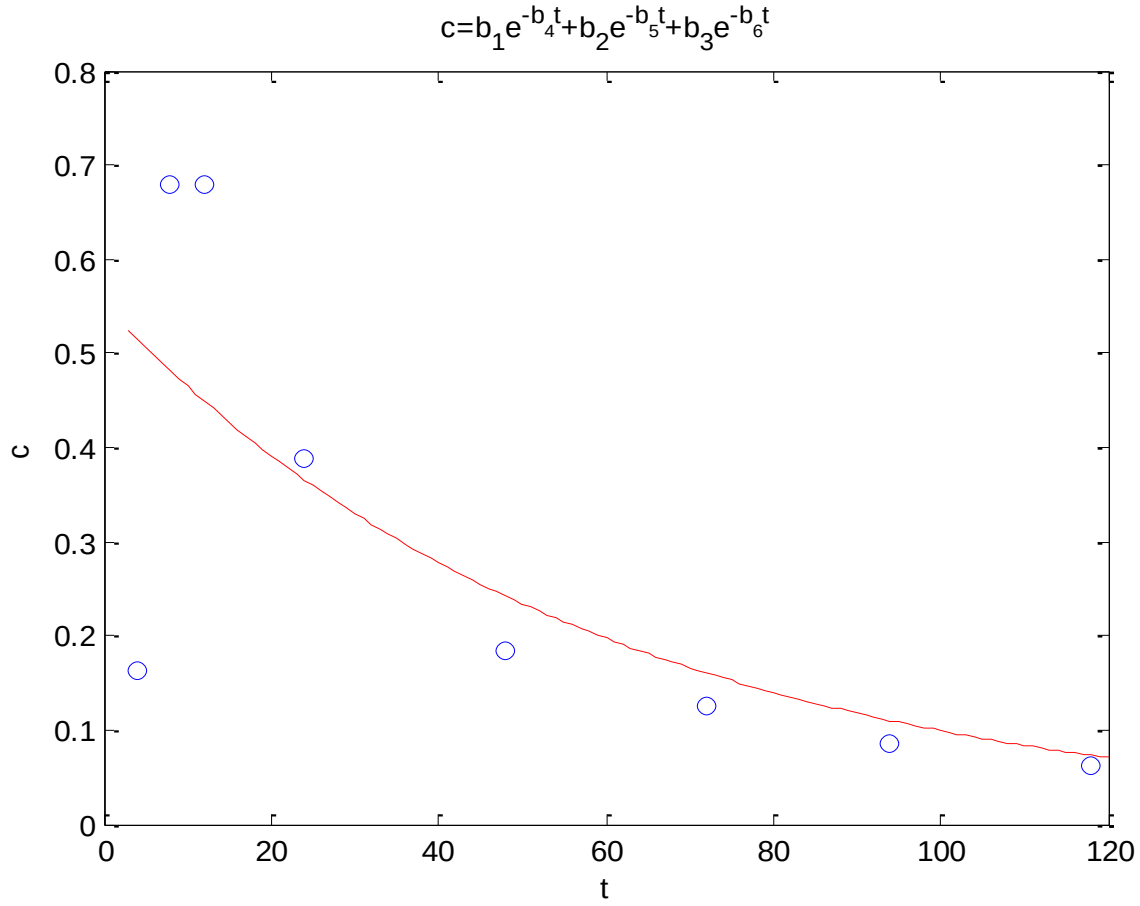
Example Global Optimization Problems

Why didn't `fminunc` find the maximum efficiency?



Example Global Optimization Problems

Why didn’t nonlinear regression find a good fit?



Global Optimization

Goal:

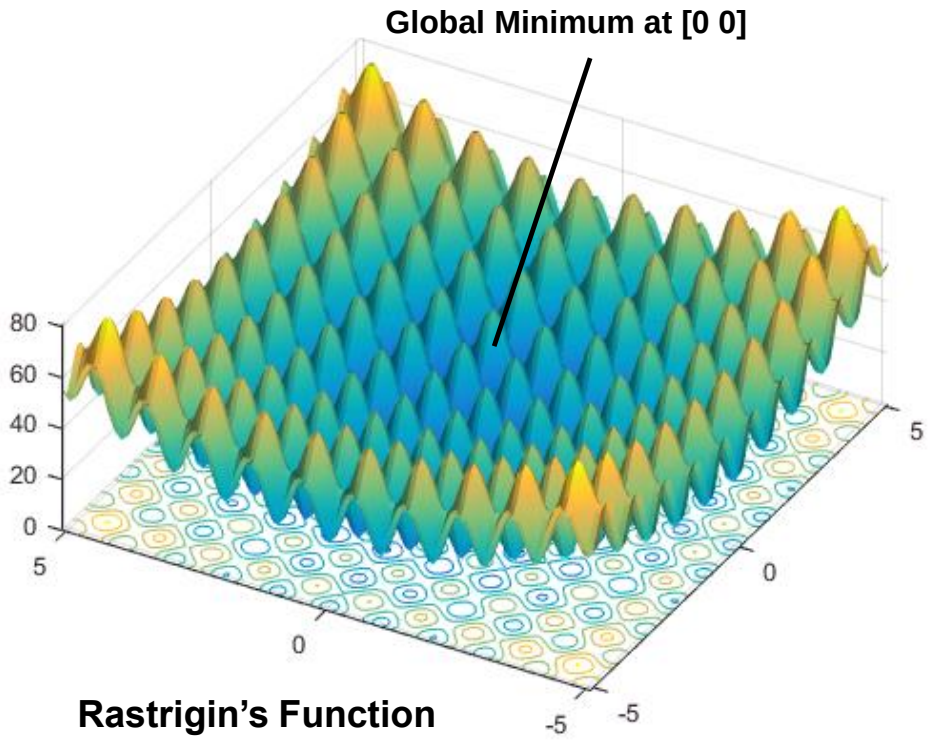
Want to find the **lowest/largest** value of the nonlinear function that has **many local minima/maxima**

Problem:

Traditional solvers often return one of the local minima (not the global)

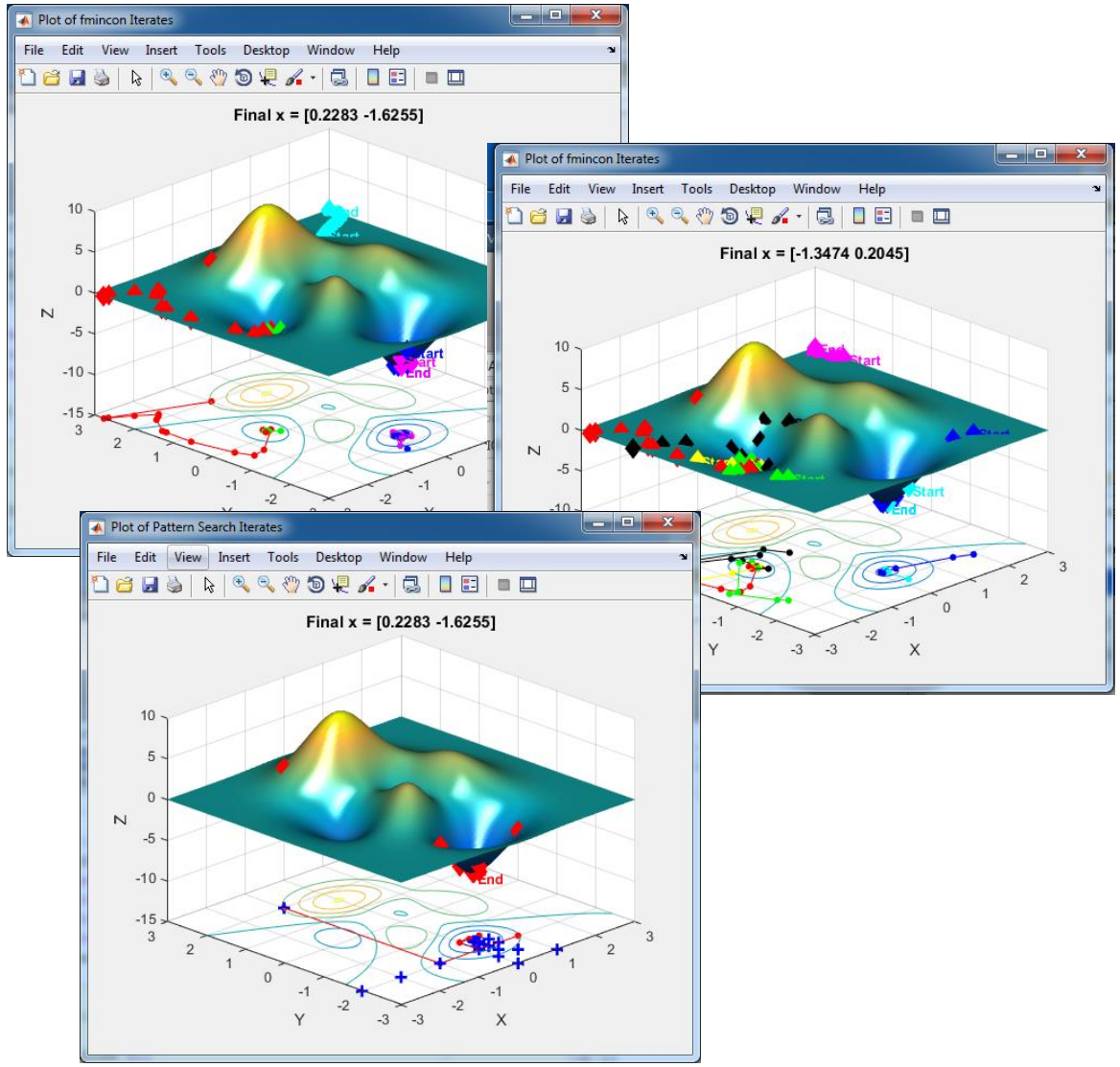
Solution:

A solver that locates globally optimal solutions



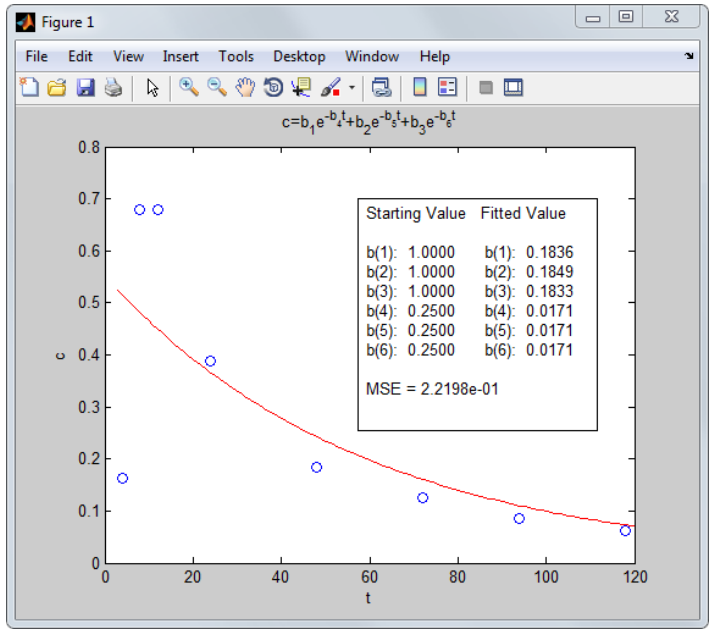
Global Optimization Solvers Covered Today

- Multi Start and Global Search
- Pattern Search
- Genetic Algorithm
- Surrogate Optimization
- Particle Swarm
- Simulated Annealing

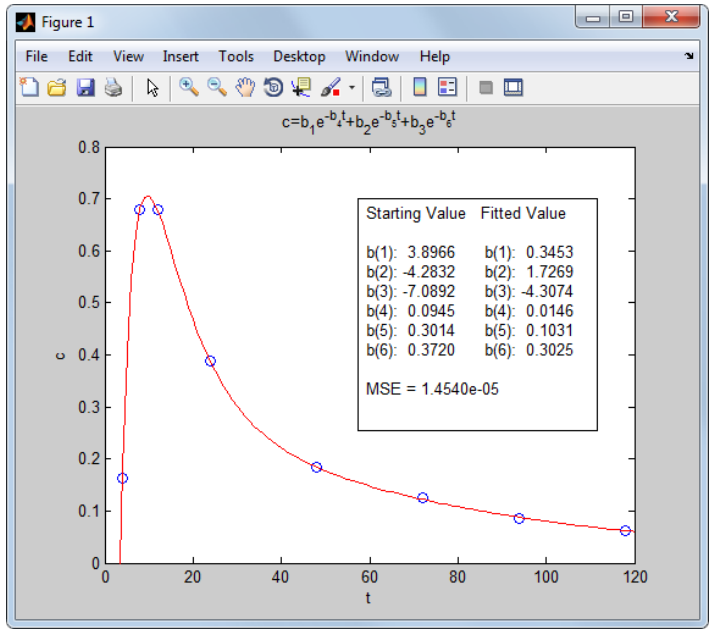


MultiStart Demo – Nonlinear Regression

`lsqcurvefit` solution



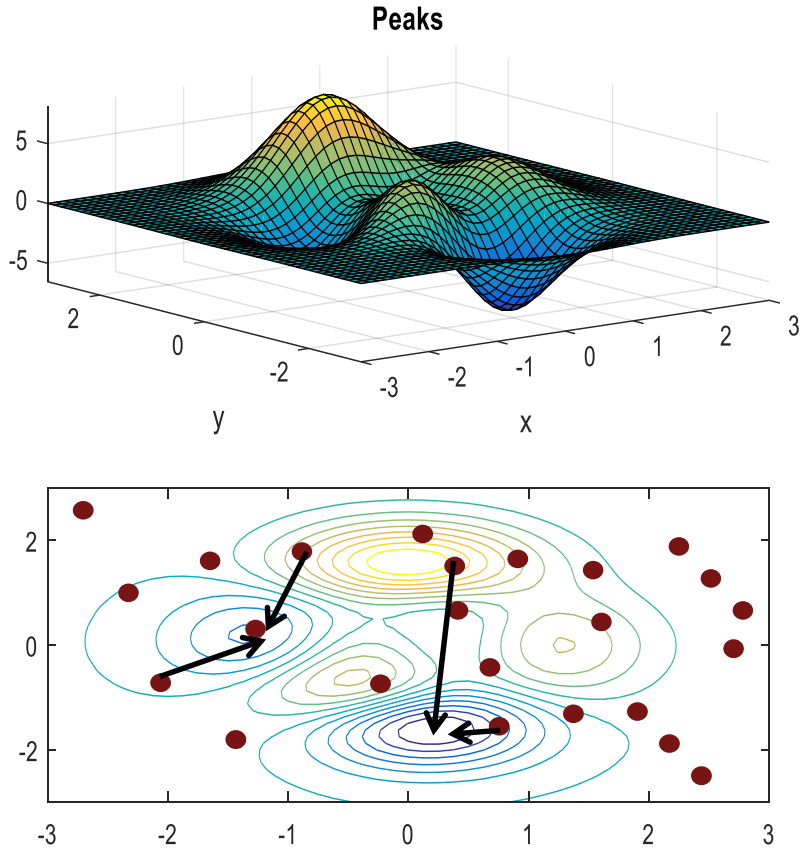
`MultiStart` solution



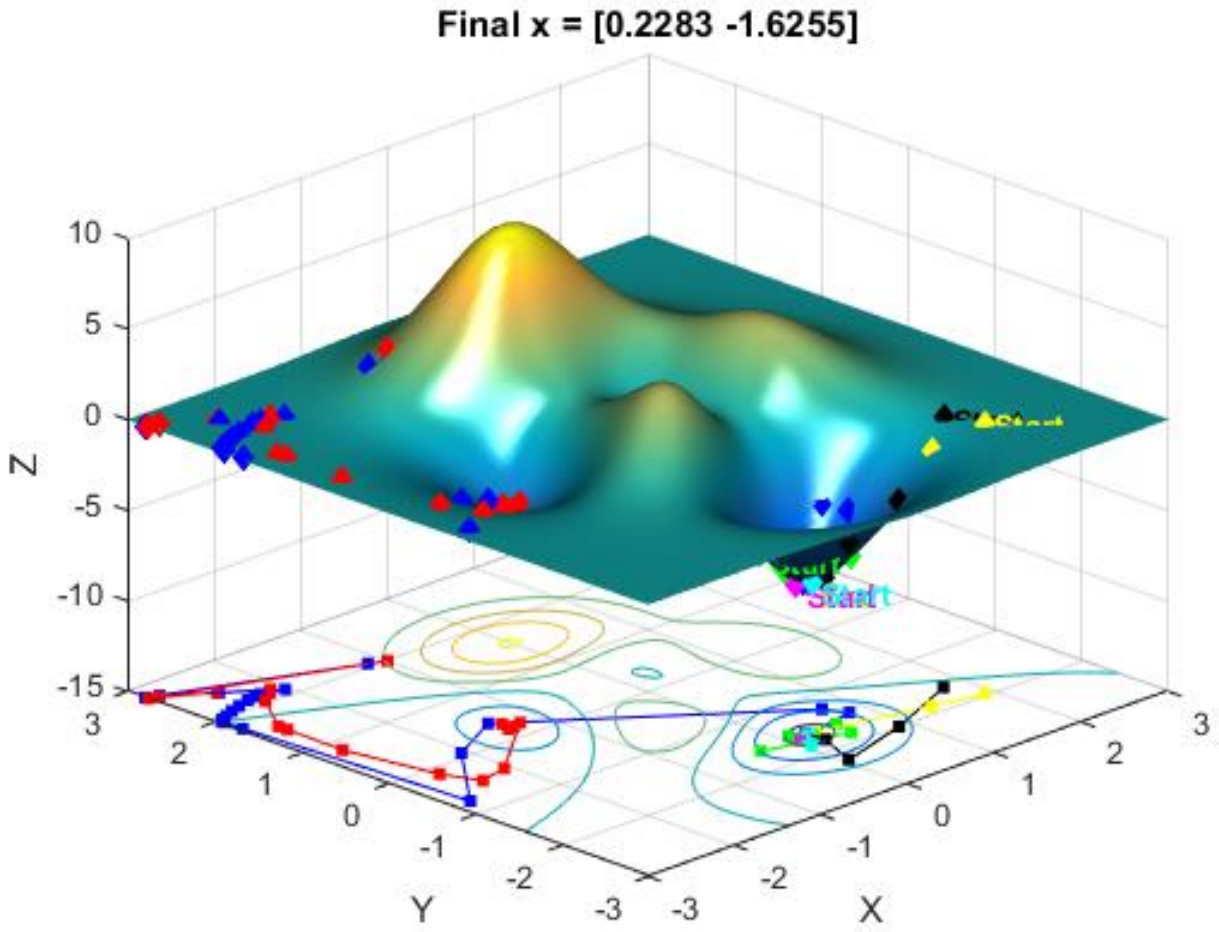
MULTISTART

What is MultiStart?

- Run a local solver from each set of start points
- Option to filter starting points based on feasibility
- Supports parallel computing



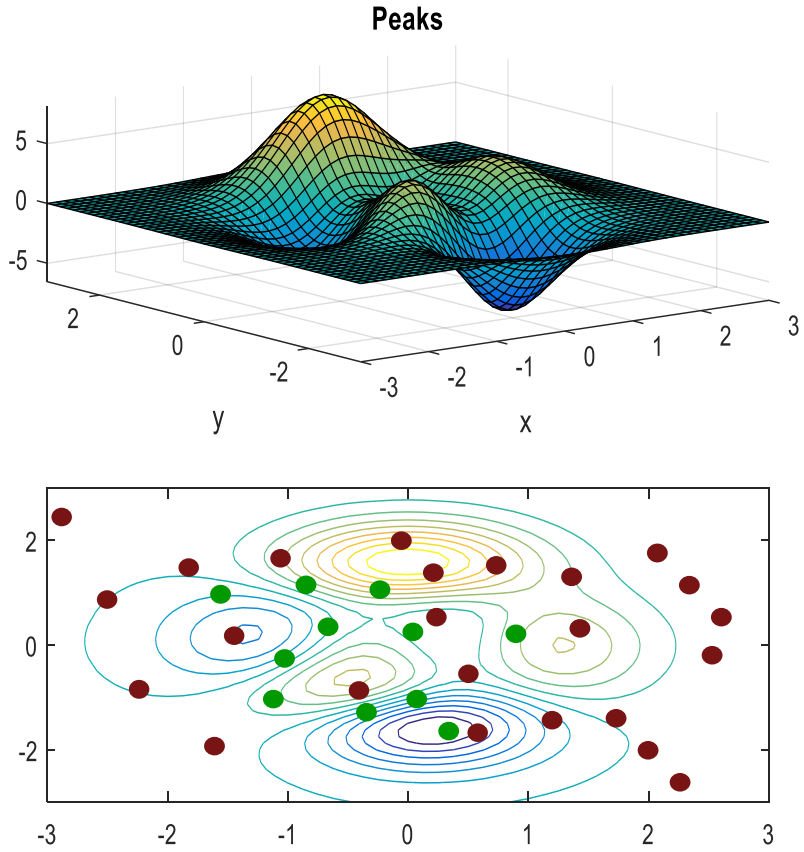
MultiStart Demo – Peaks Function



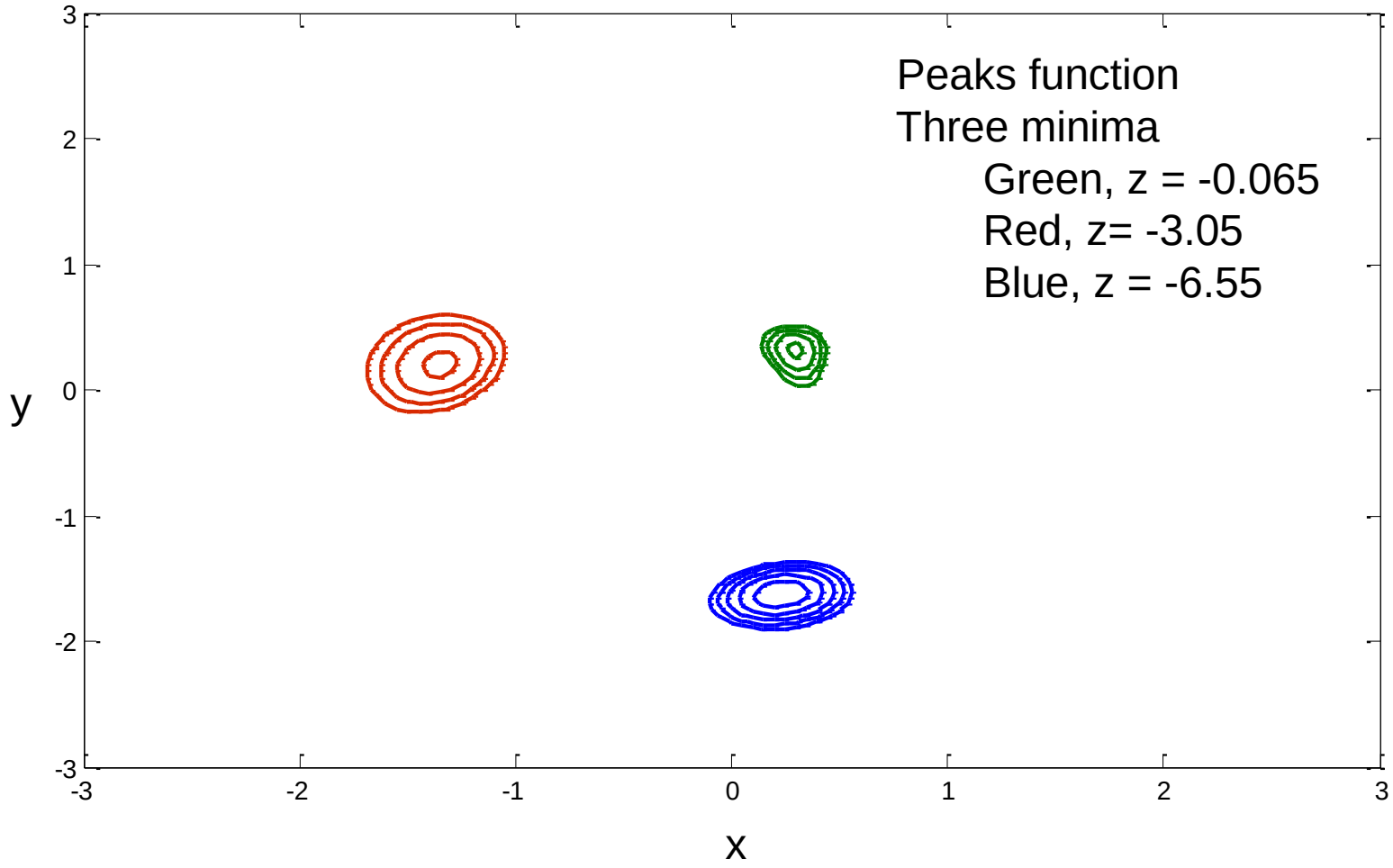
GLOBAL SEARCH

What is GlobalSearch?

- Multistart heuristic algorithm
- Calls `fmincon` from multiple start points to try and find a global minimum
- Filters/removes non-promising start points

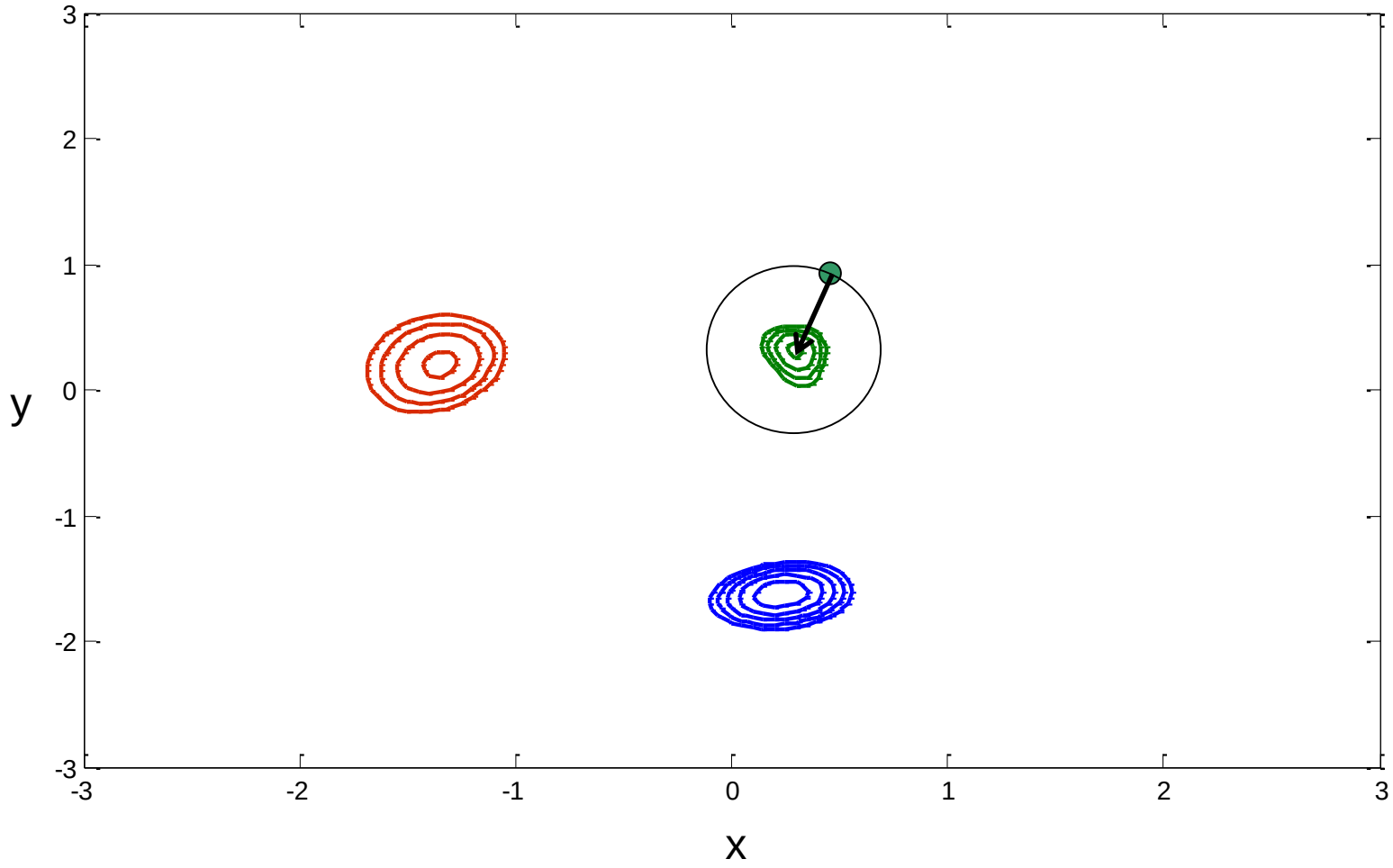


GlobalSearch Overview
Schematic Problem

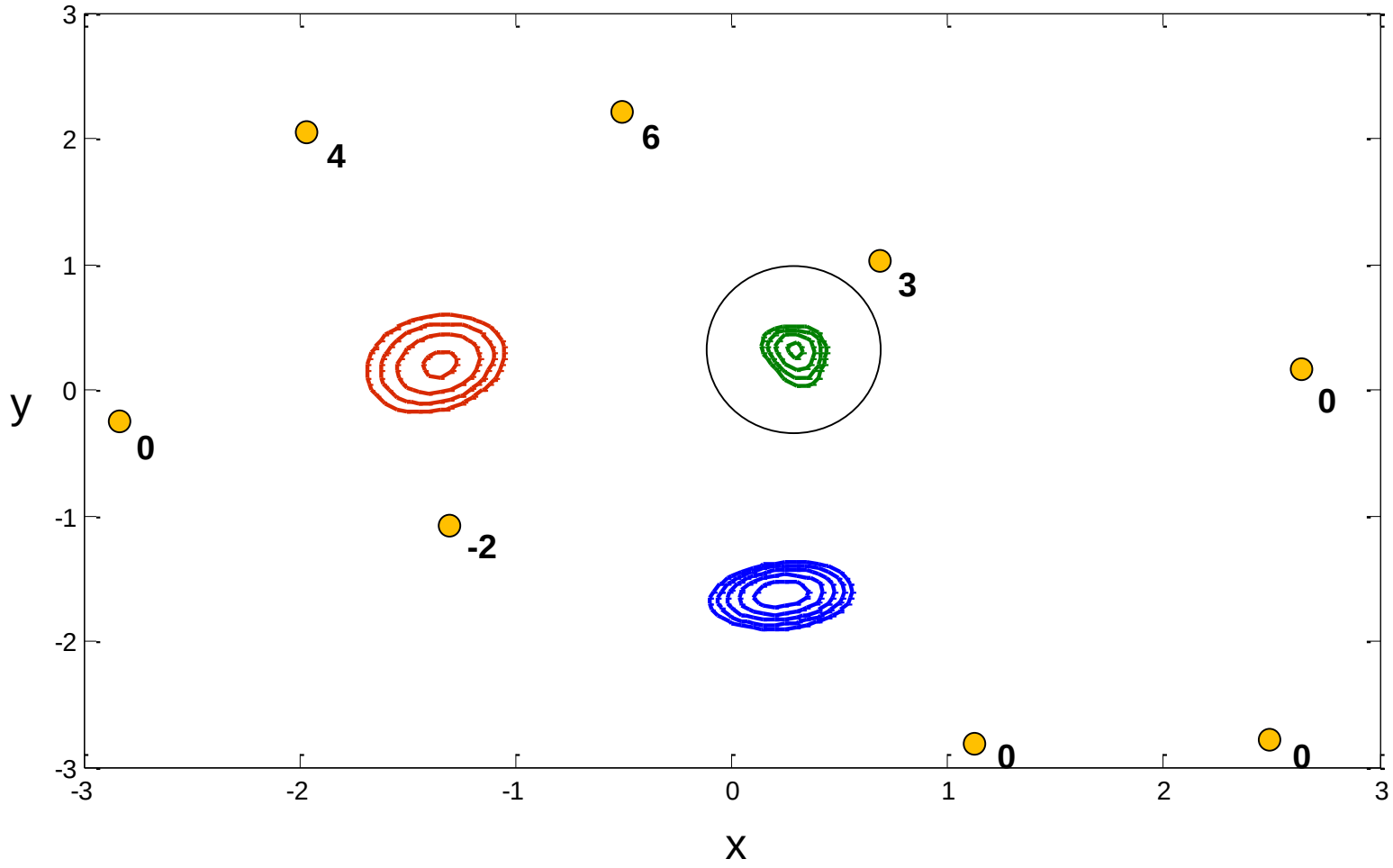


GlobalSearch Overview – Stage 0

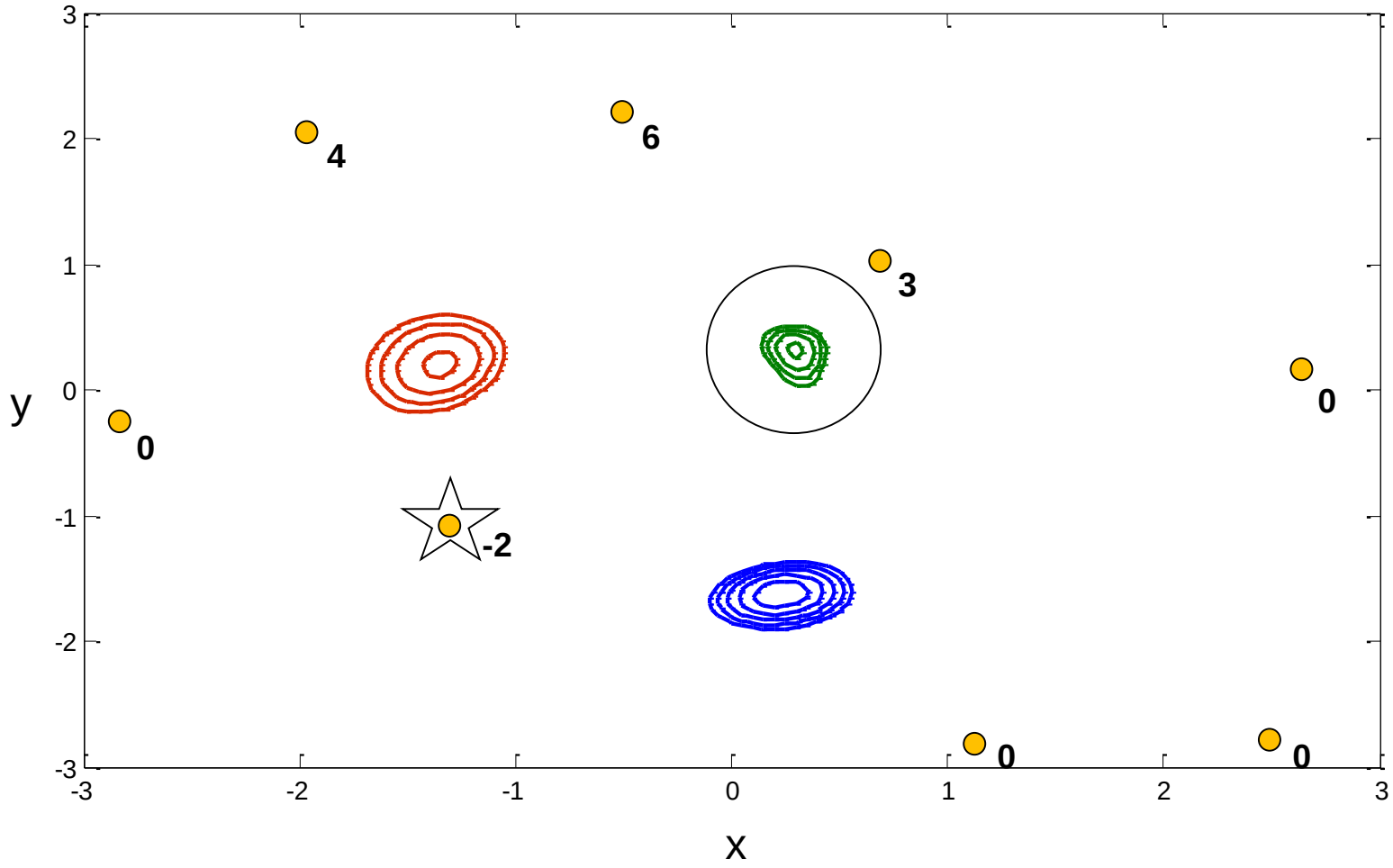
Run from specified x0



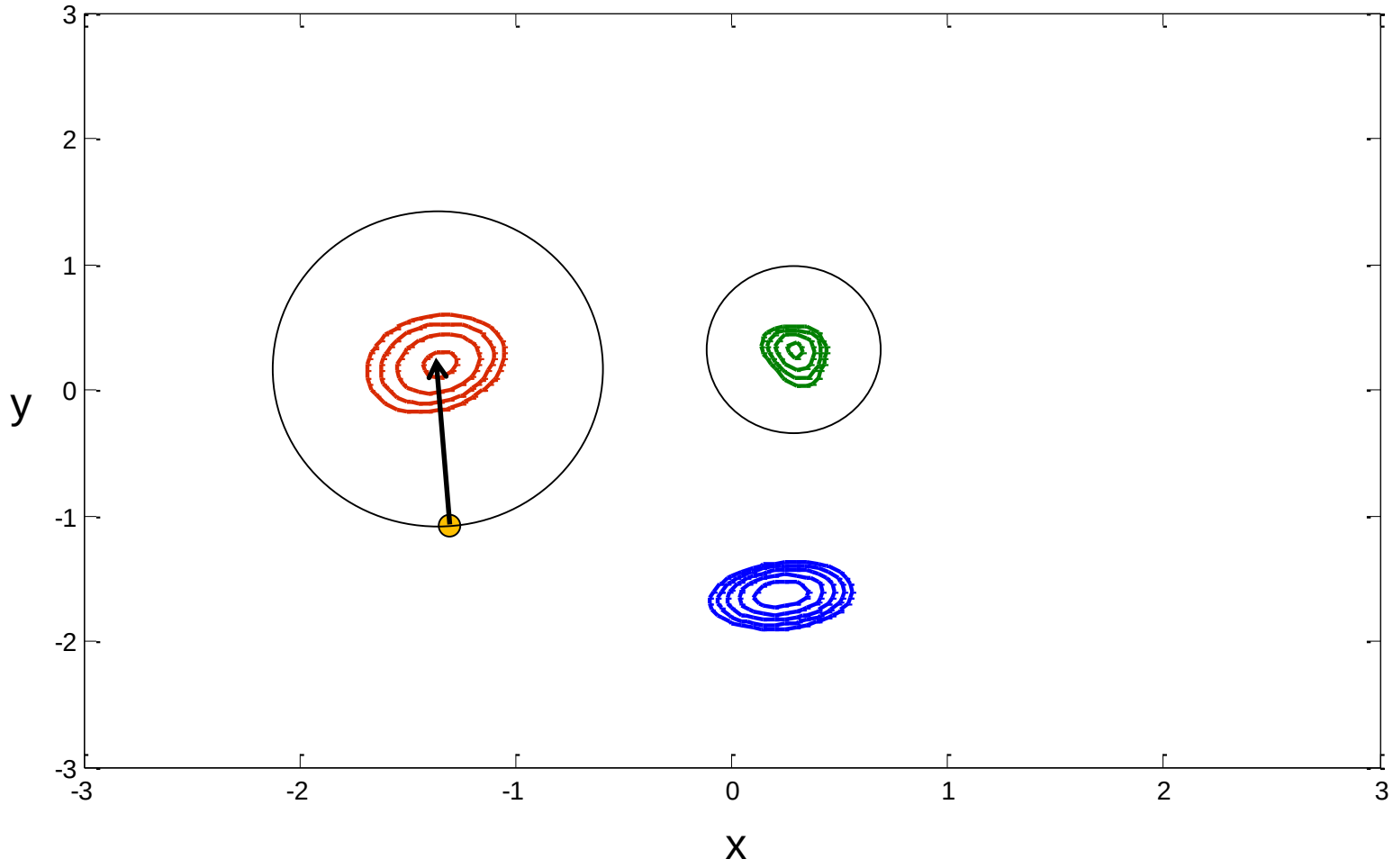
GlobalSearch Overview – Stage 1



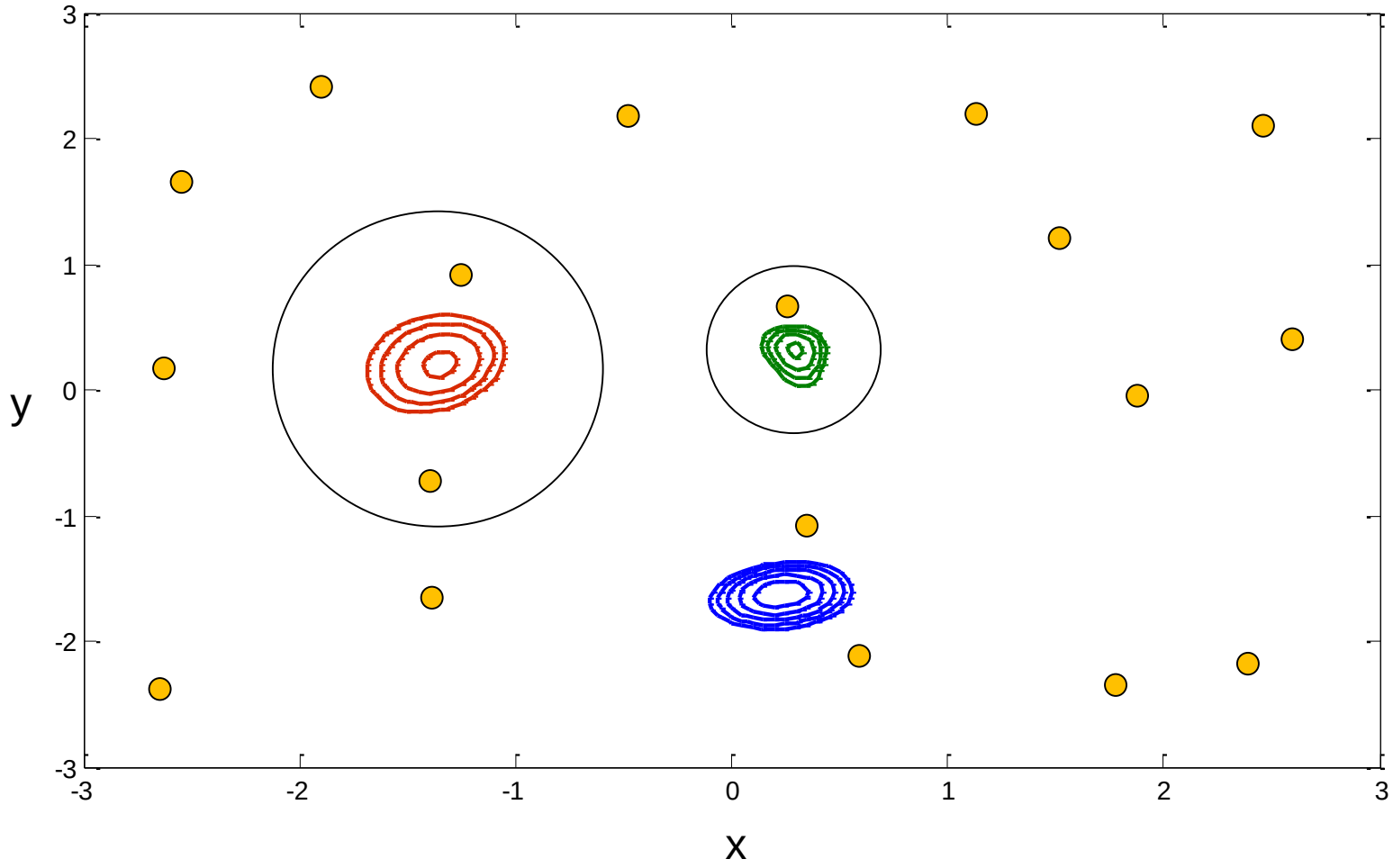
GlobalSearch Overview – Stage 1



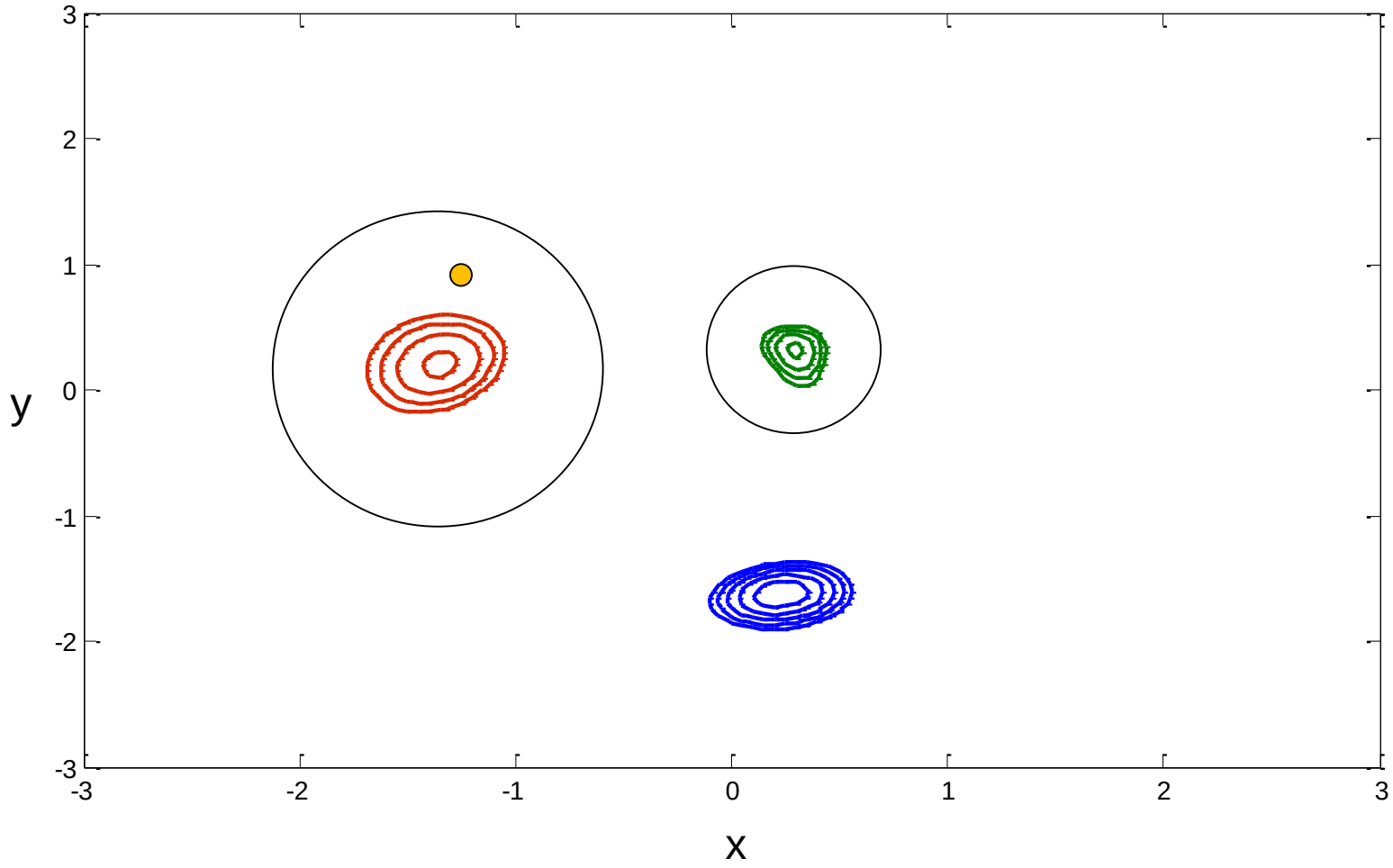
GlobalSearch Overview – Stage 1



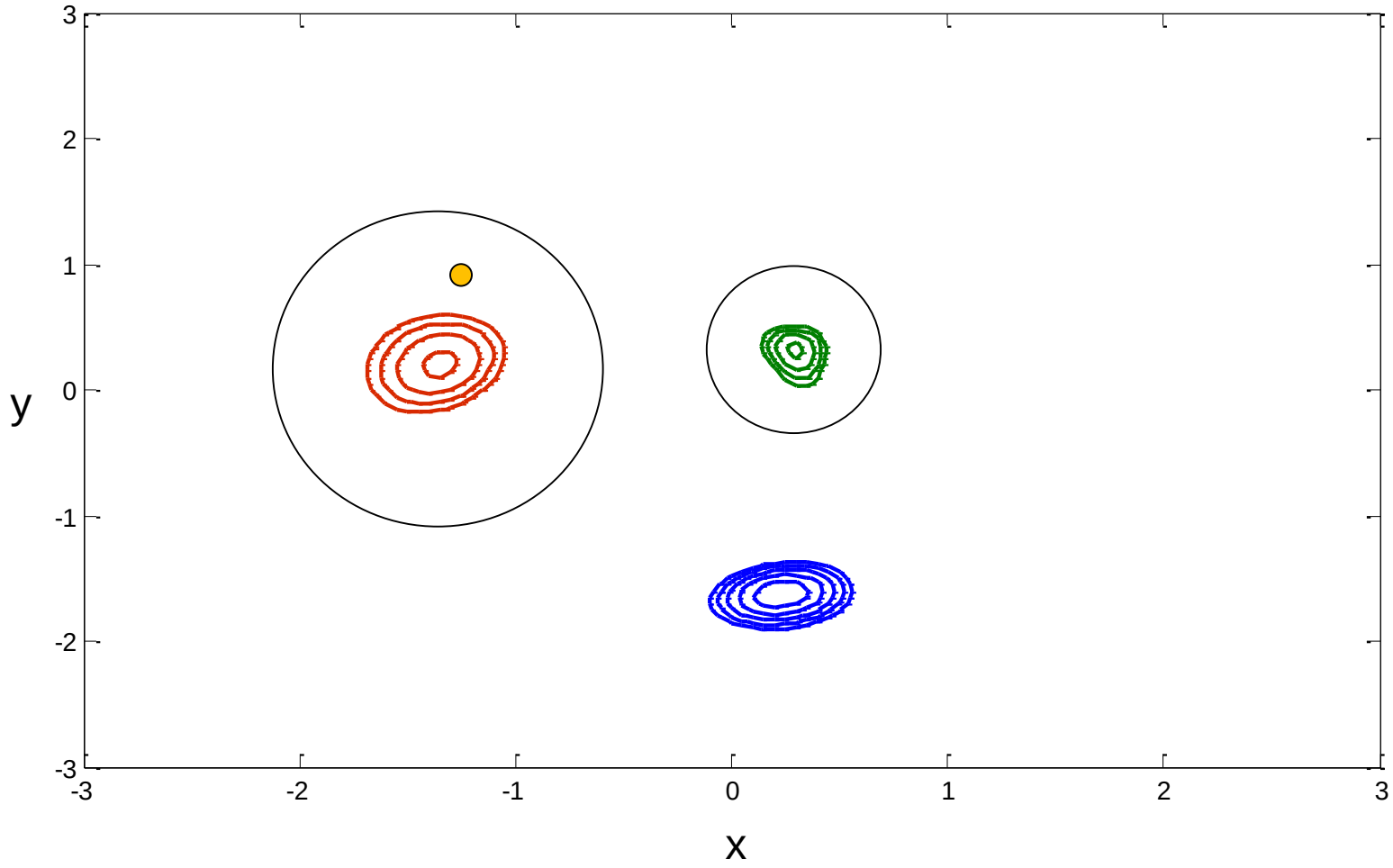
GlobalSearch Overview – Stage 2



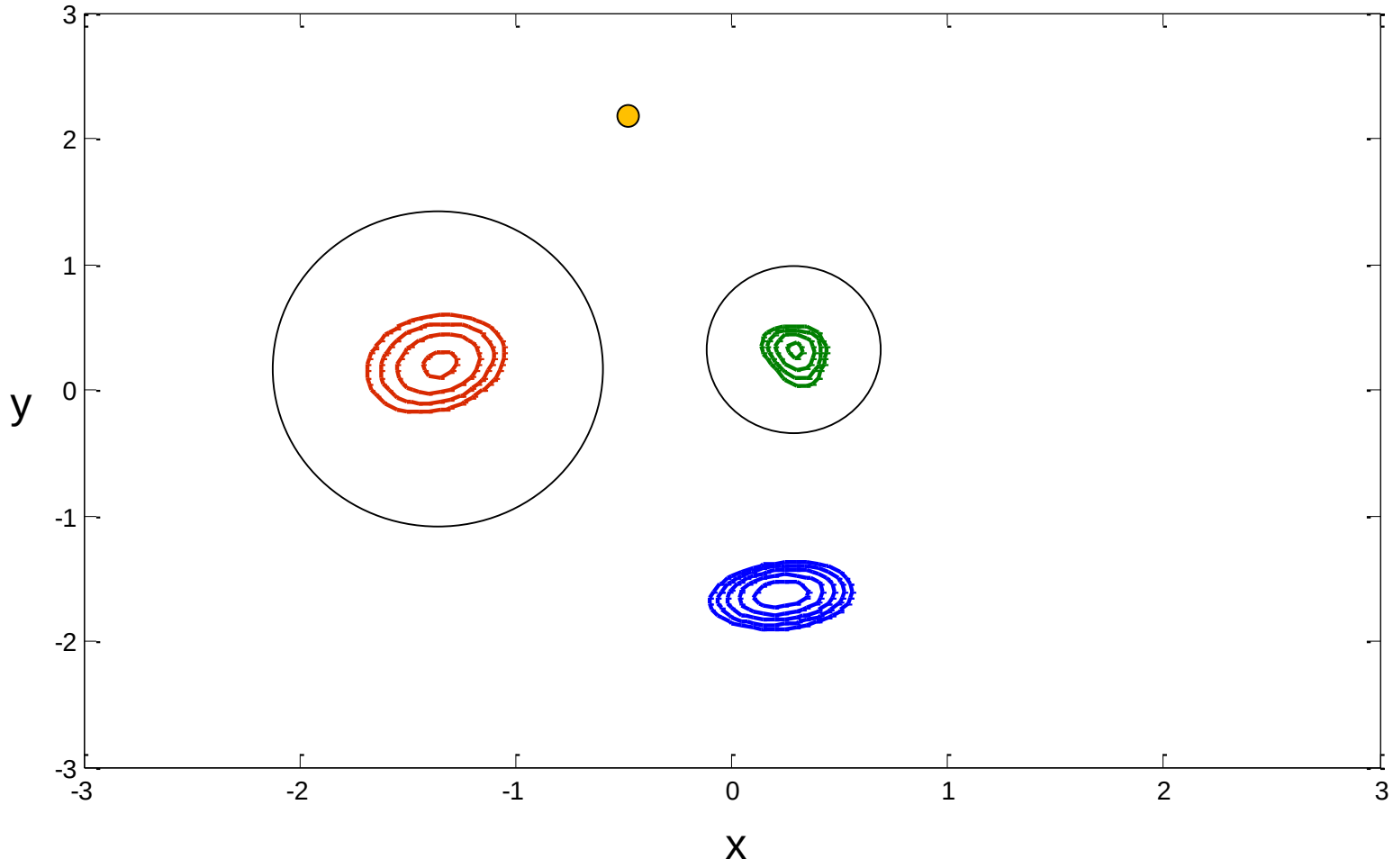
GlobalSearch Overview – Stage 2



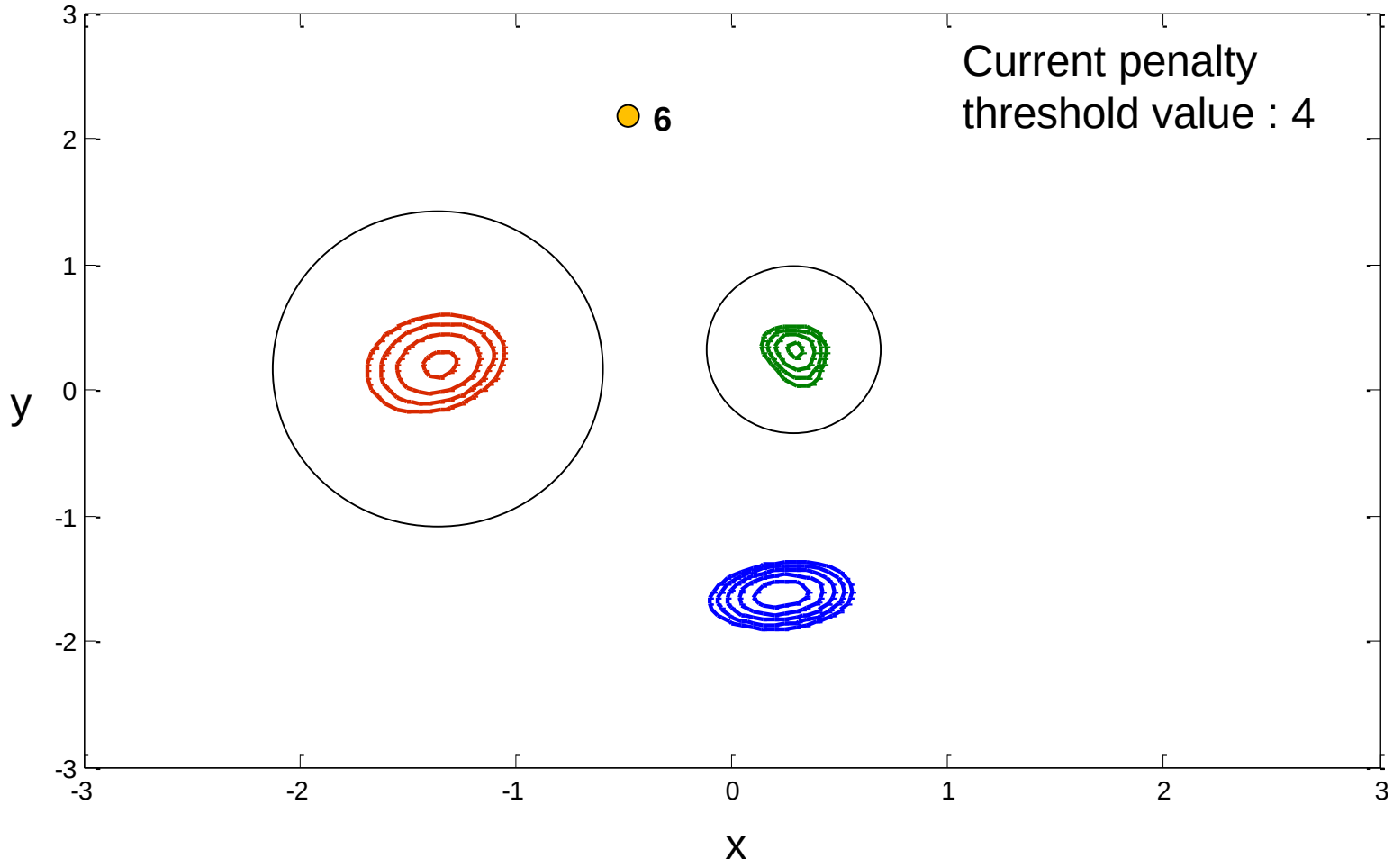
GlobalSearch Overview – Stage 2



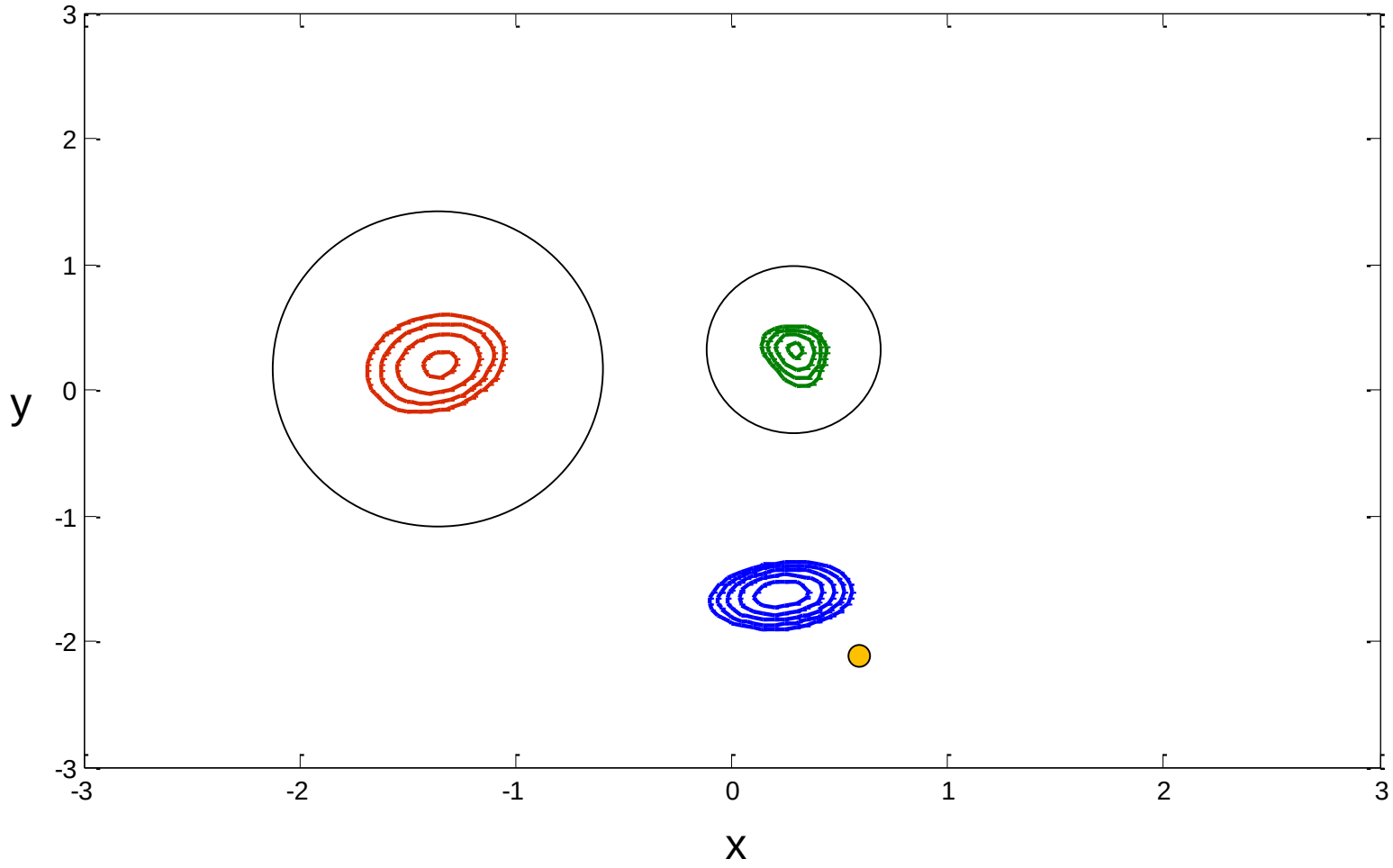
GlobalSearch Overview – Stage 2



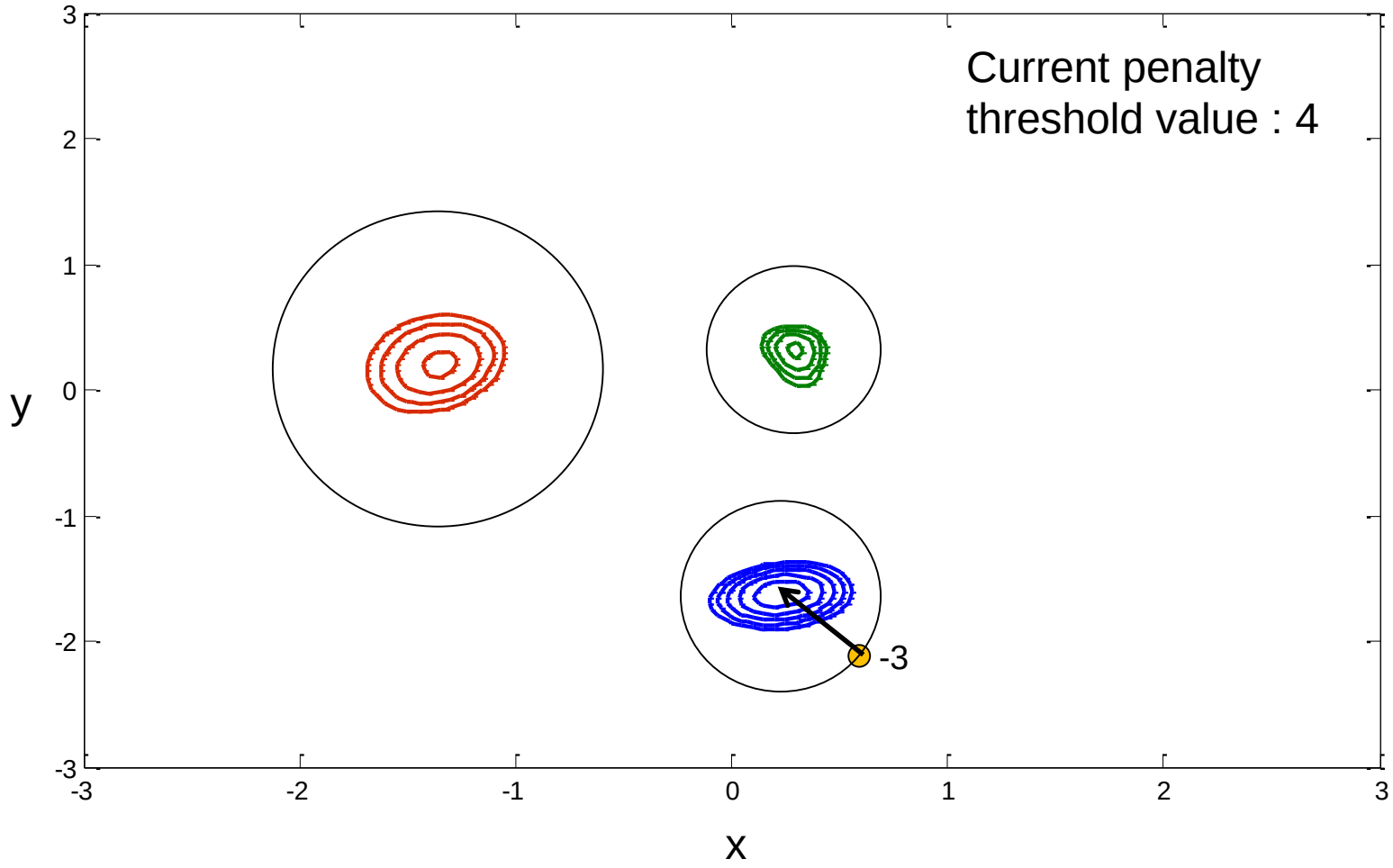
GlobalSearch Overview – Stage 2



GlobalSearch Overview – Stage 2

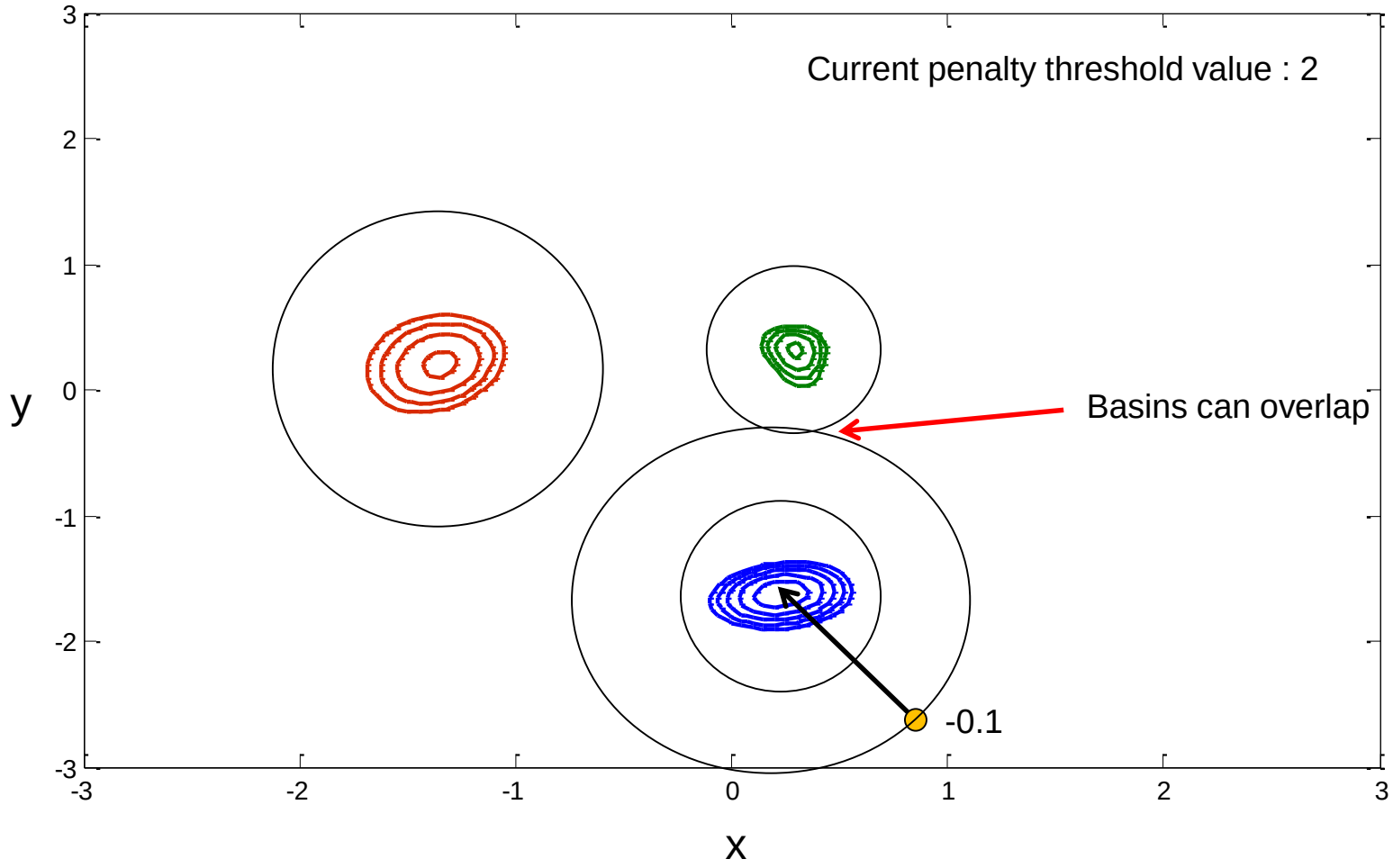


GlobalSearch Overview – Stage 2

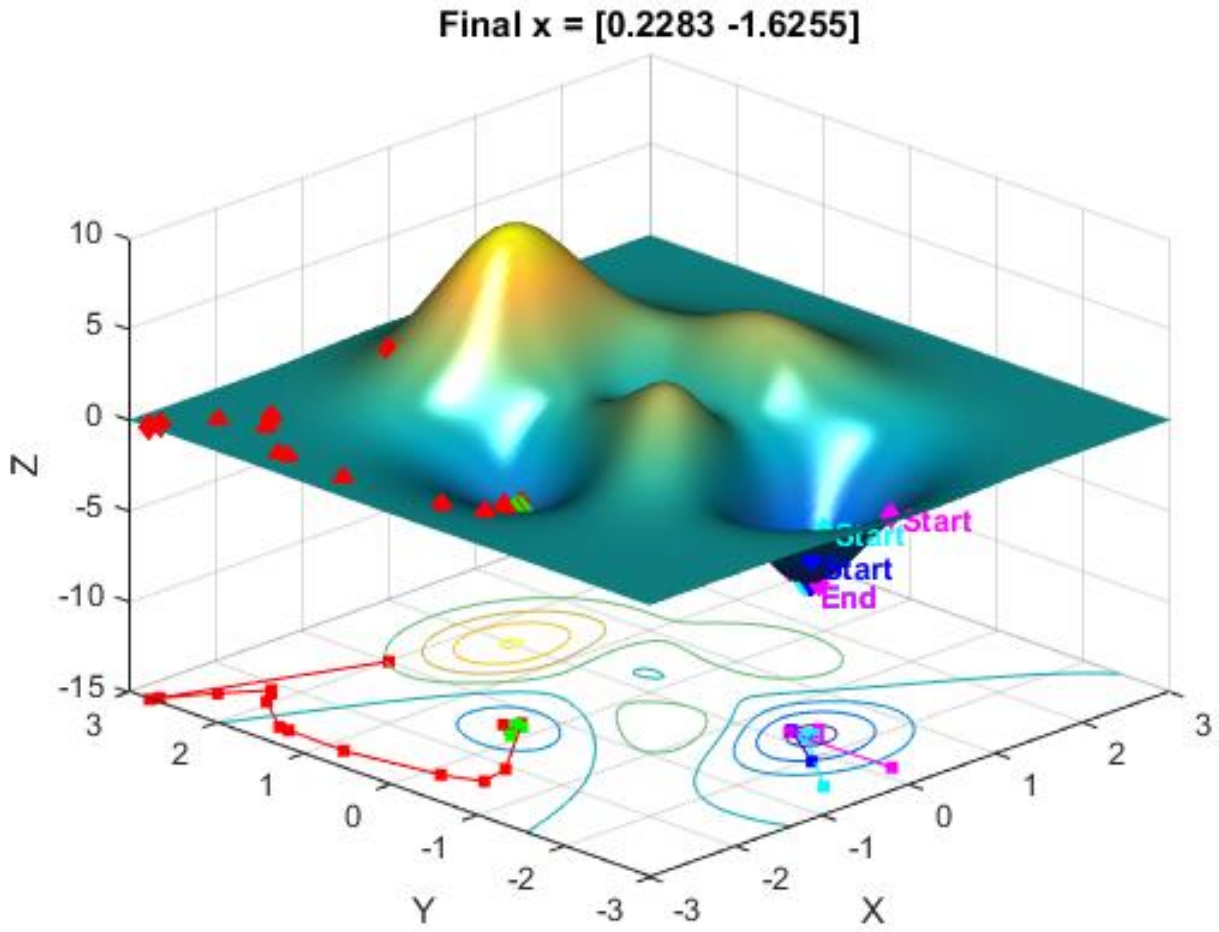


GlobalSearch Overview – Stage 2

Expand basin of attraction if minimum already found



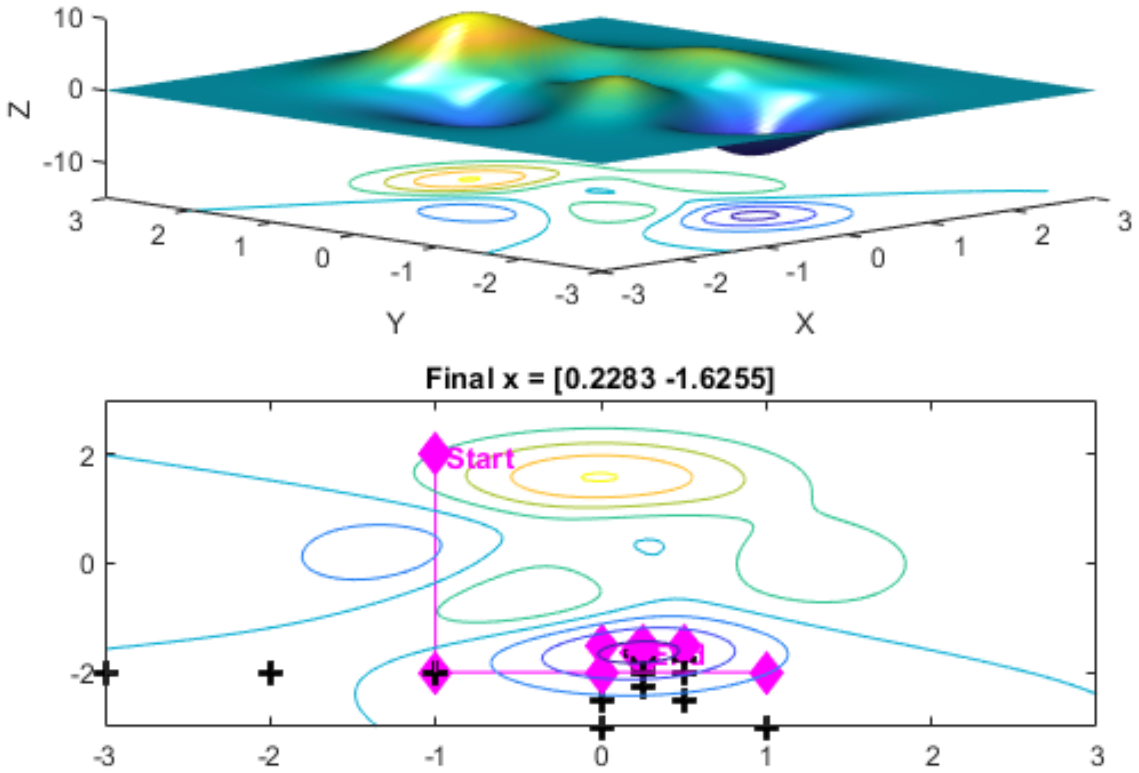
GlobalSearch Demo – Peaks Function



PATTERN SEARCH (DIRECT SEARCH)

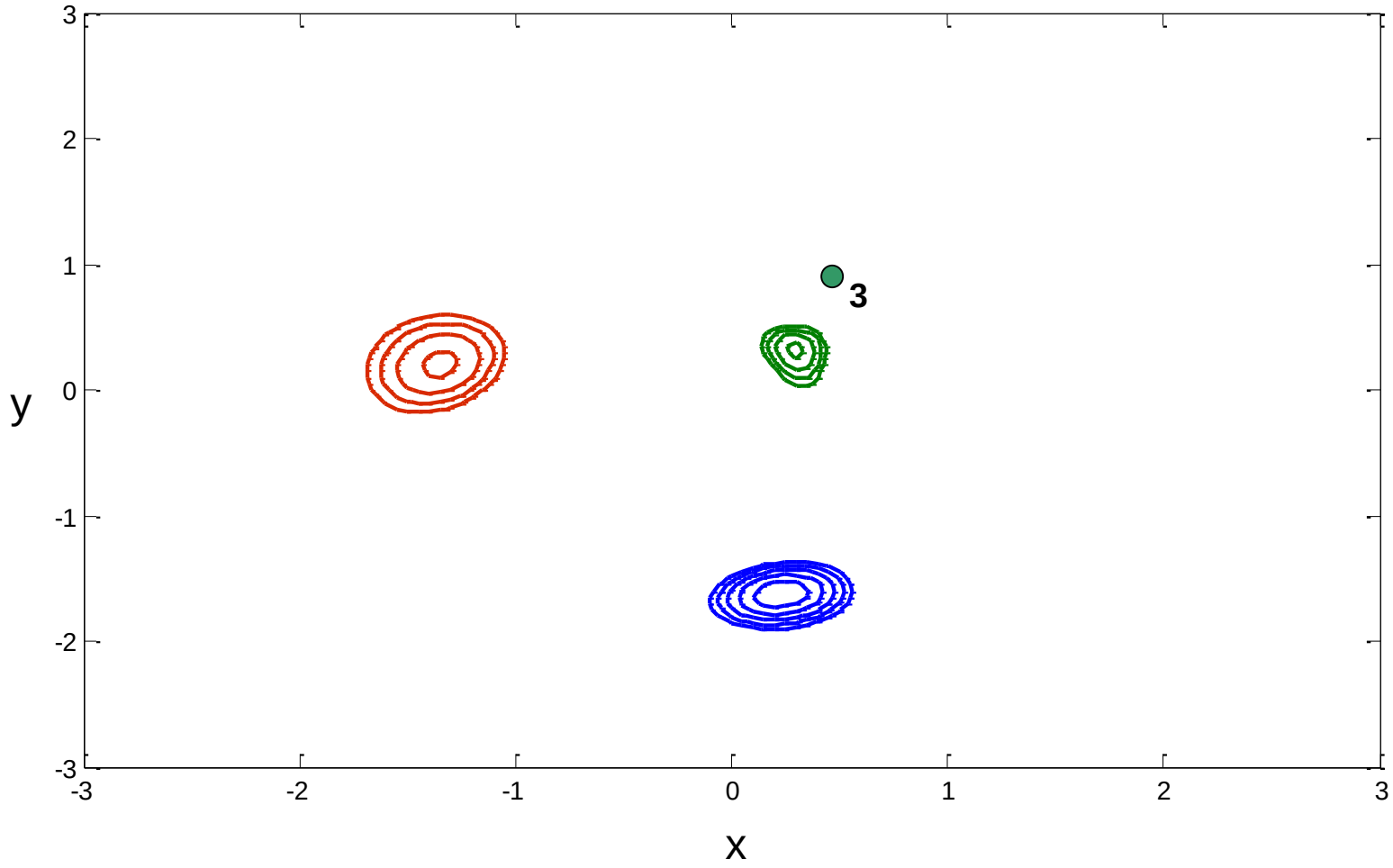
What is Pattern Search?

- An approach that uses a pattern of search directions around the existing points
- Expands/contracts around the current point when a solution is not found
- Does not rely on gradients: works on smooth and nonsmooth problems



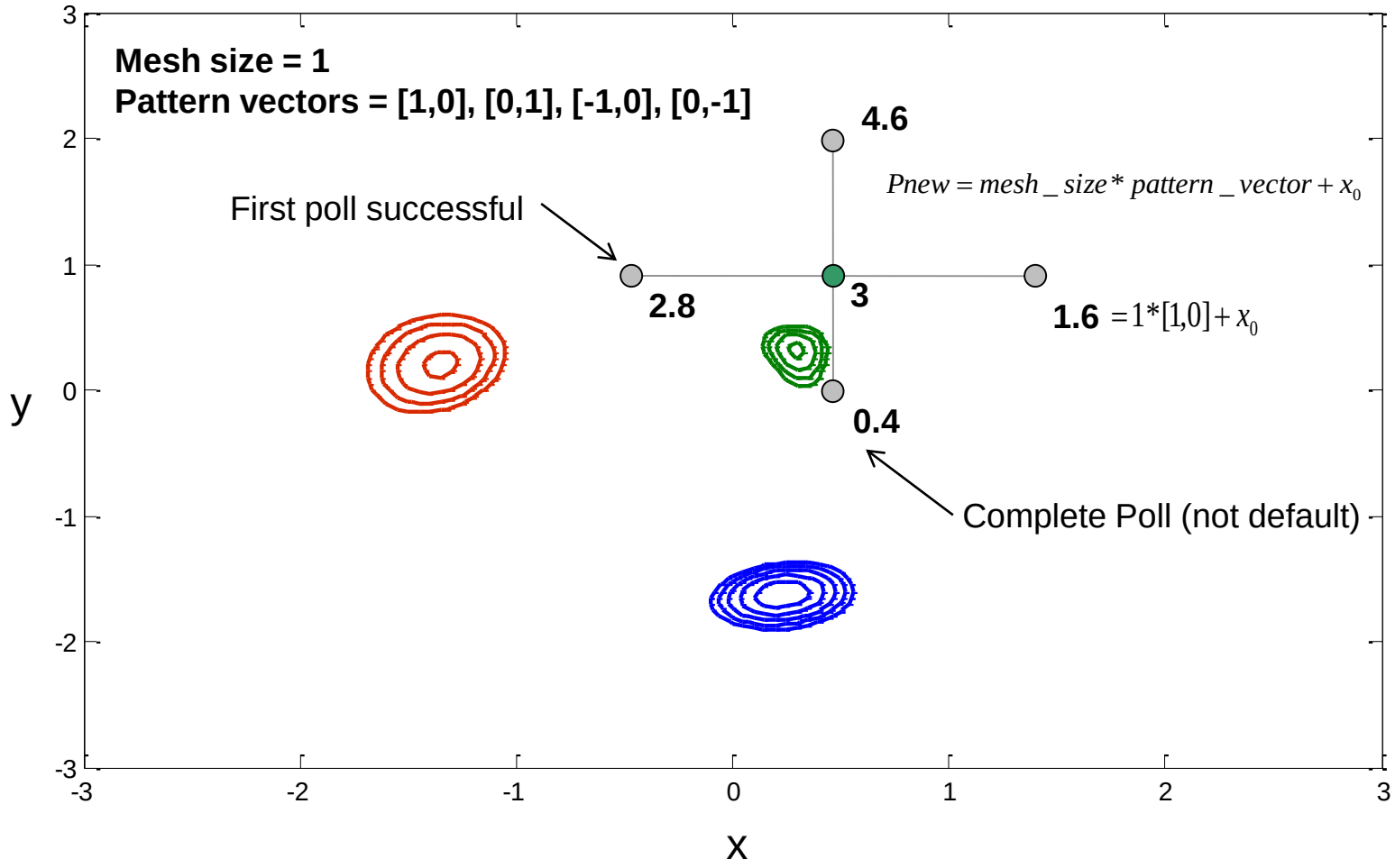
Pattern Search Overview – Iteration 1

Run from specified x0

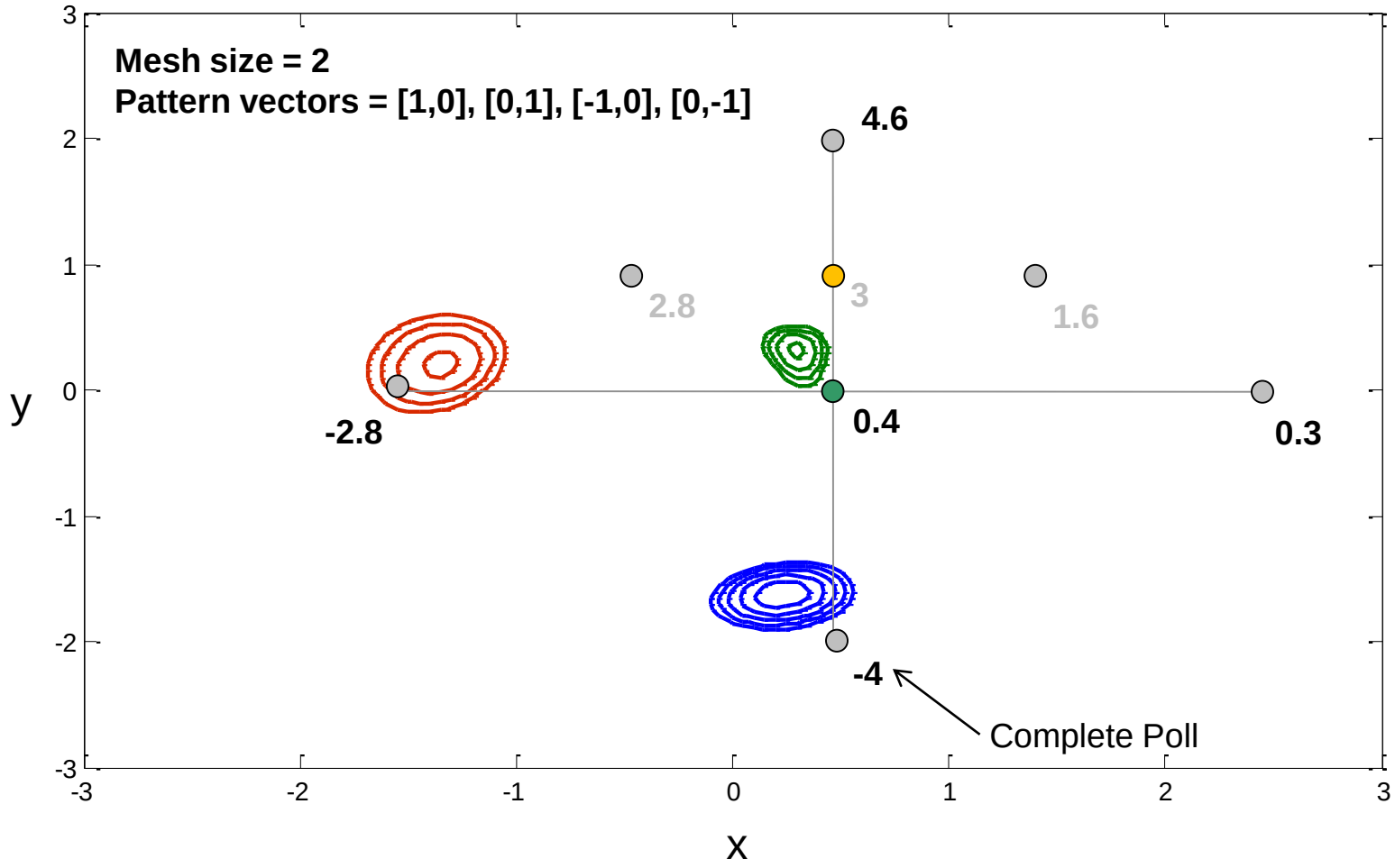


Pattern Search Overview – Iteration 1

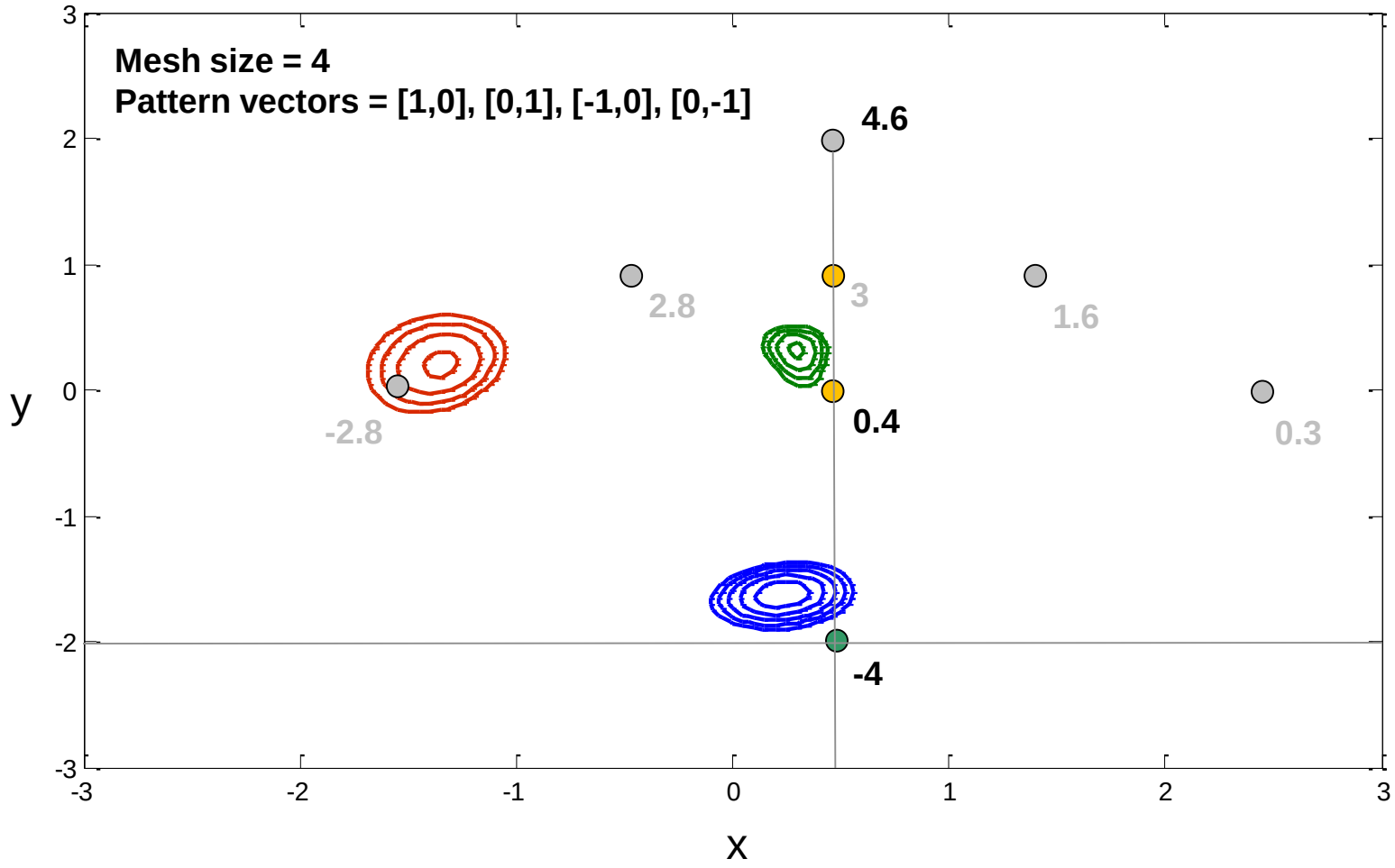
Apply pattern vector, poll new points for improvement



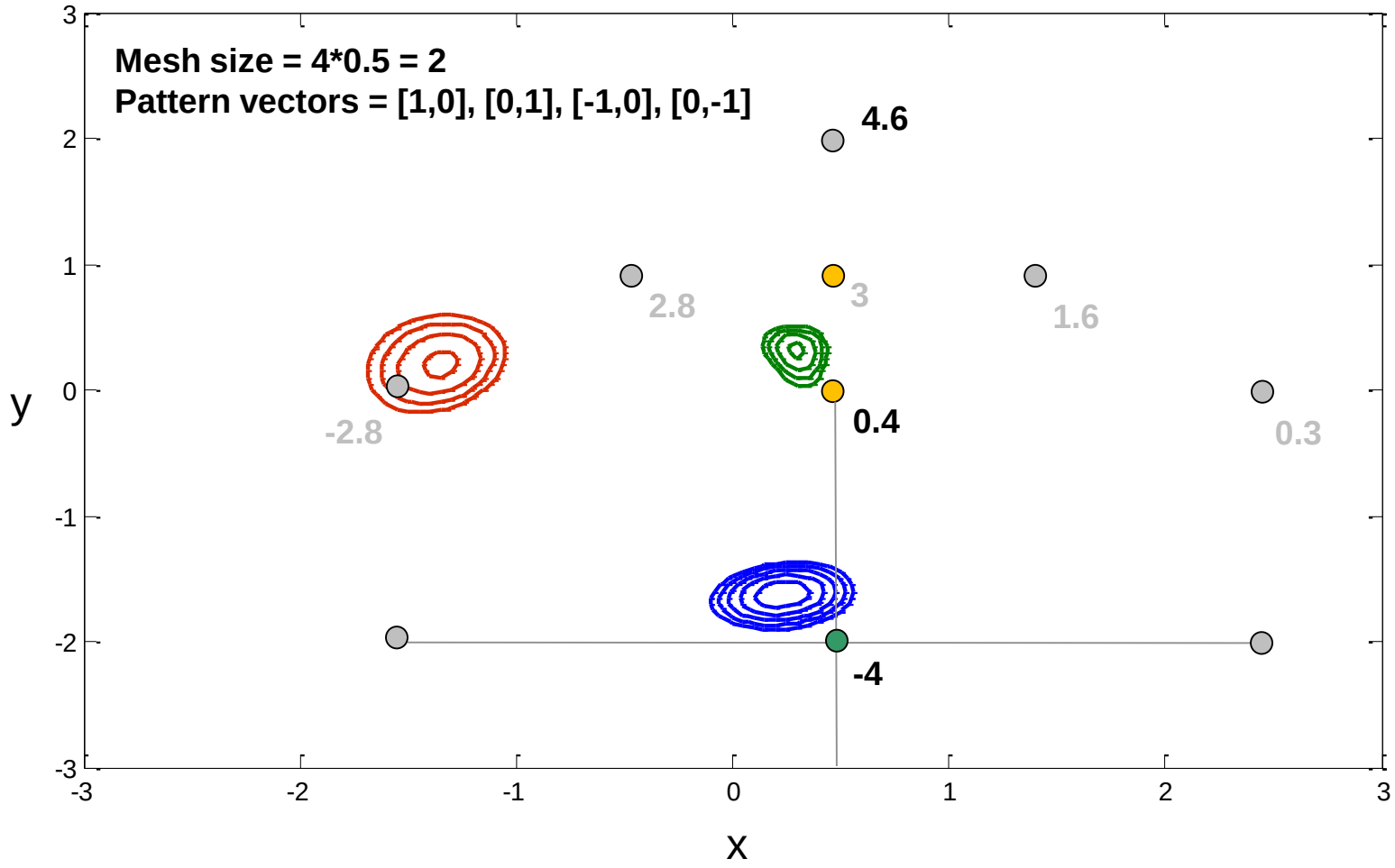
Pattern Search Overview – Iteration 2



Pattern Search Overview – Iteration 3

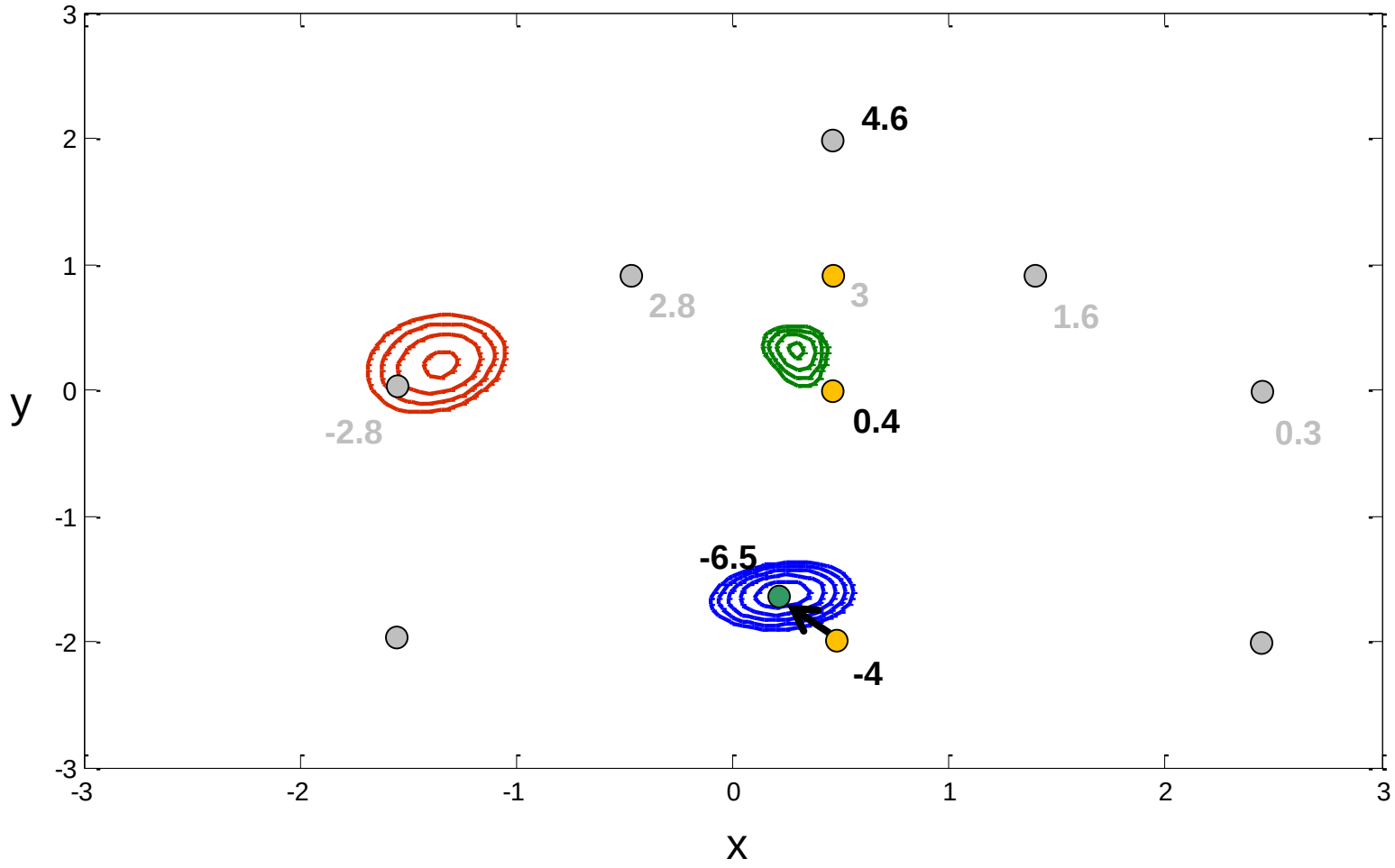


Pattern Search Overview – Iteration 4

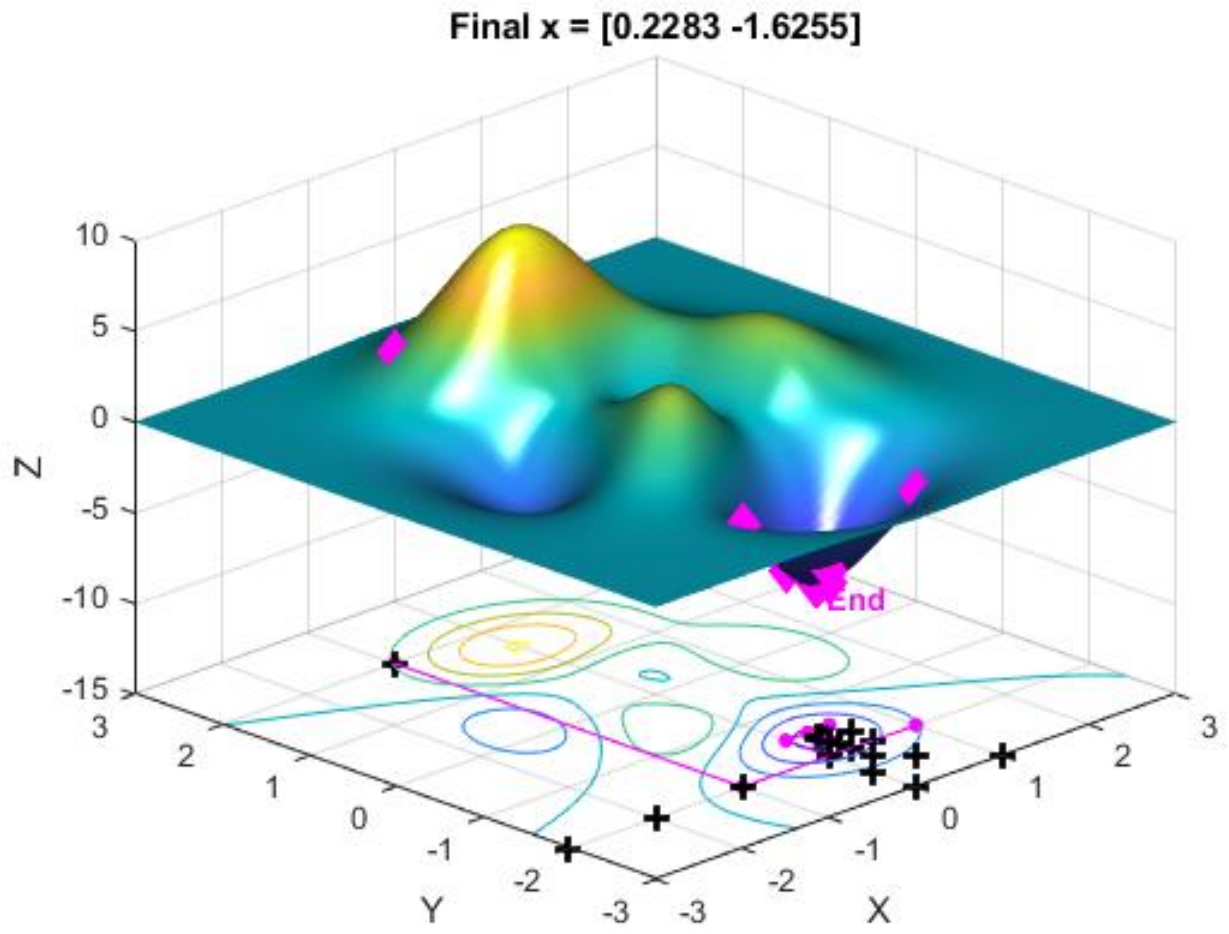


Pattern Search Overview – Iteration N

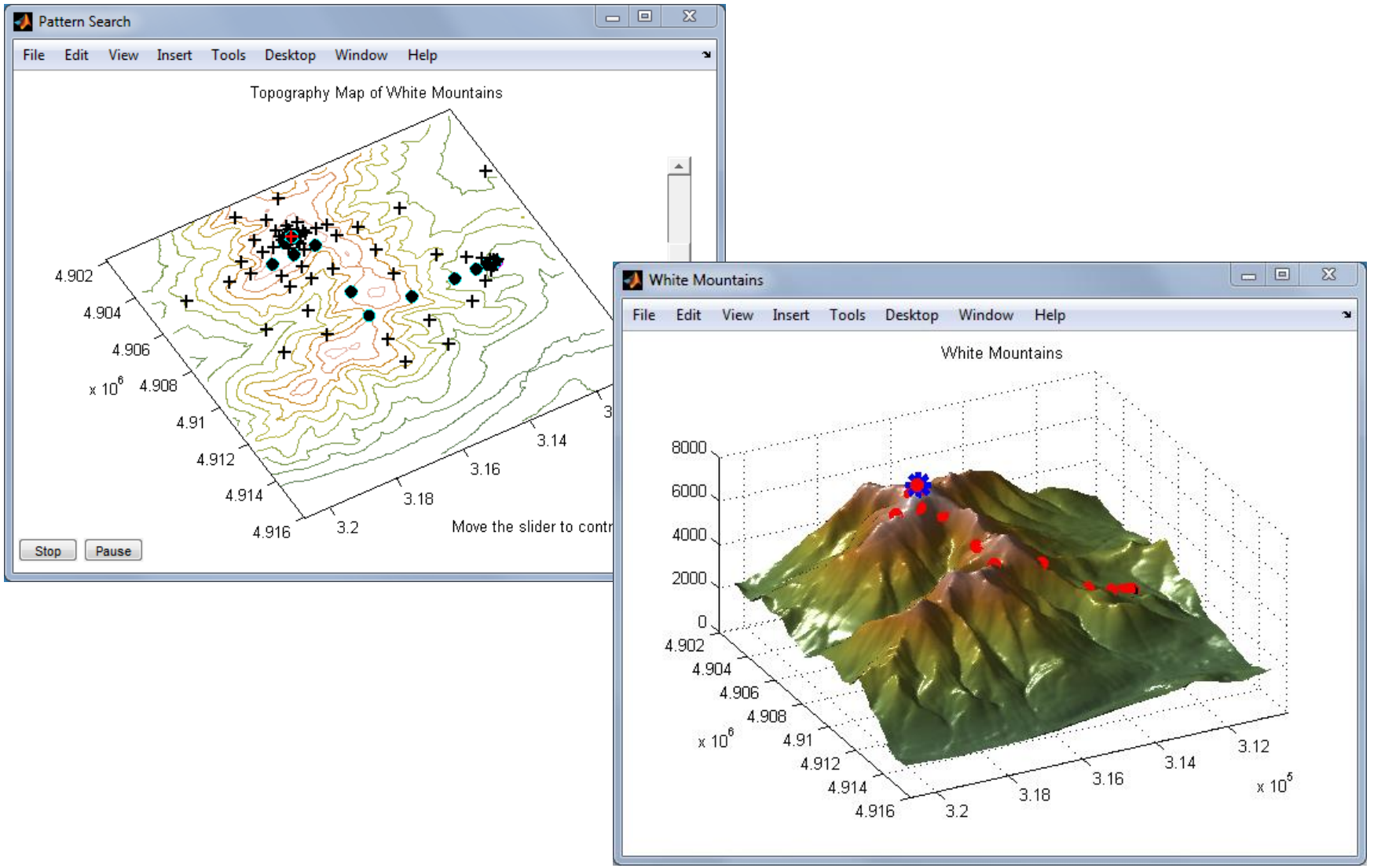
Continue expansion/contraction until convergence…



Pattern Search – Peaks Function



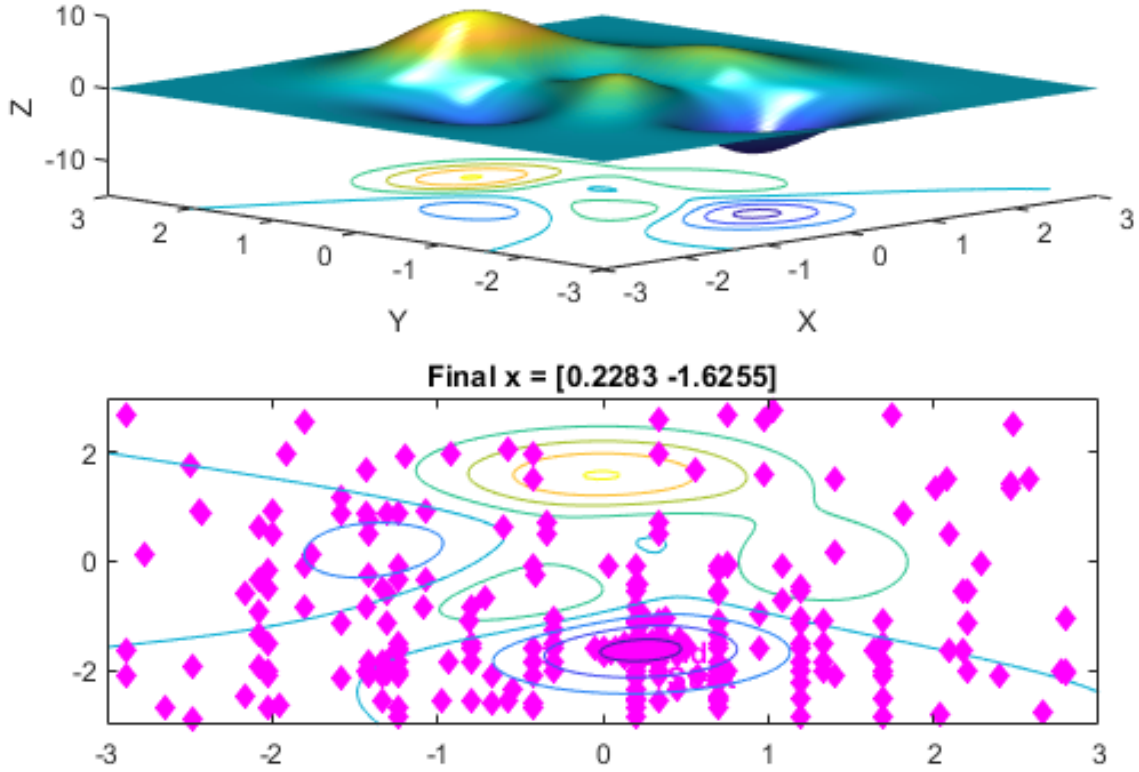
Pattern Search Climbs Mount Washington



GENETIC ALGORITHM

What is a Genetic Algorithm?

- Uses concepts from *evolutionary biology*
- Start with an initial generation of candidate solutions that are tested against the objective function
- Subsequent generations evolve from the 1st through *selection*, *crossover* and *mutation*



How Evolution Works – Binary Case

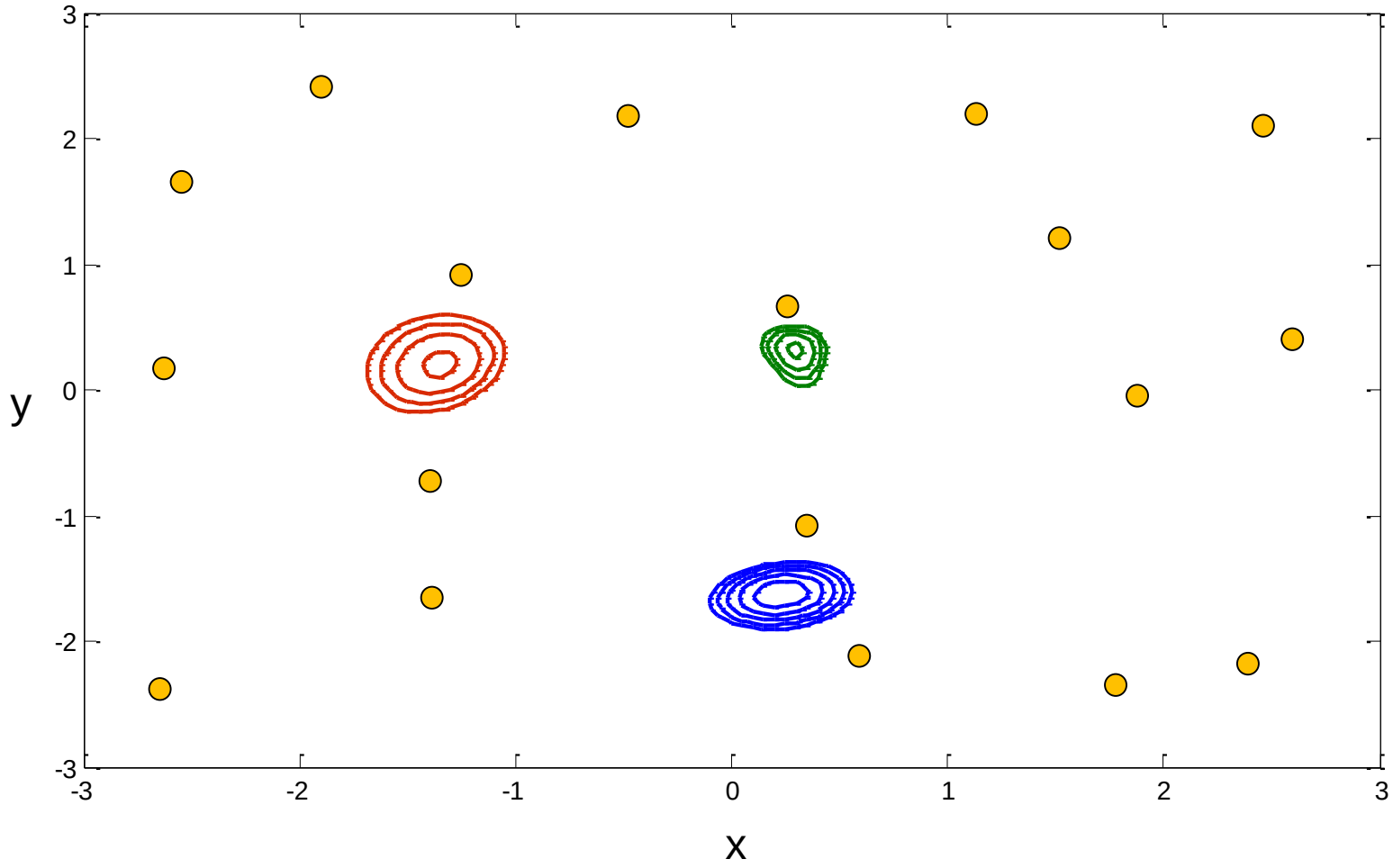
- Selection
  - *Retain* the best performing bit strings from one generation to the next. *Favor these for reproduction*
  - parent1 = [1 0 1 0 0 1 1 0 0 0]
  - parent2 = [1 0 0 1 0 0 1 0 1 0]

- Crossover
  - parent1 = [1 0 1 0 0 1 1 0 0 0]
  - parent2 = [1 0 0 1 0 0 1 0 1 0]
  - child = [1 0 0 0 0 1 1 0 1 0]

- Mutation
  - parent = [1 0 1 0 0 1 1 0 0 0]
  - child = [0 1 0 1 0 1 0 0 0 1]

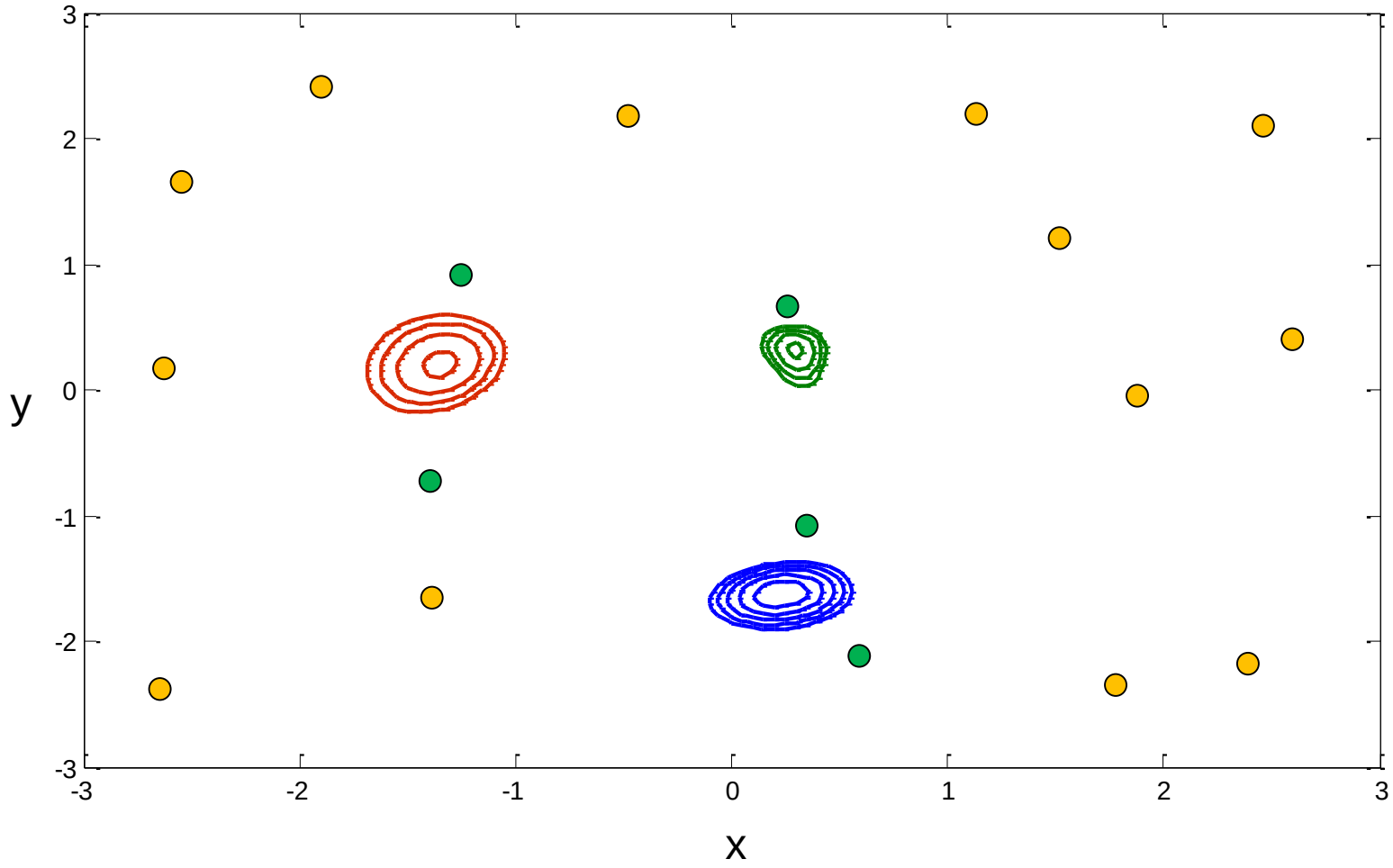
Genetic Algorithm – Iteration 1

Evaluate initial population



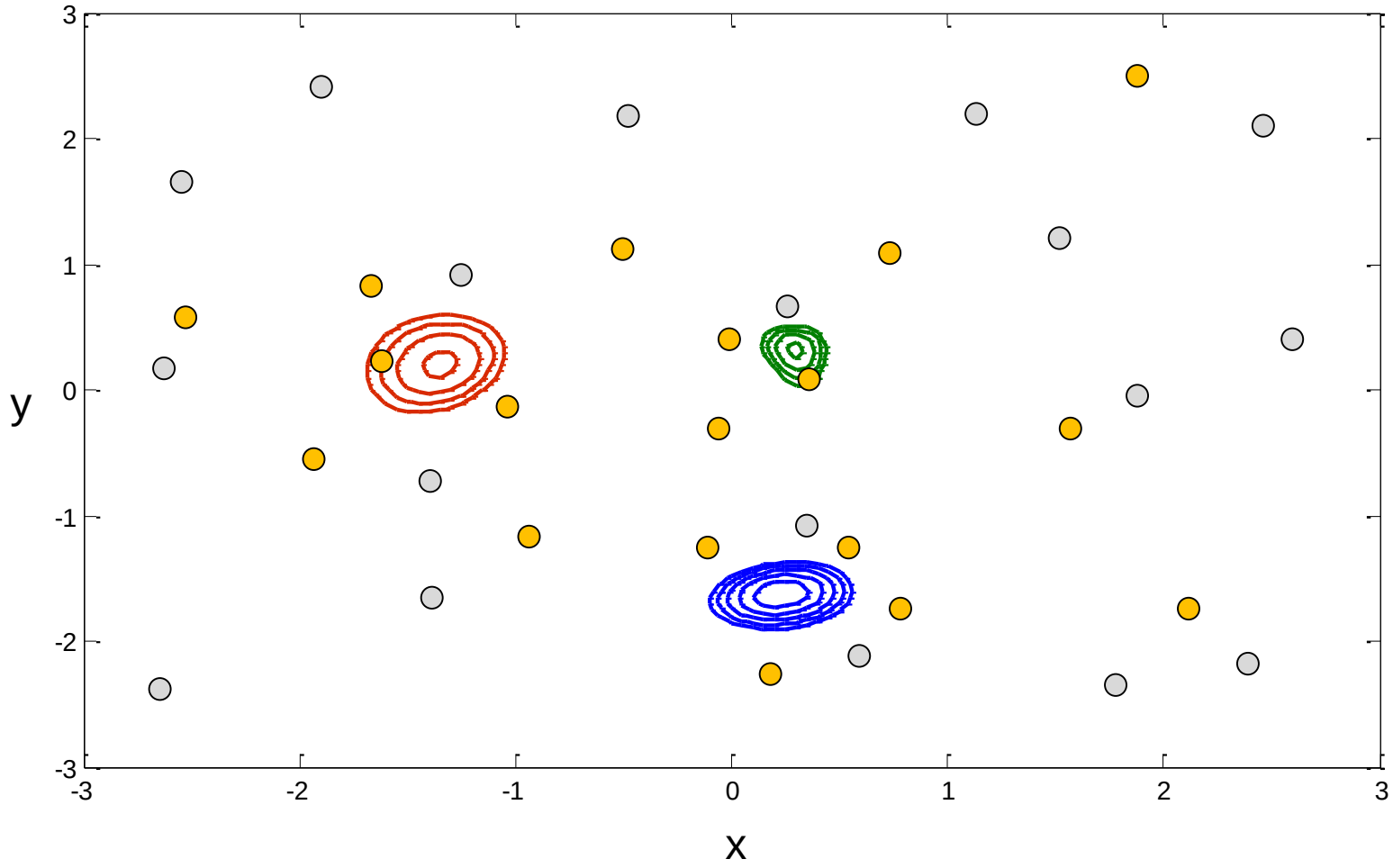
Genetic Algorithm – Iteration 1

Select a few good solutions for reproduction

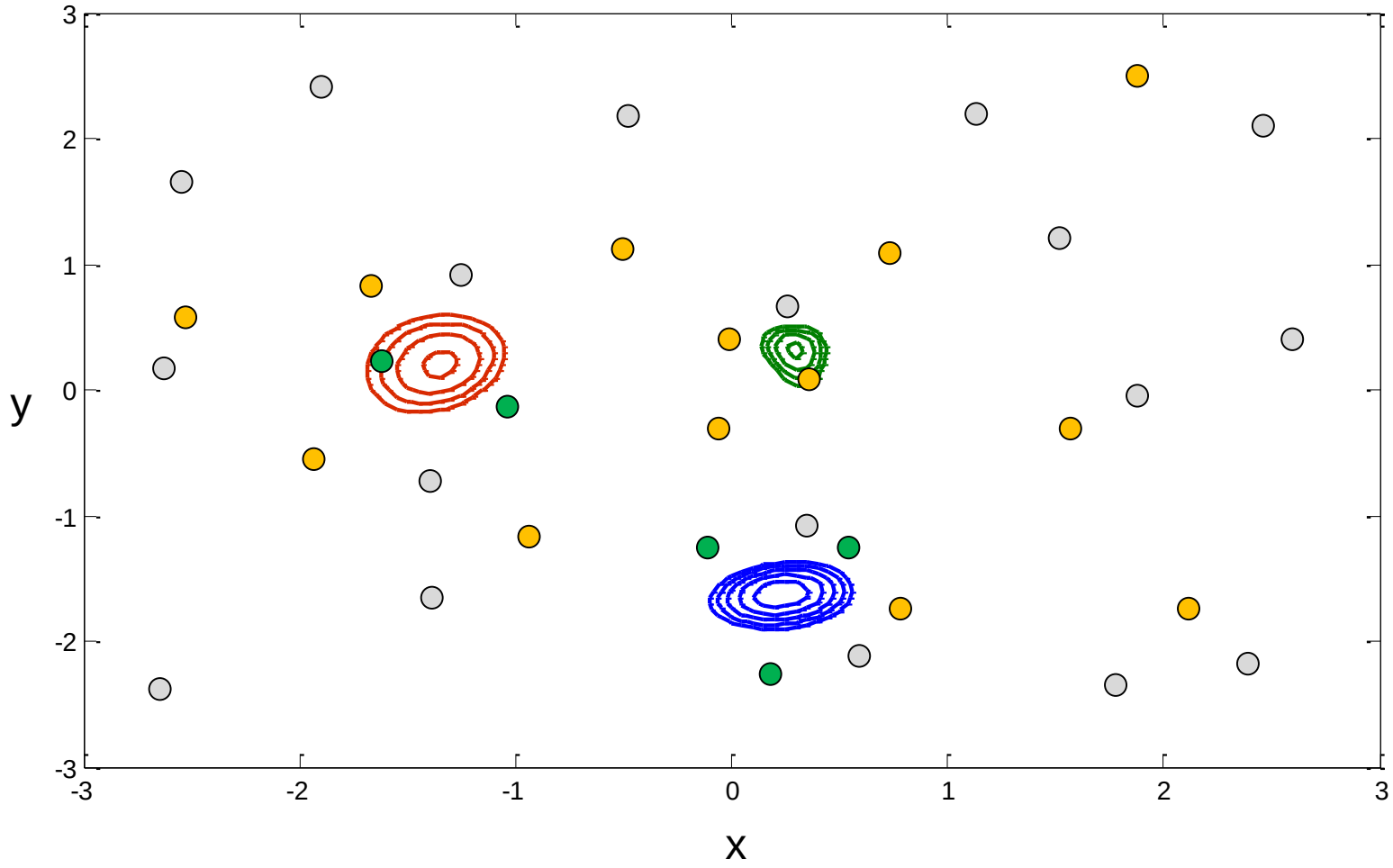


Genetic Algorithm – Iteration 2

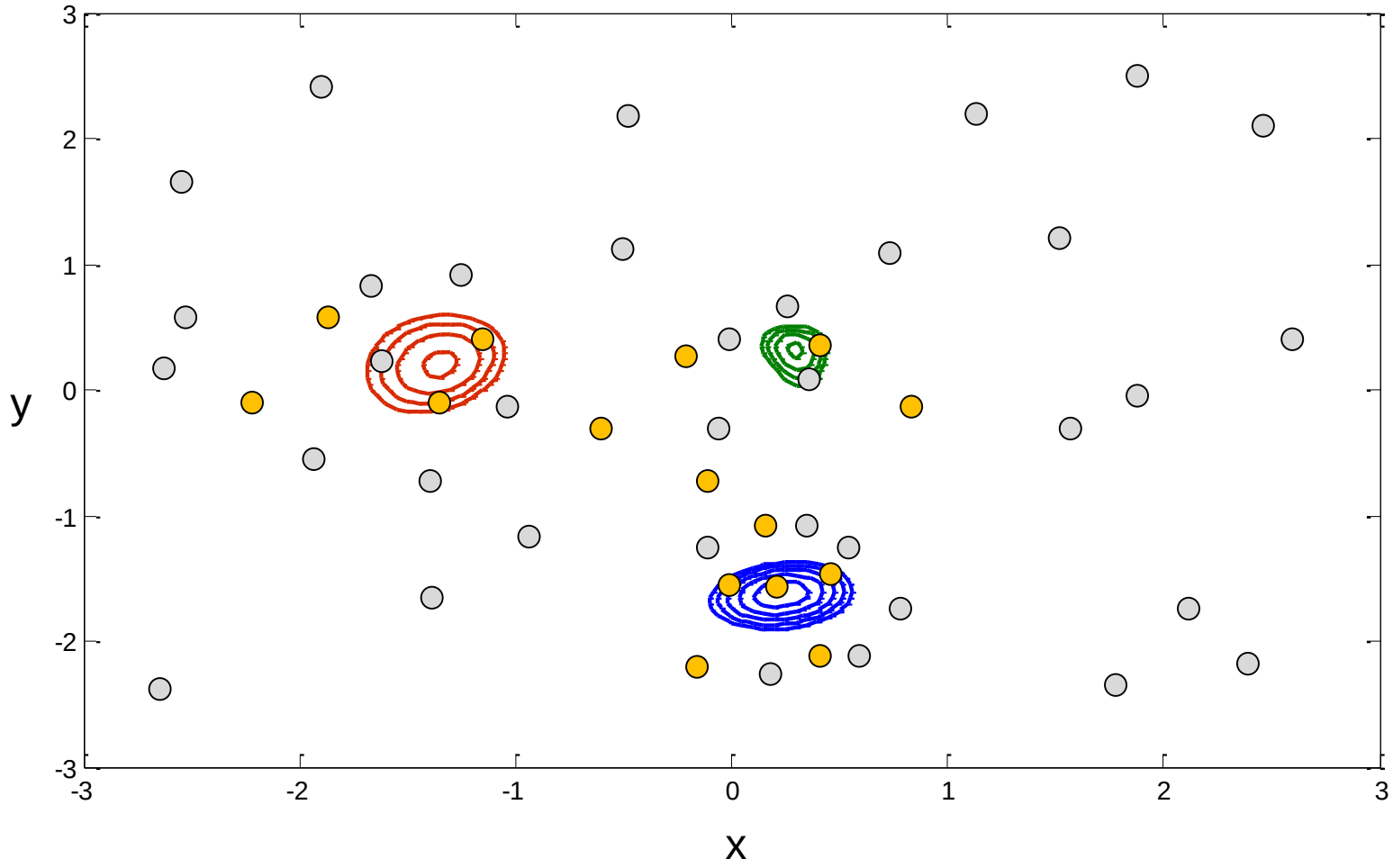
Generate new population and evaluate



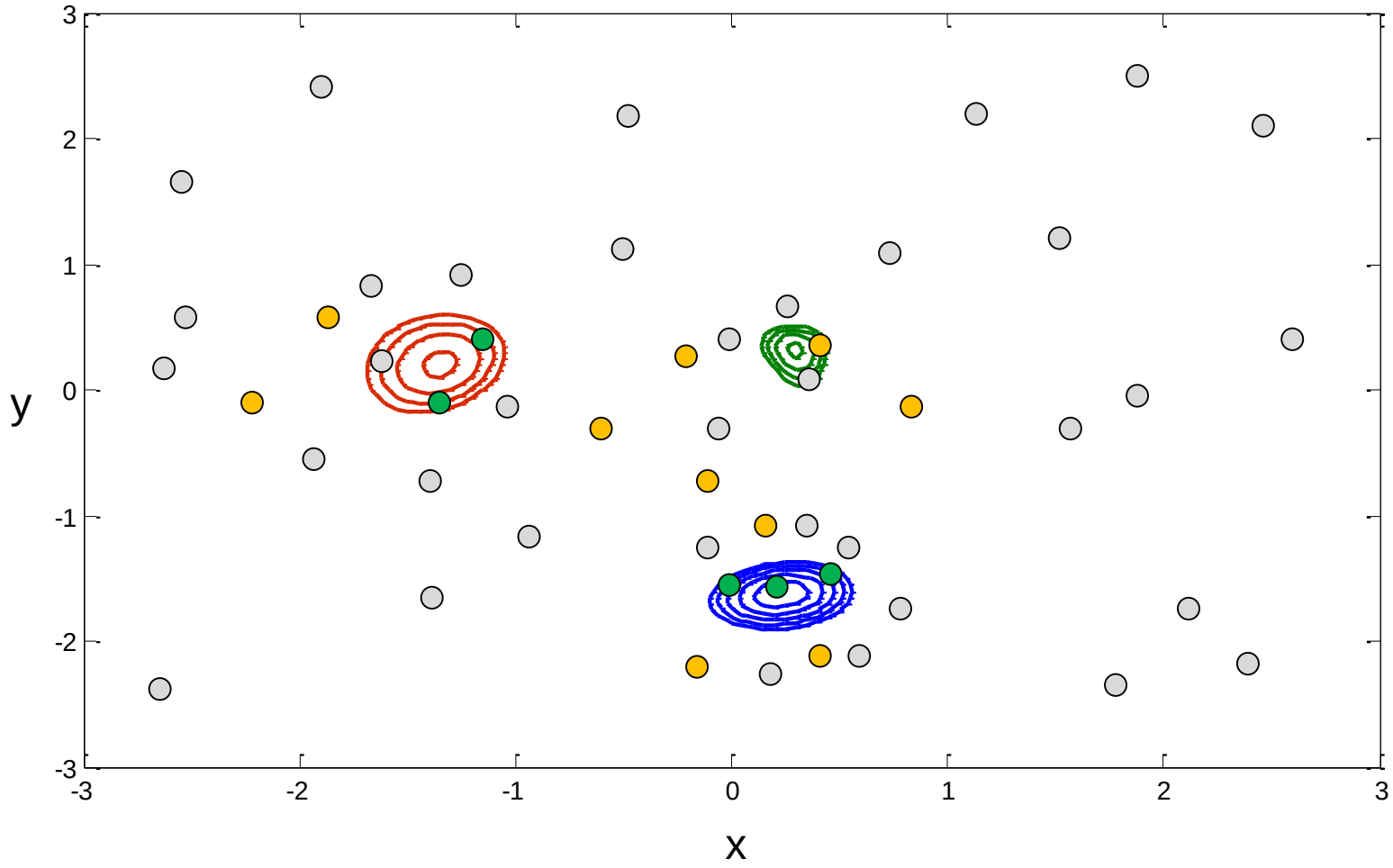
Genetic Algorithm – Iteration 2



Genetic Algorithm – Iteration 3

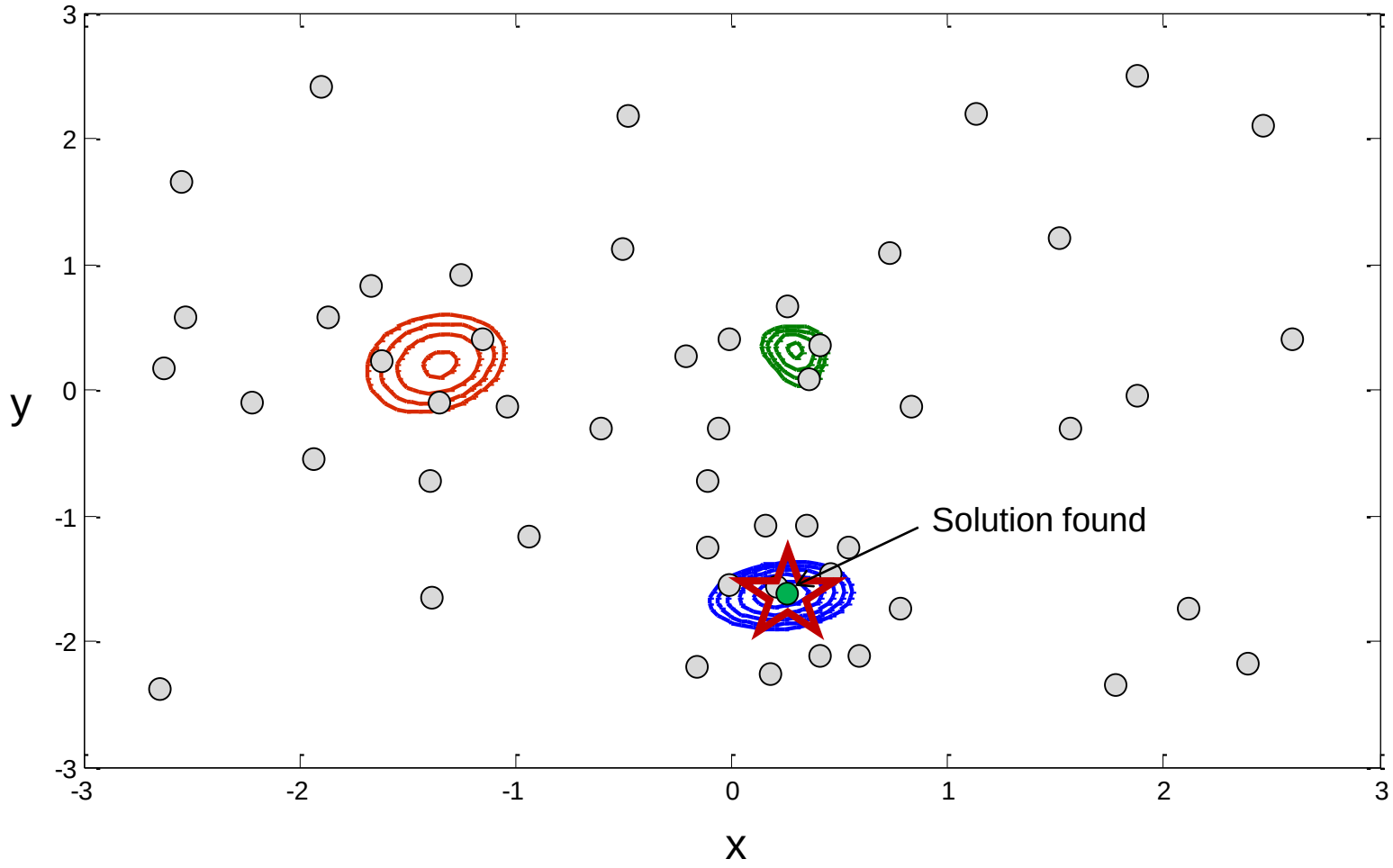


Genetic Algorithm – Iteration 3

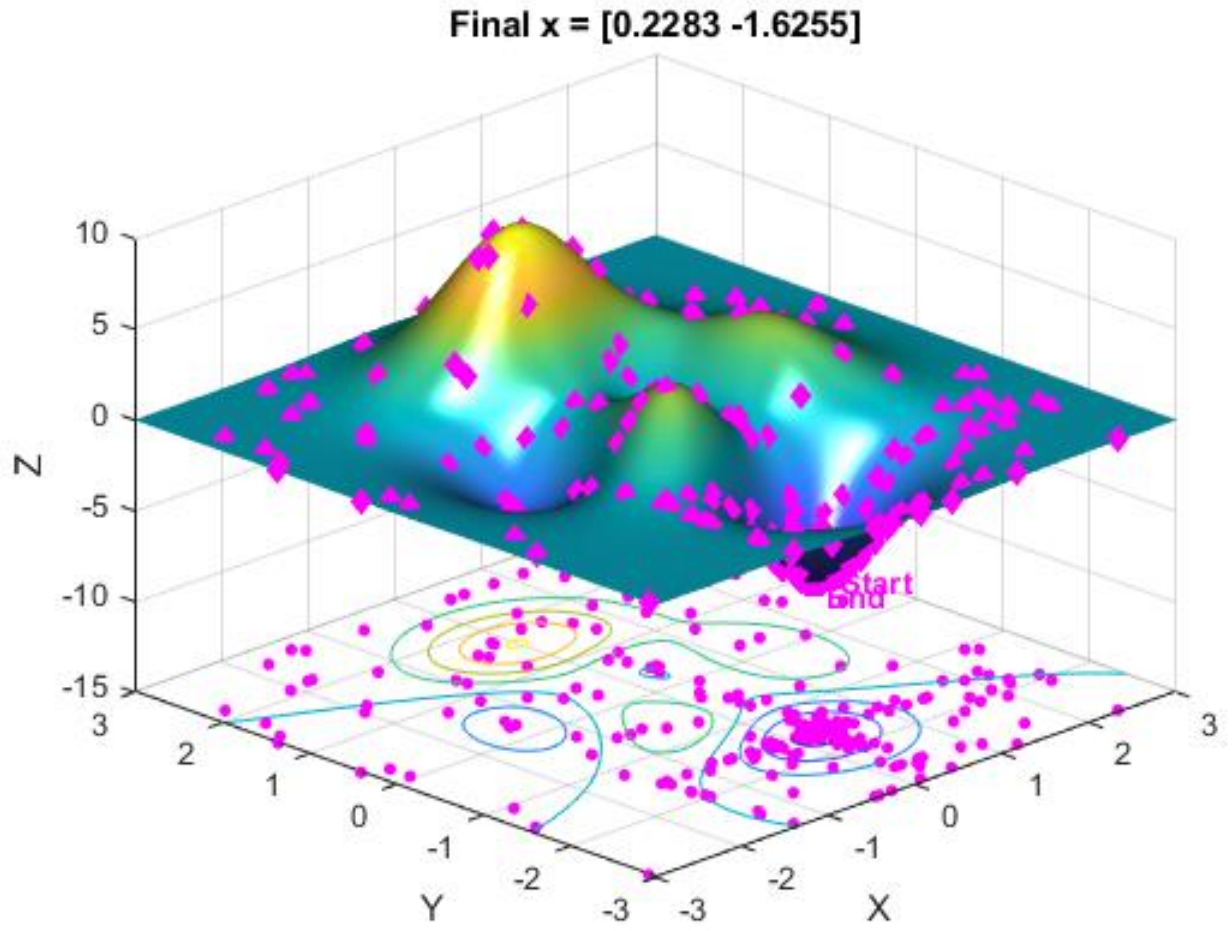


Genetic Algorithm – Iteration N

Continue process until stopping criteria are met



Genetic Algorithm – Peaks Function



Genetic Algorithm – Integer Constraints

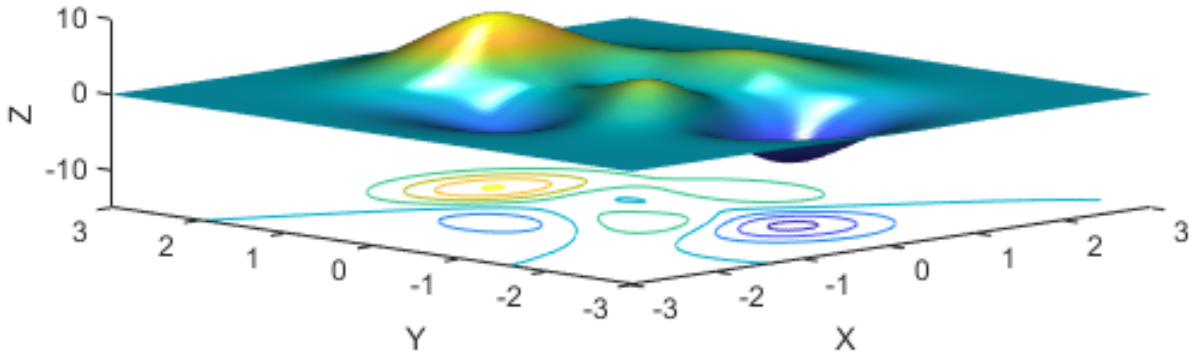
Mixed Integer Optimization

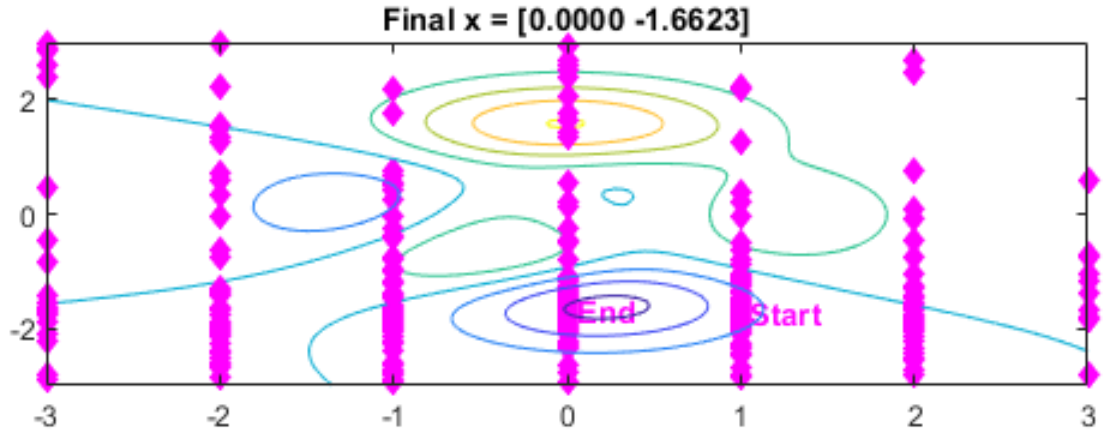
$$\min_x f(x)$$

s.t. some *x* are integers

Examples

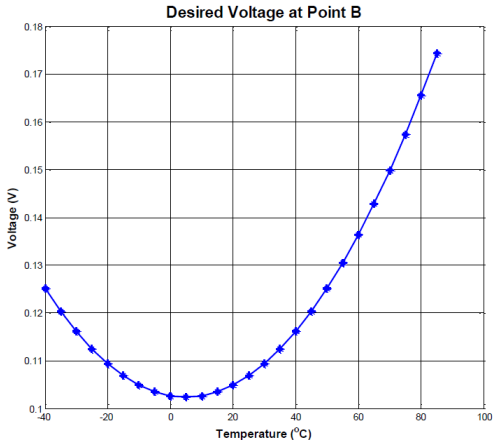
- Only certain sizes of components available
- Can only purchase whole shares of stock

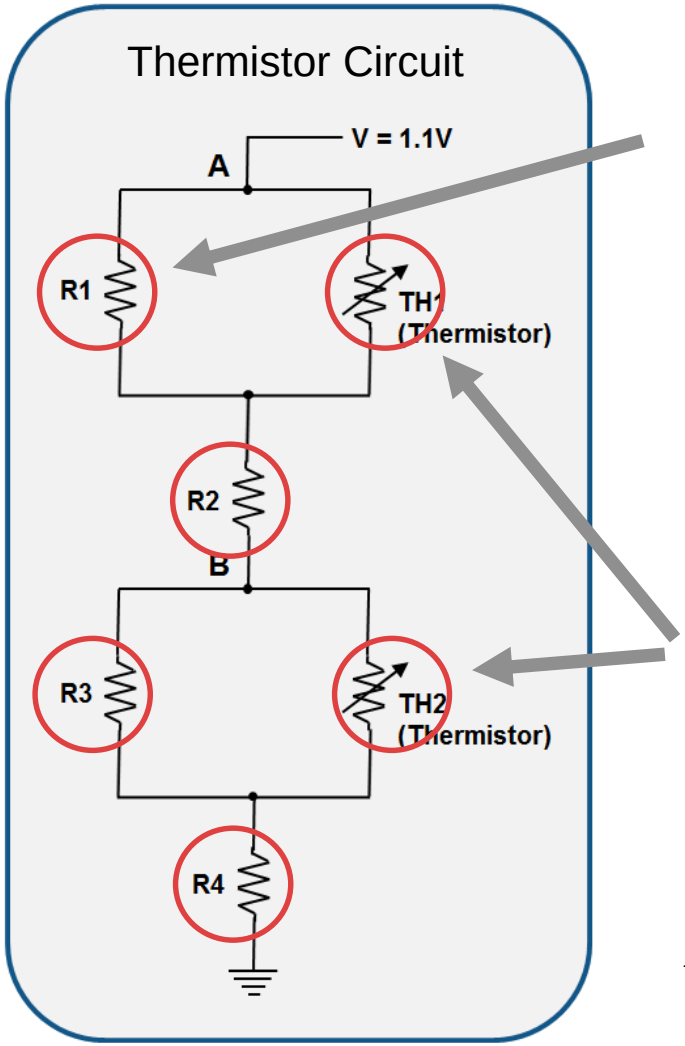




Application: Circuit Component Selection

- 6 components to size
- Only certain sizes available
- Objective:
  - Match Voltage vs. Temperature curve





300 Ω
330 Ω
360 Ω
…
180k Ω
200k Ω
220k Ω

?

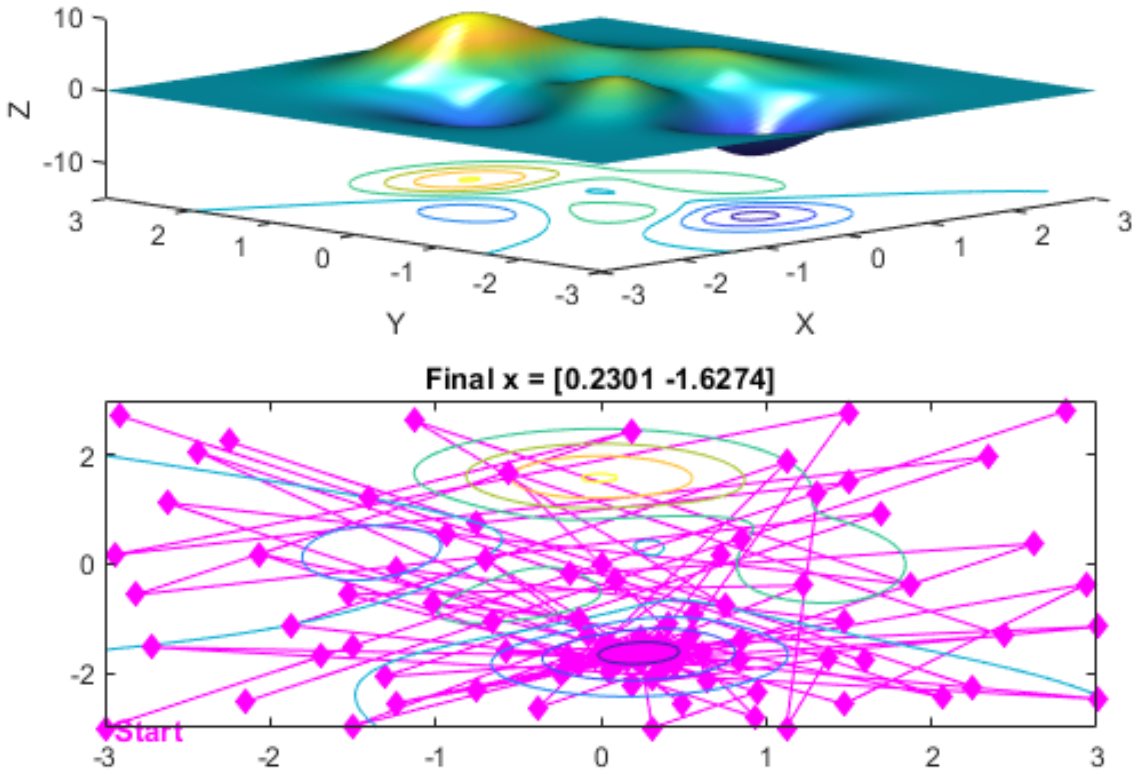
Thermistors:
Resistance varies nonlinearly with temperature

$$R_{TH} = \frac{R_{TH,Nom}}{e^{\beta\frac{T-T_{Nom}}{T*T_{Nom}}}}$$

SURROGATE OPTIMIZATION

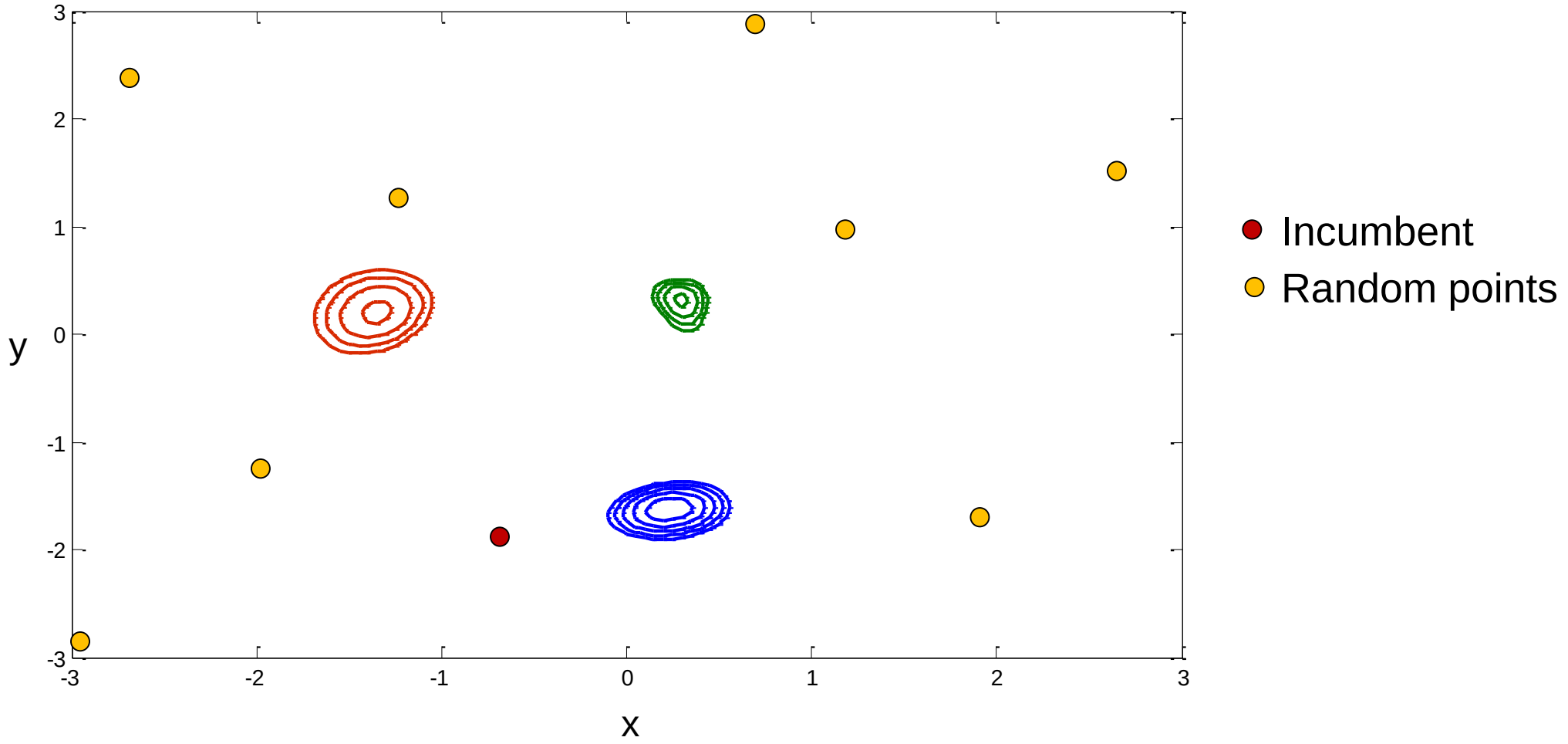
What is Surrogate Optimization?

- An approach that creates and optimizes a surrogate of the function
- Searches randomly to explore and adaptively to refine
- Does not rely on gradients: works on smooth and nonsmooth problems



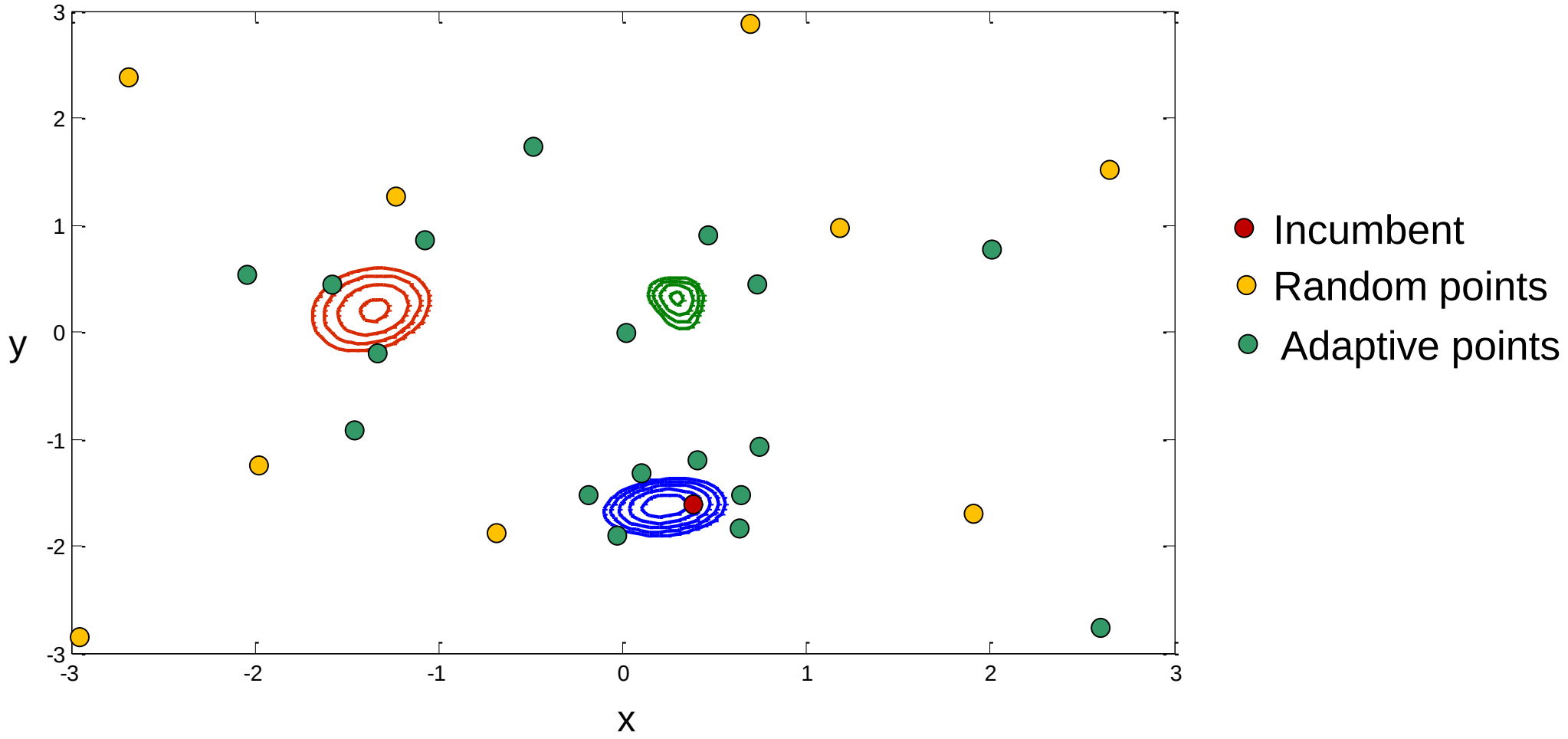
Surrogate Optimization Overview

Construction phase – evaluate at random points to construct surrogate



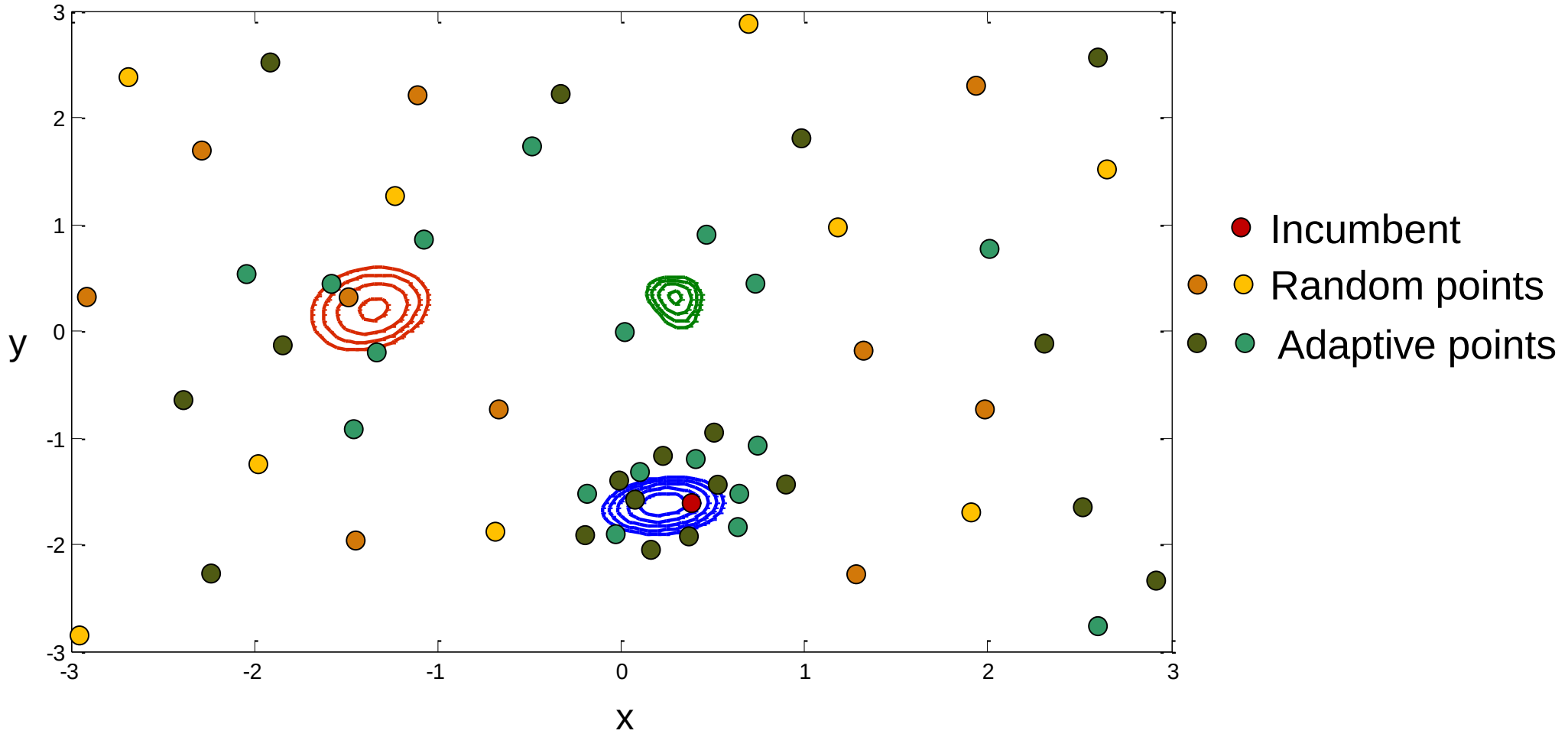
Surrogate Optimization Overview

Search phase – minimize merit function of surrogate and point spread



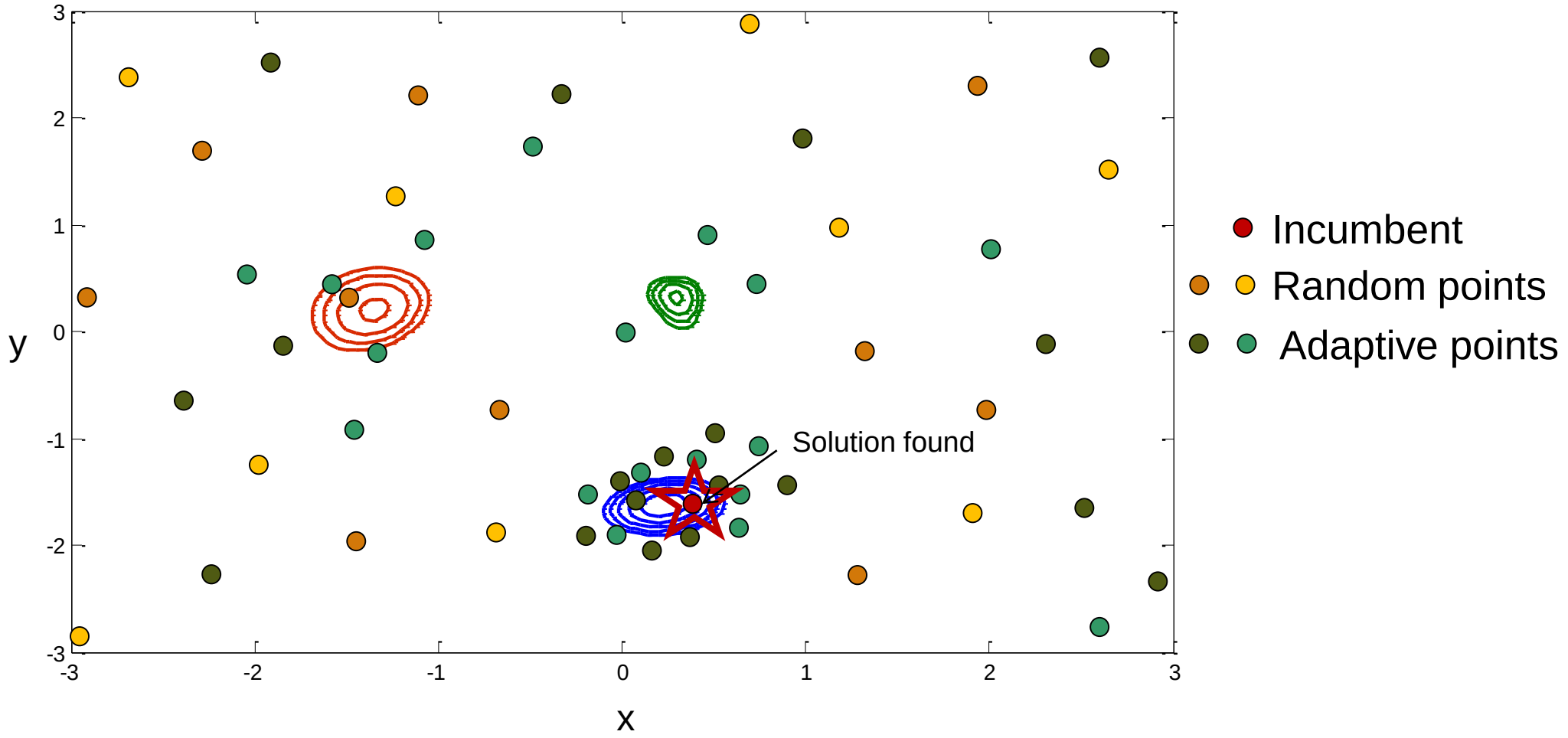
Surrogate Optimization Overview

Reset – Repeat construction and search phases

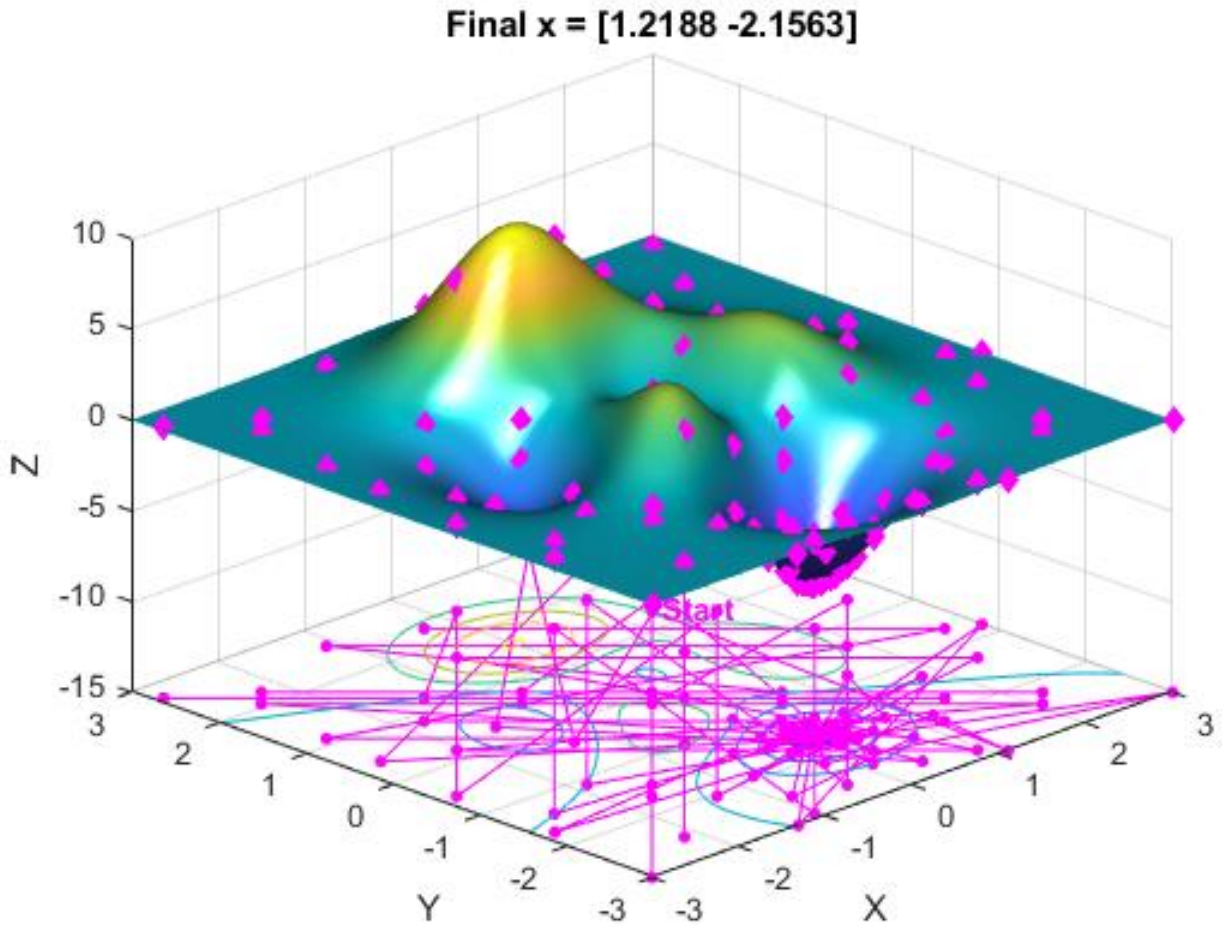


Surrogate Optimization Overview

Continue process until stopping criteria are met



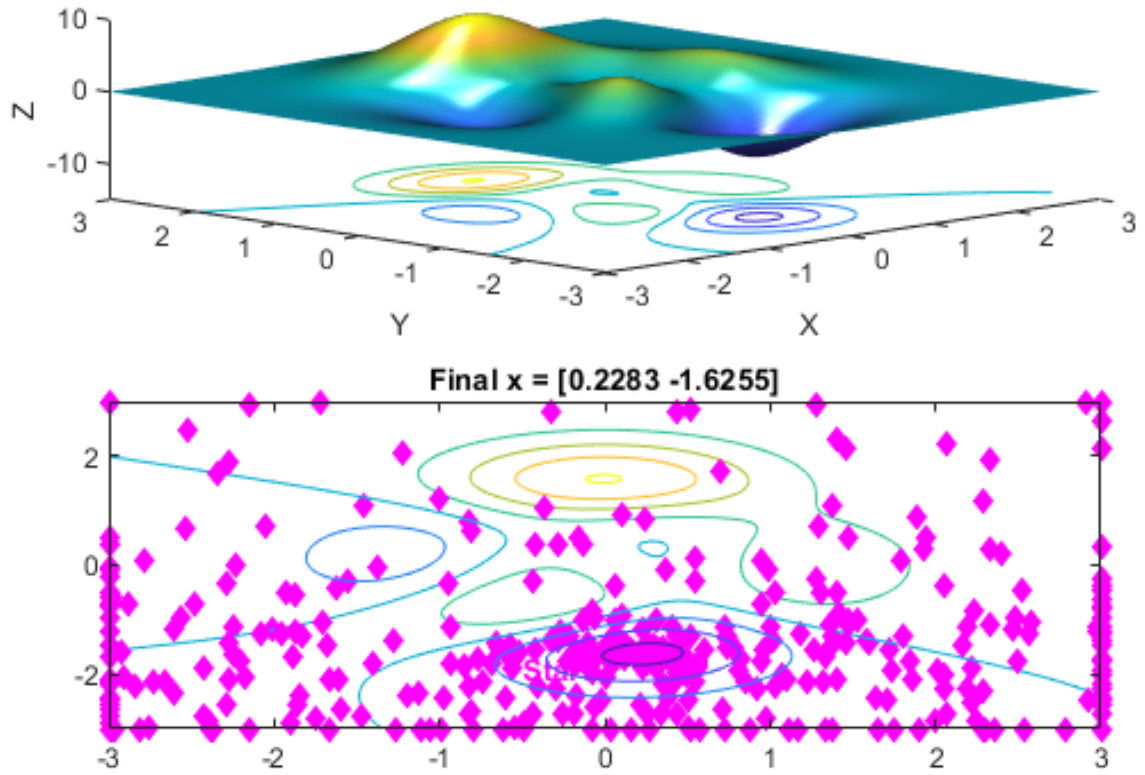
Surrogateopt Demo – Peaks Function



PARTICLE SWARM

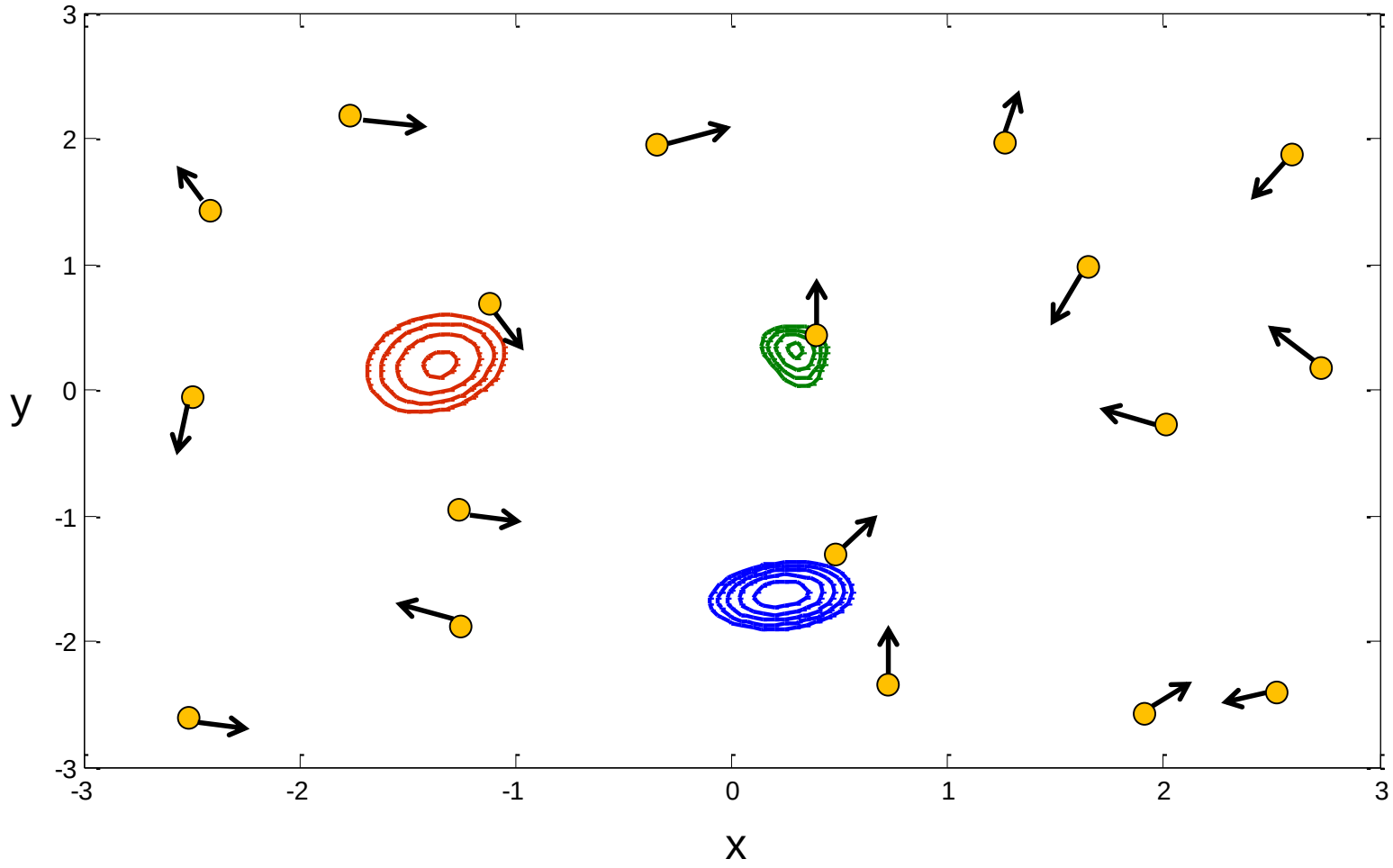
What is Particle Swarm Optimization?

- A collection of particles move throughout the region
- Particles have velocity and are affected by the other particles in the swarm
- Does not rely on gradients: works on smooth and nonsmooth problems



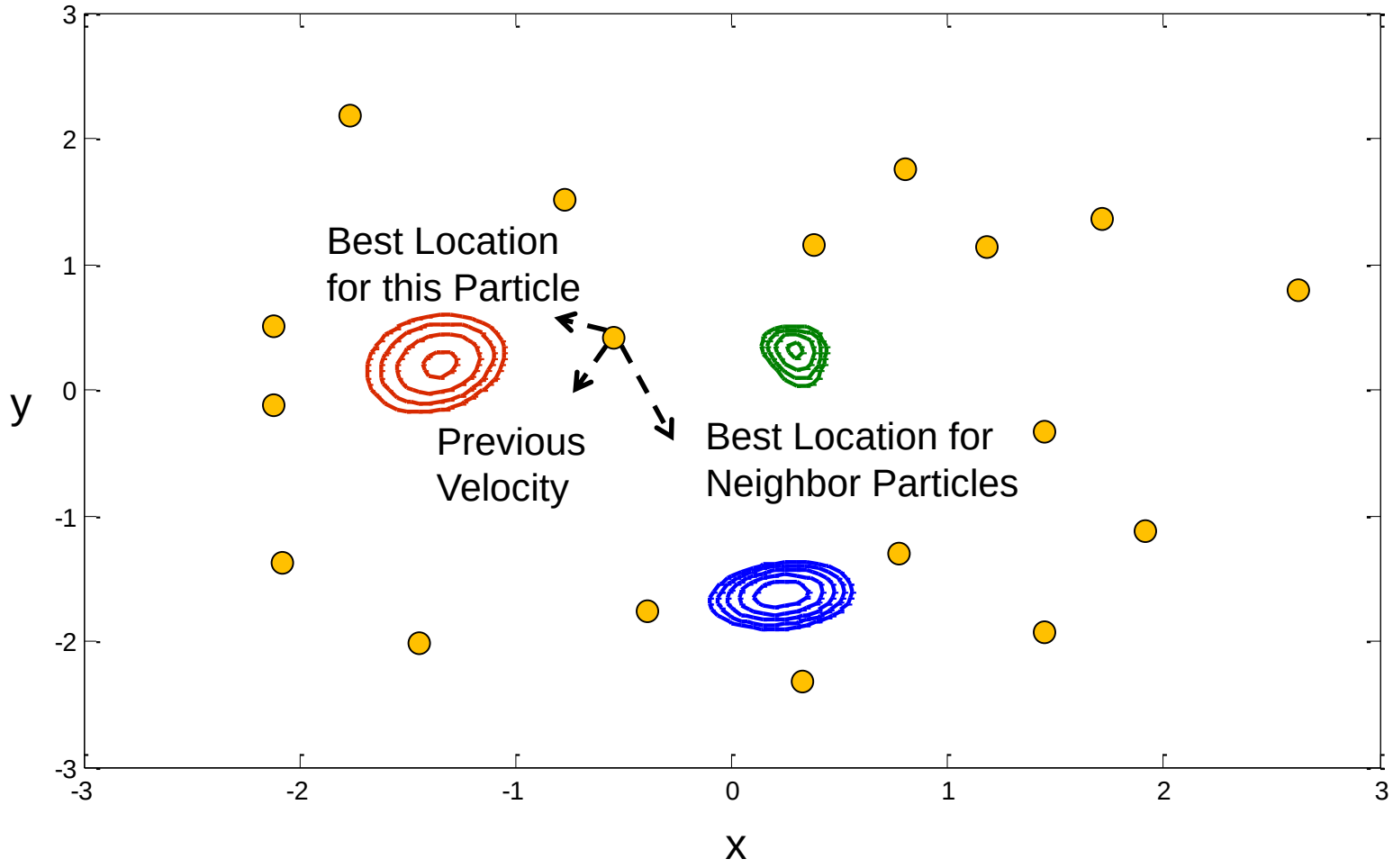
Particle Swarm Overview – Iteration 1

Initialize particle locations and velocities, evaluate all locations



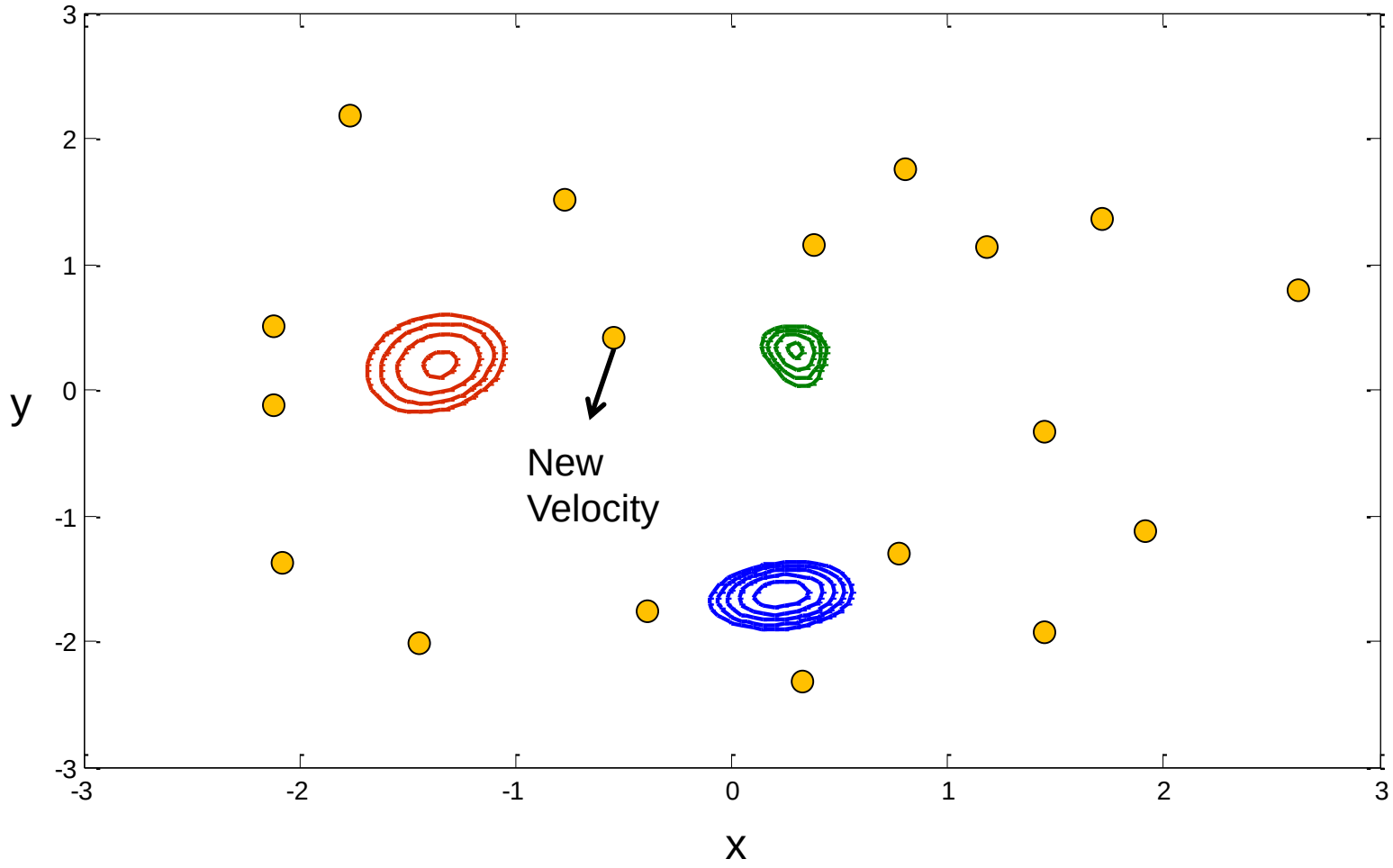
Particle Swarm Overview – Iteration N

Update velocities for each particle



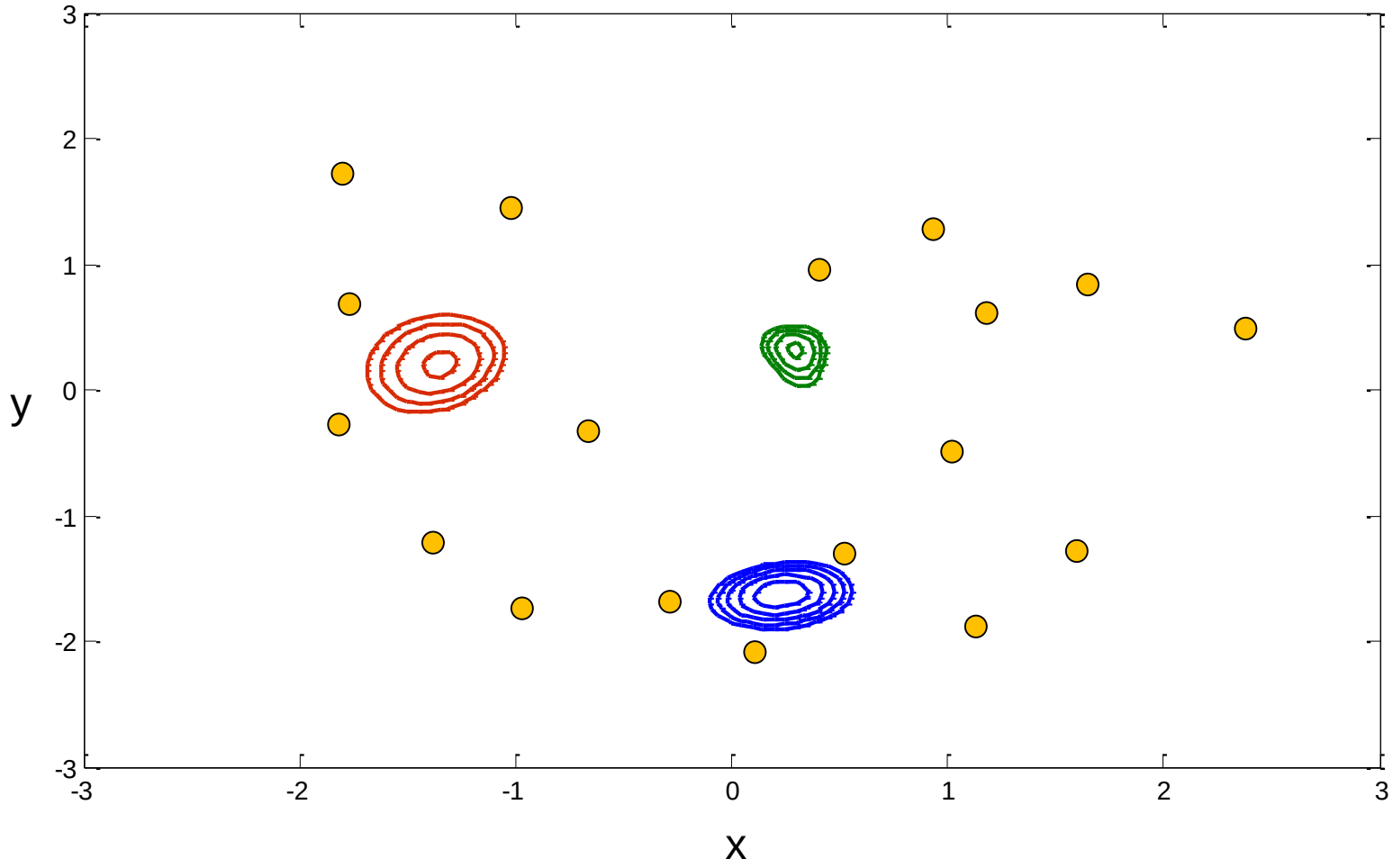
Particle Swarm Overview – Iteration N

Update velocities for each particle



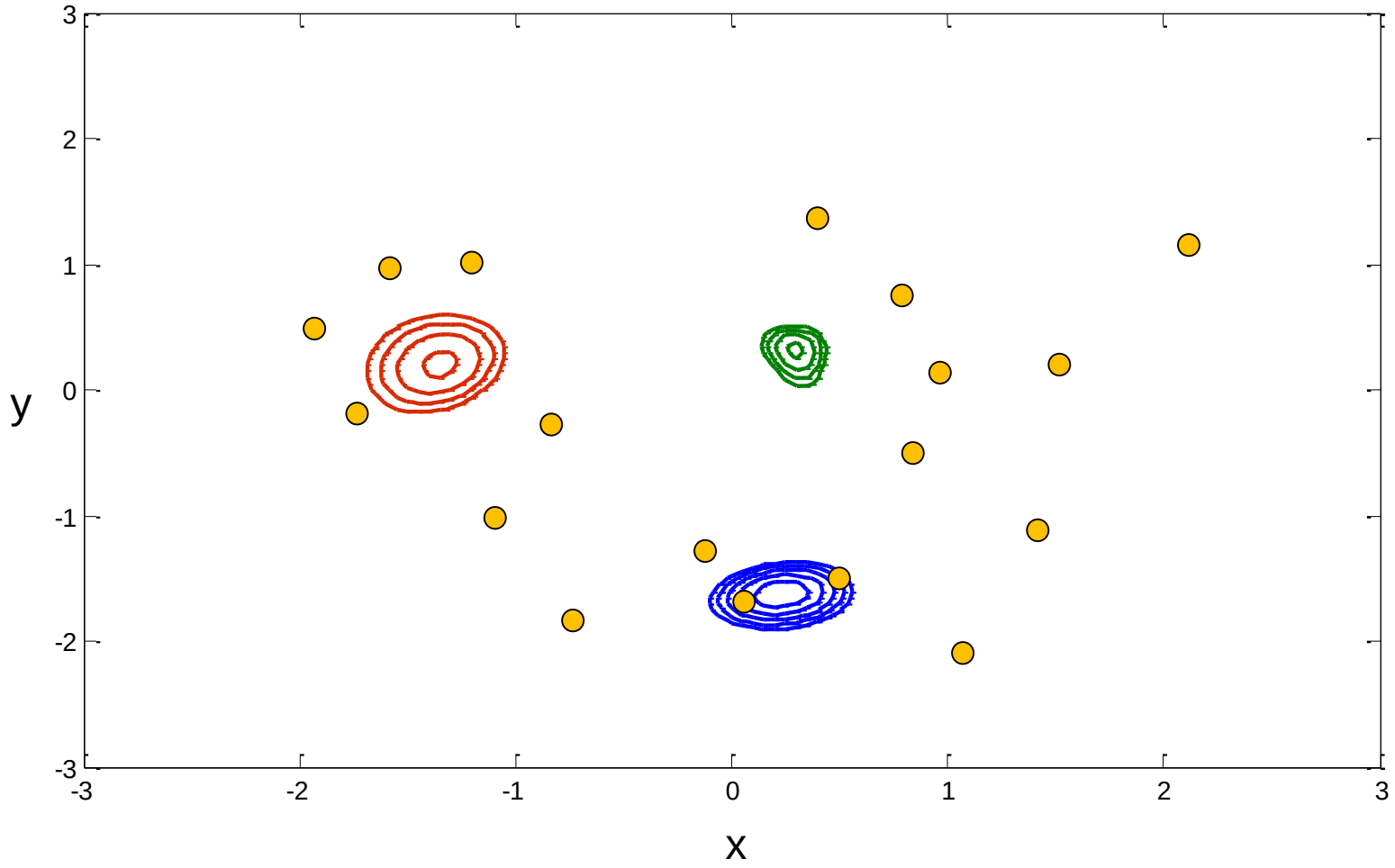
Particle Swarm Overview – Iteration N

Move particles based on new velocities

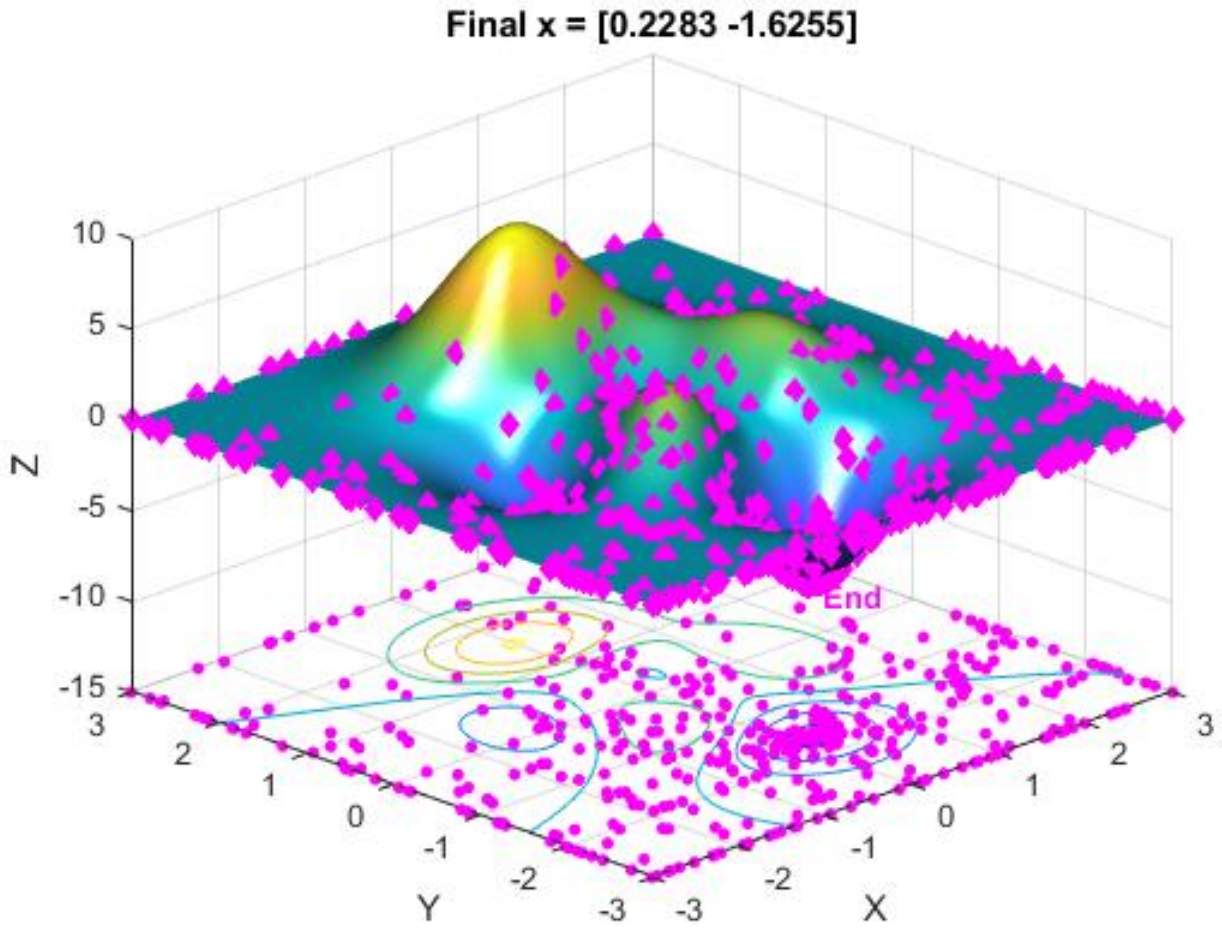


Particle Swarm Overview – Iteration N

Continue swarming until convergence



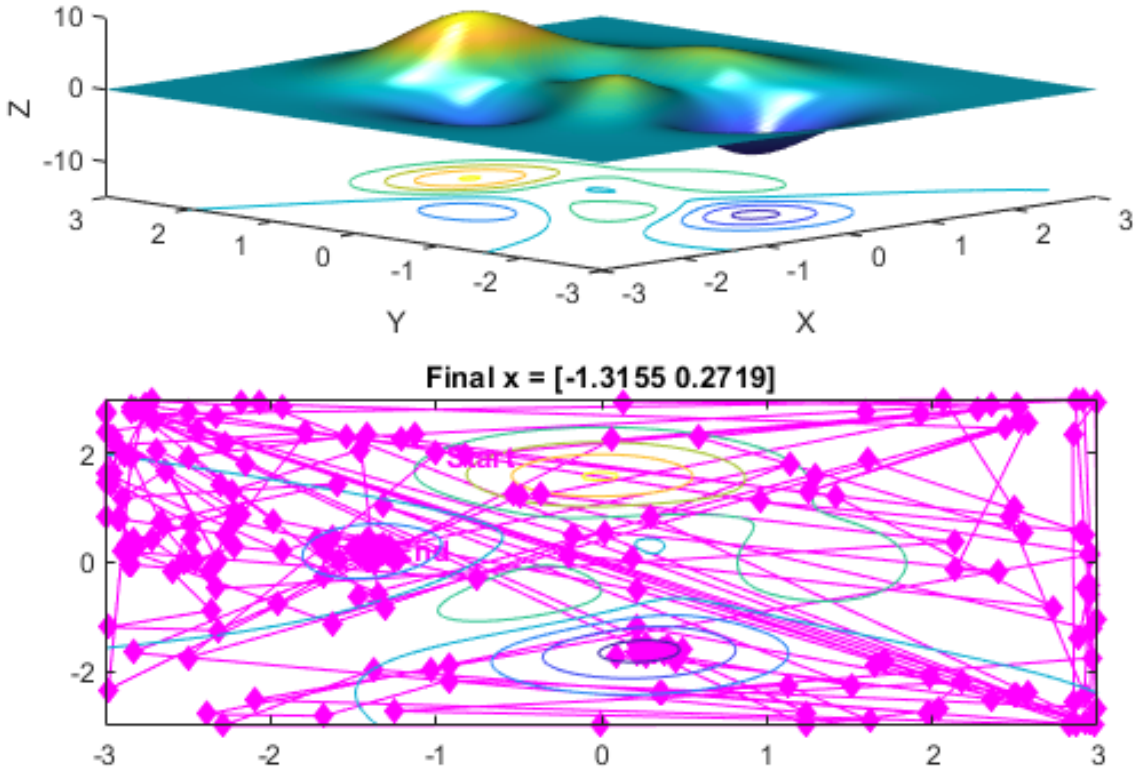
Particle Swarm – Peaks Function



SIMULATED ANNEALING

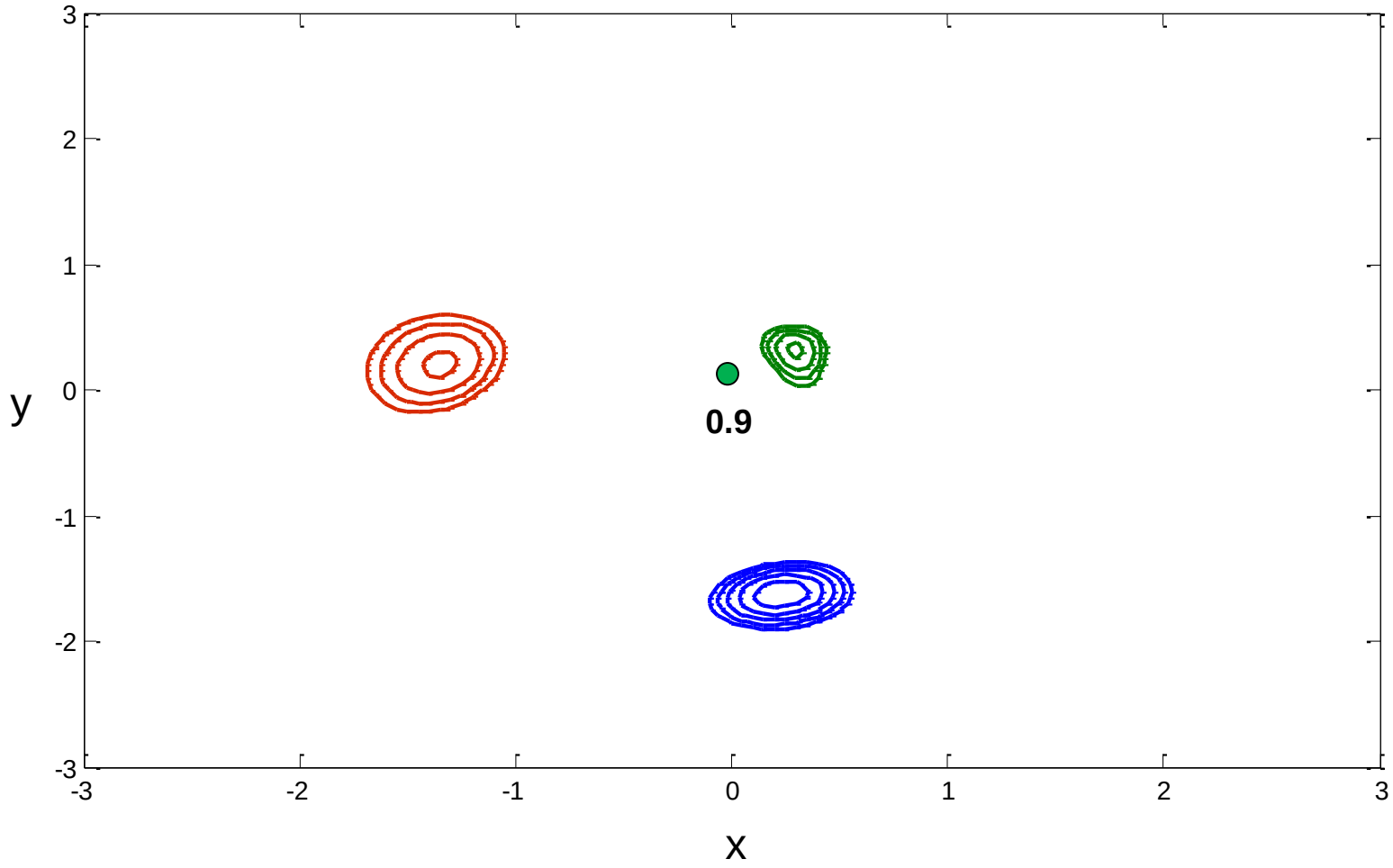
What is Simulated Annealing?

- A probabilistic metaheuristic approach based upon the physical process of annealing in metallurgy.
- Controlled cooling of a metal allows atoms to realign from a random higher energy state to an ordered crystalline (globally) lower energy state

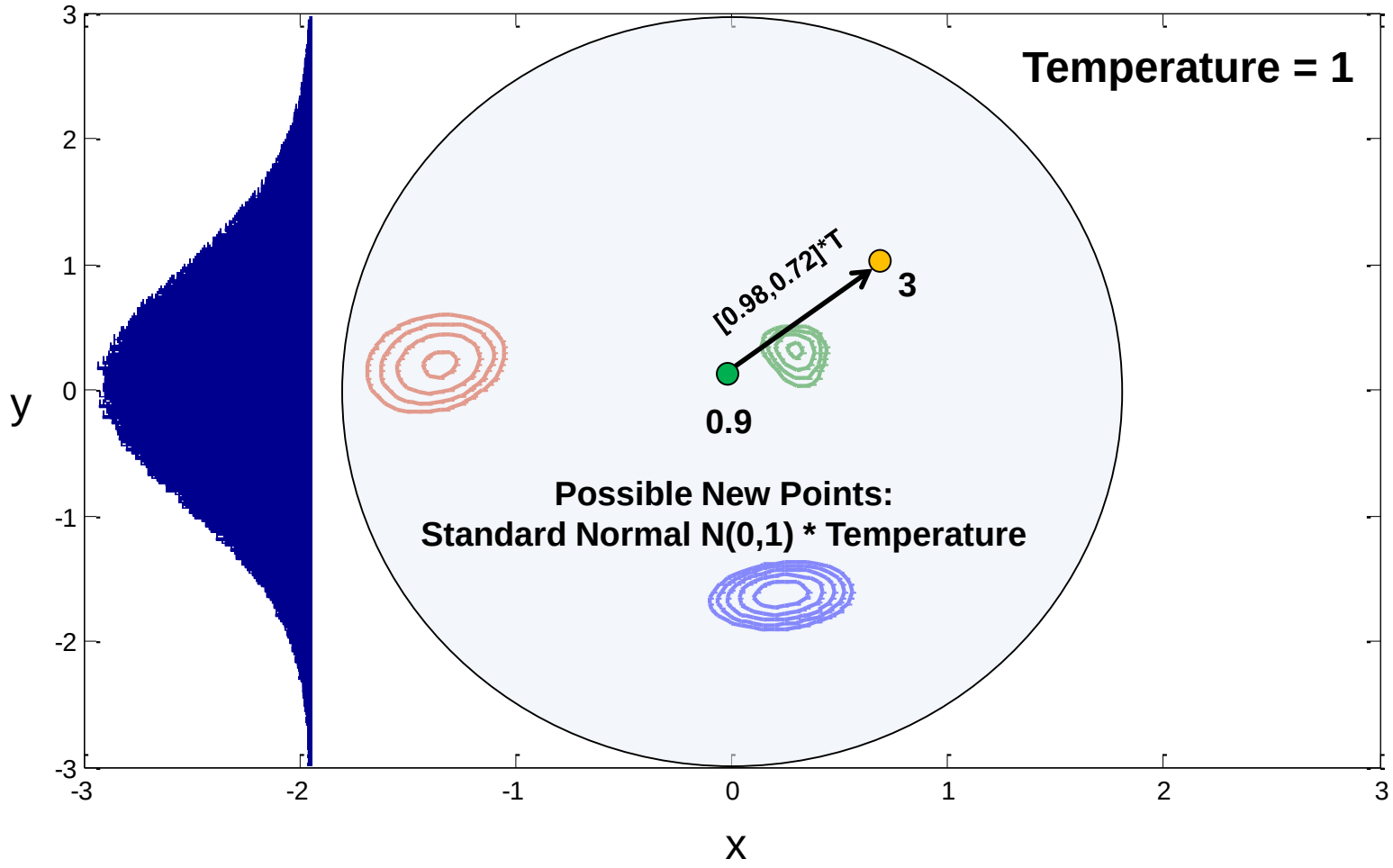


Simulated Annealing Overview – Iteration 1

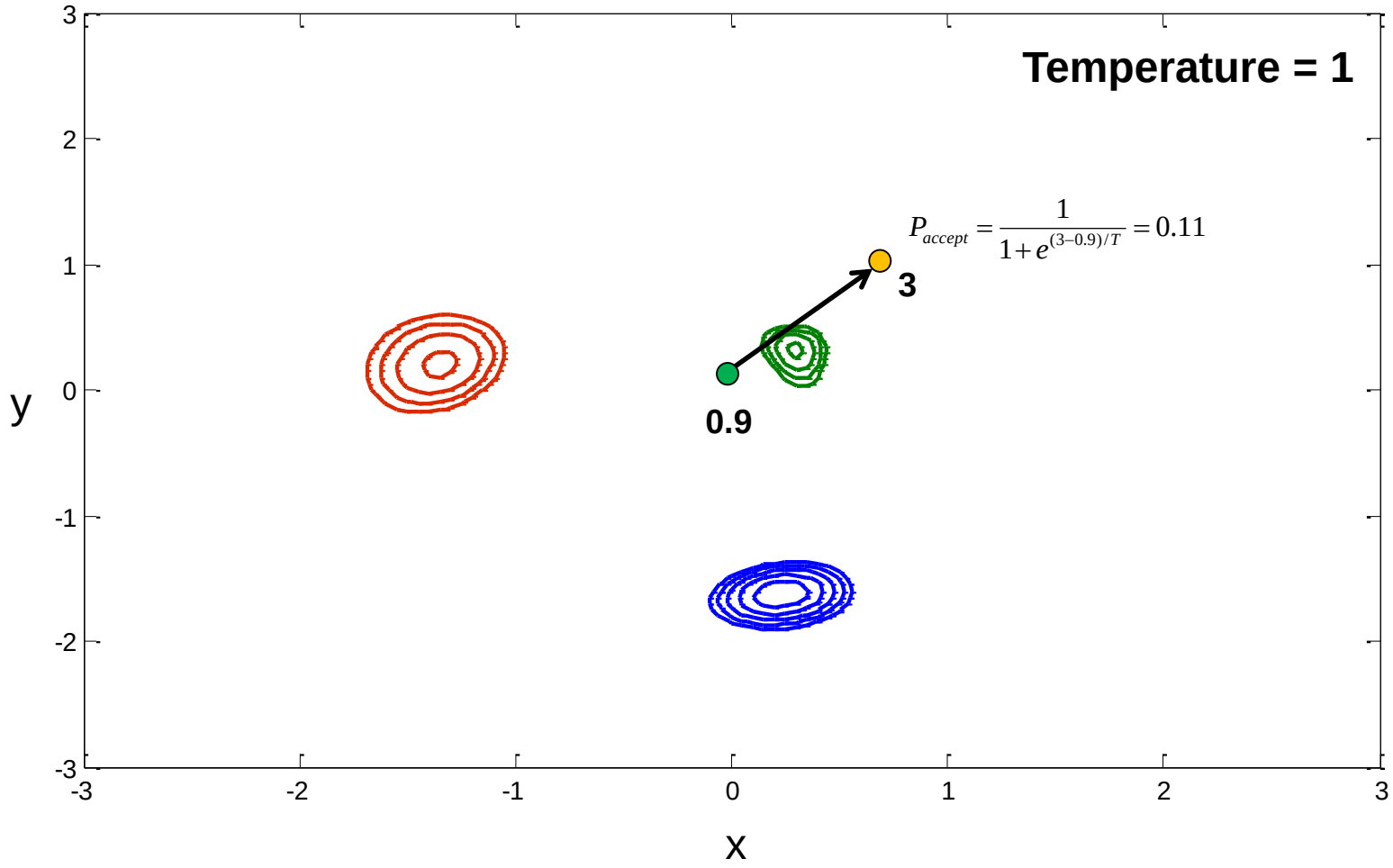
Run from specified x0



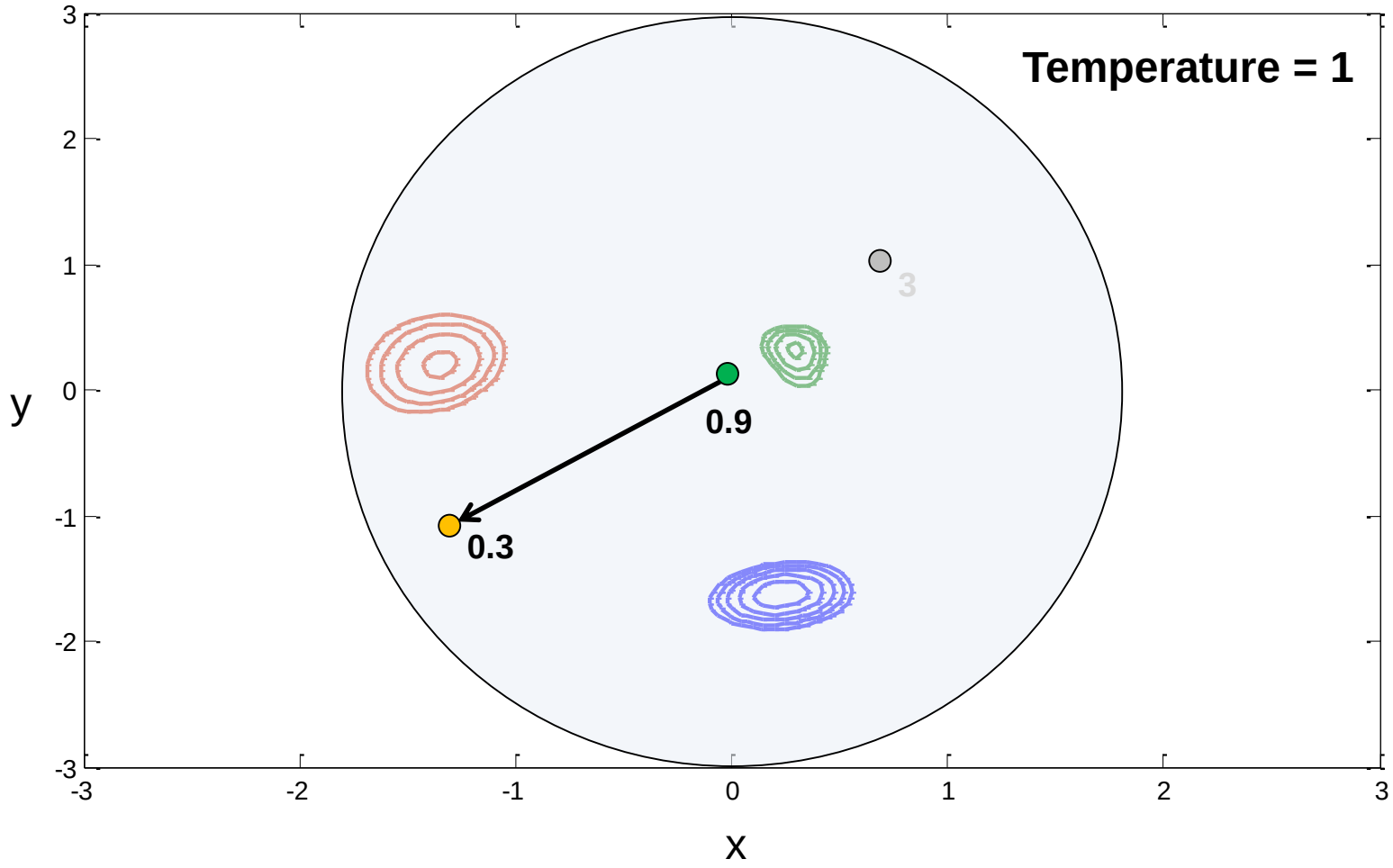
Simulated Annealing Overview – Iteration 1



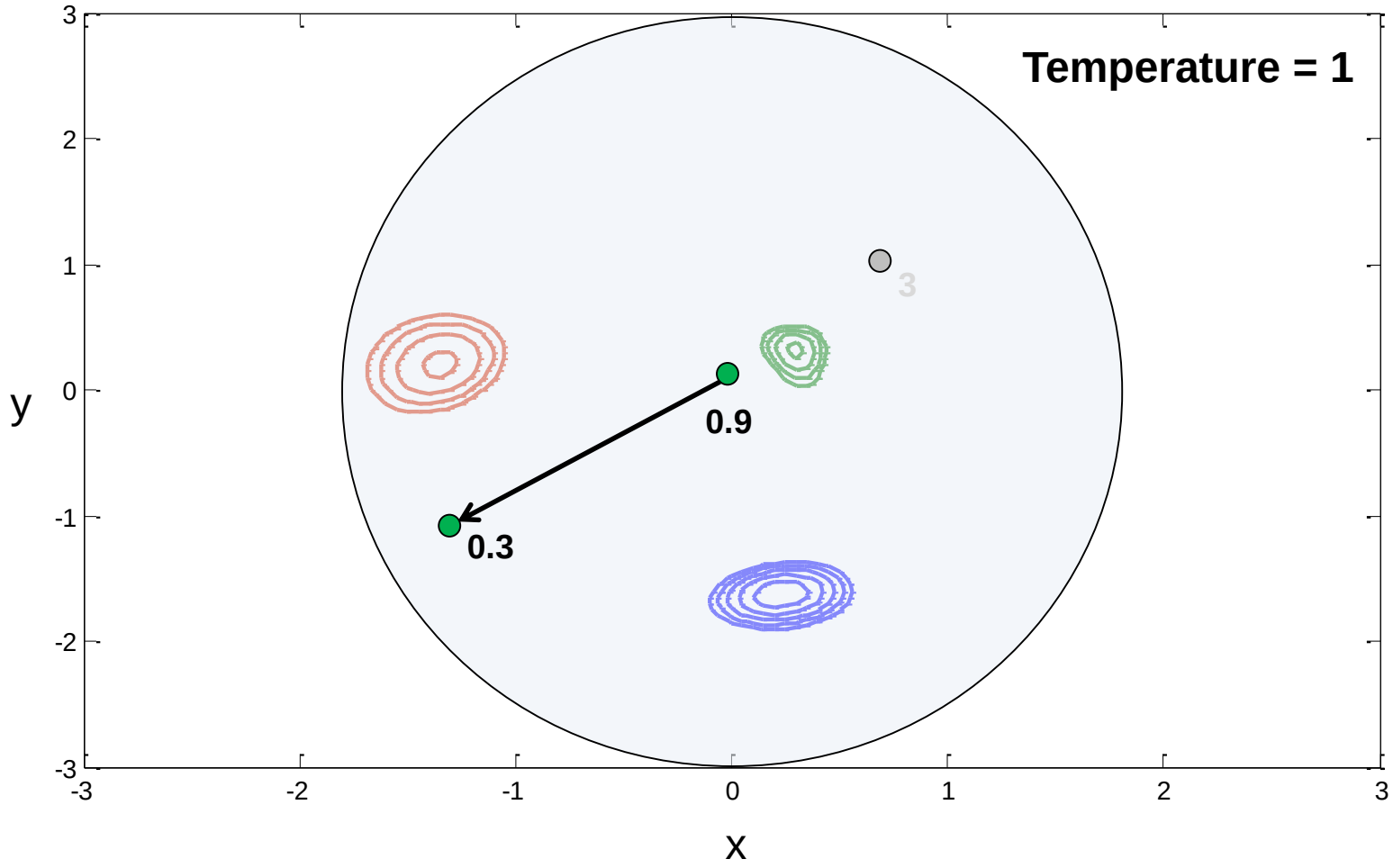
Simulated Annealing Overview – Iteration 1



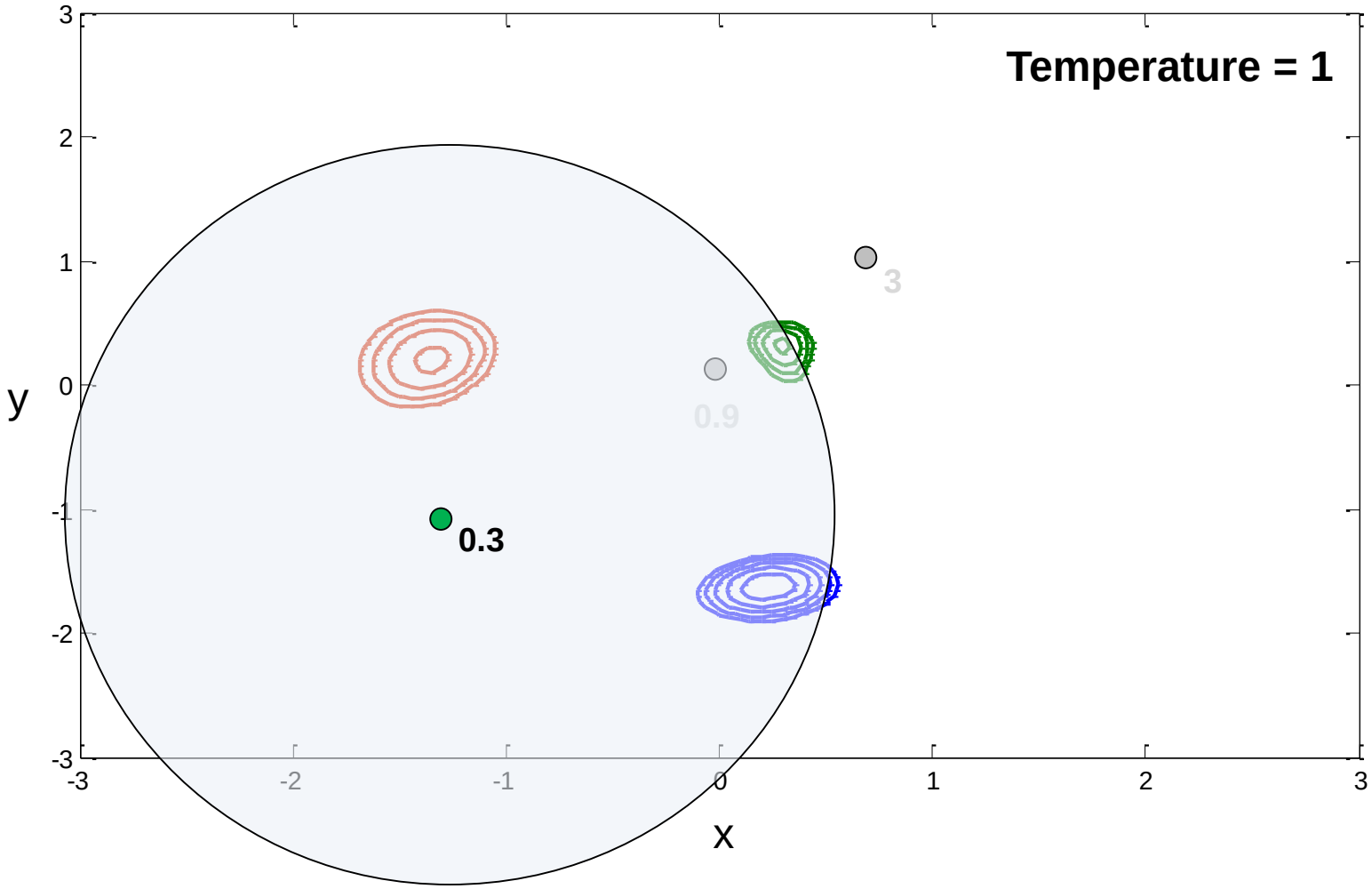
Simulated Annealing Overview – Iteration 1



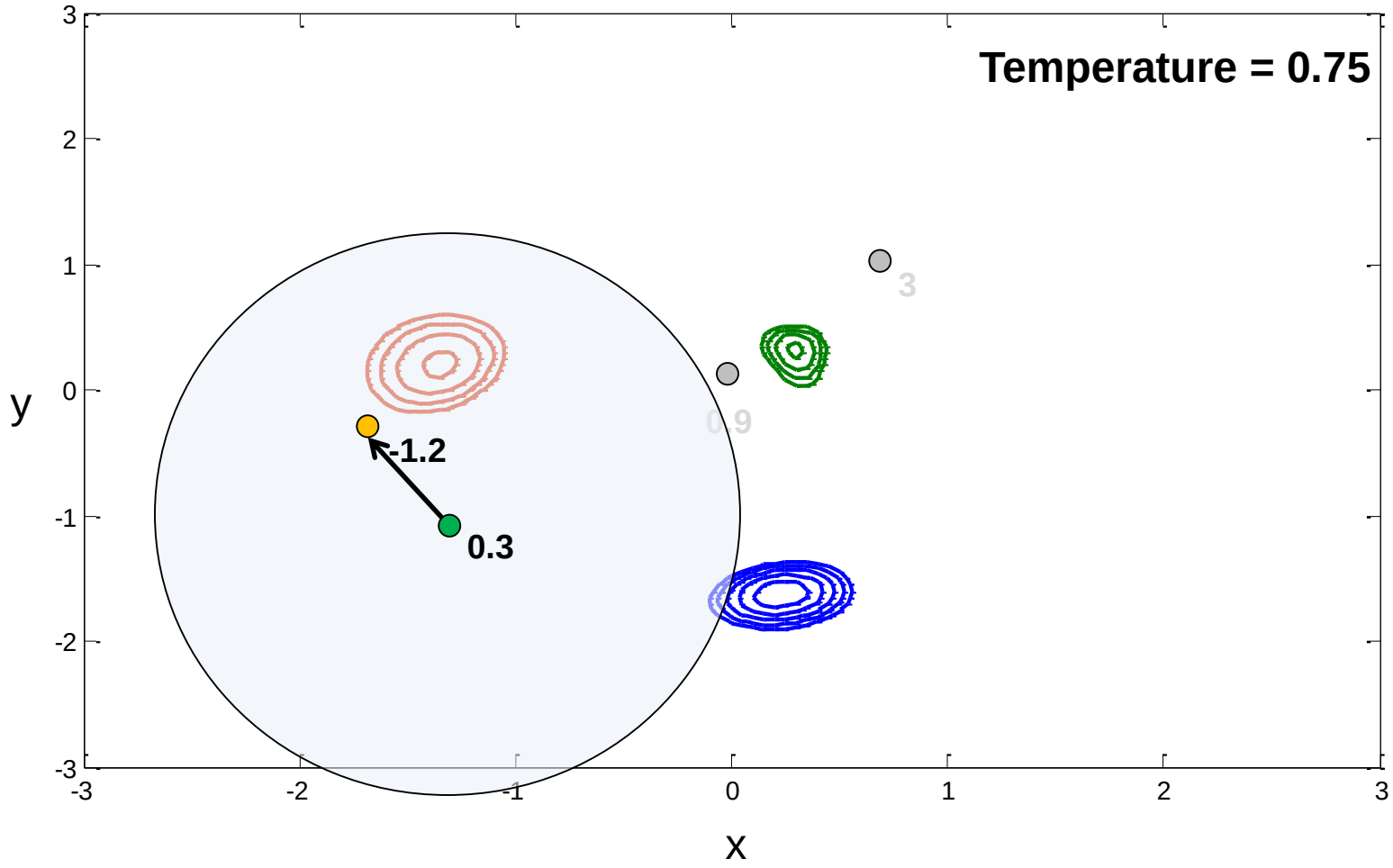
Simulated Annealing Overview – Iteration 1



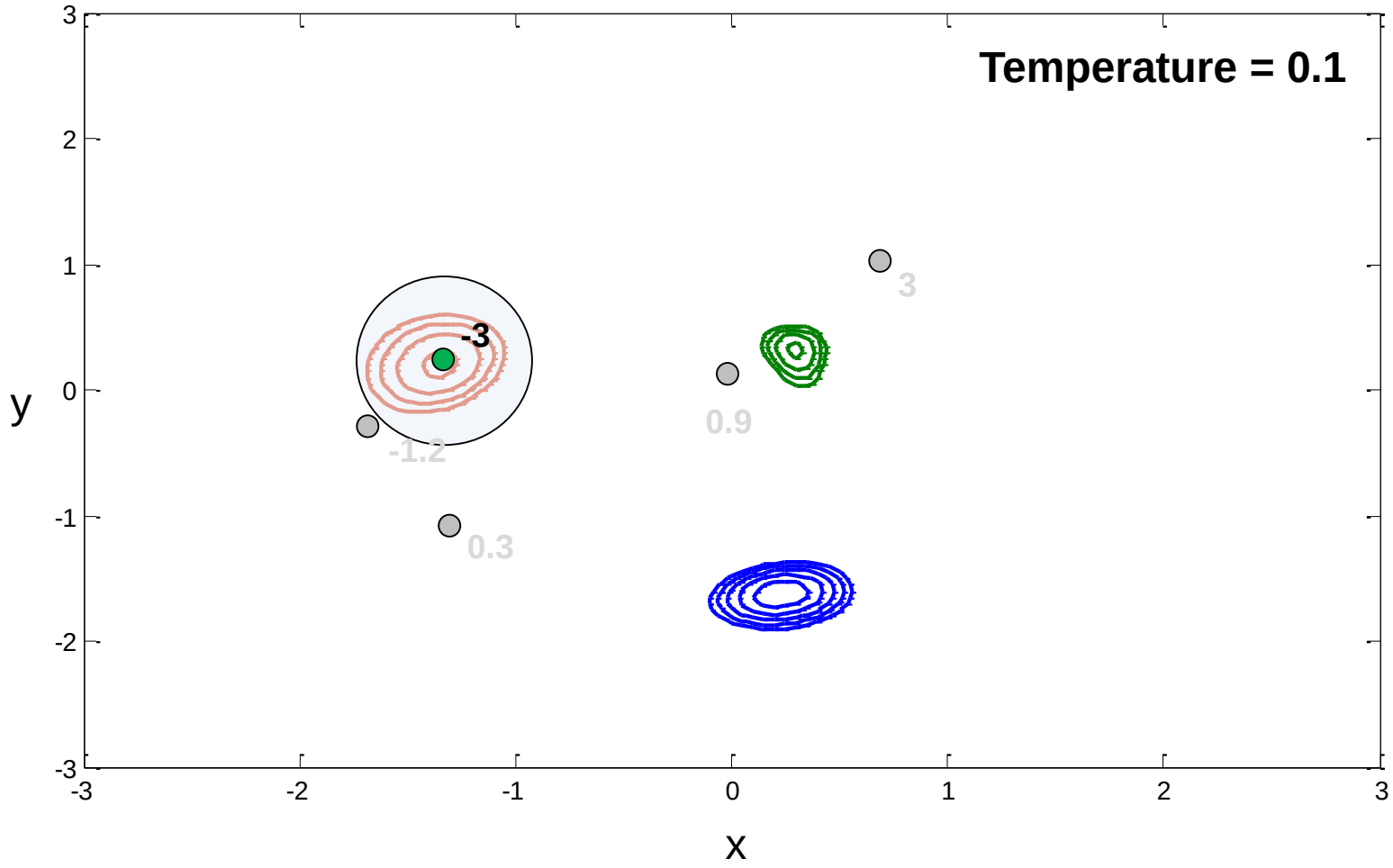
Simulated Annealing Overview – Iteration 2



Simulated Annealing Overview – Iteration 2

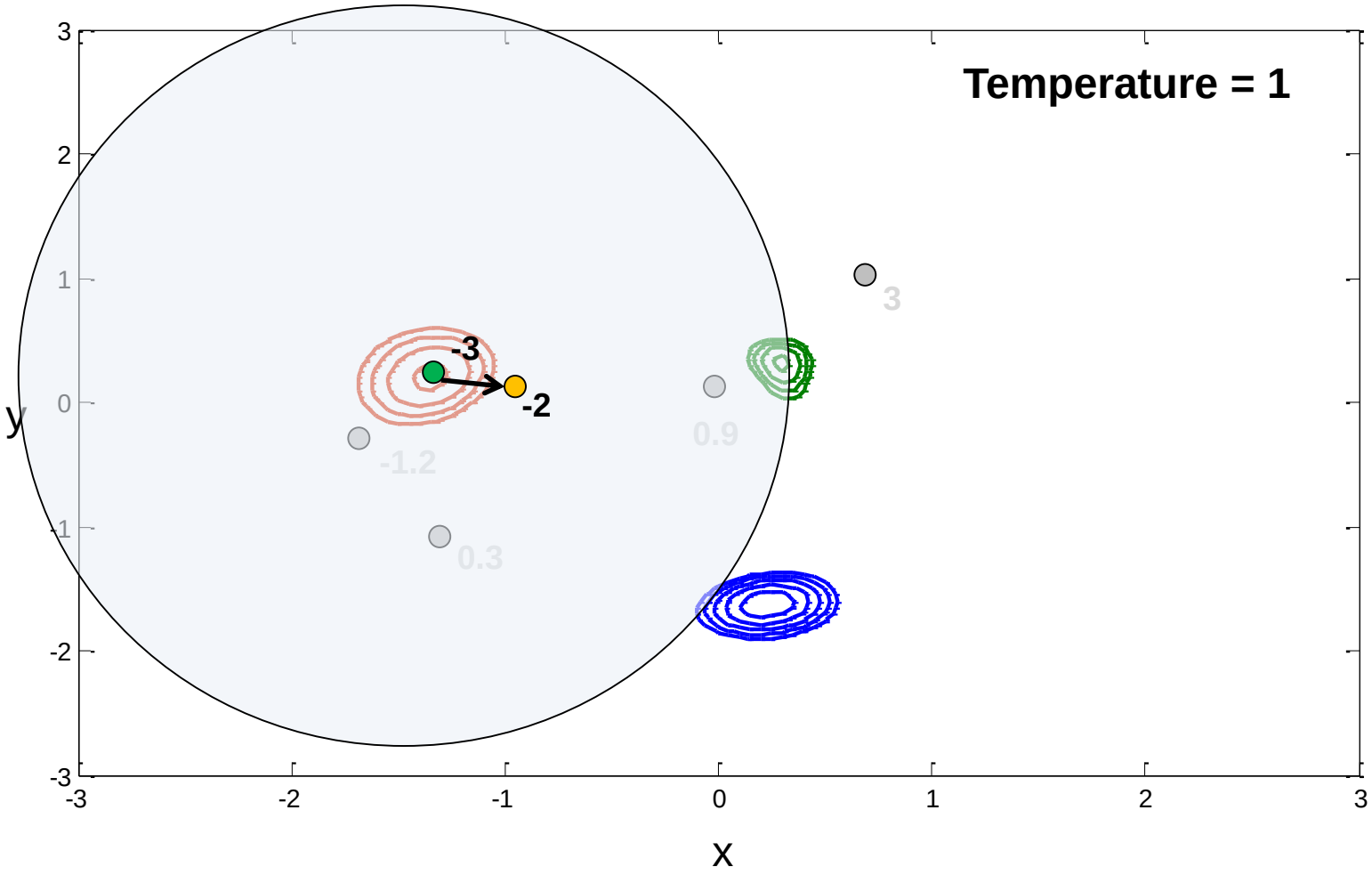


Simulated Annealing Overview – Iteration N-1



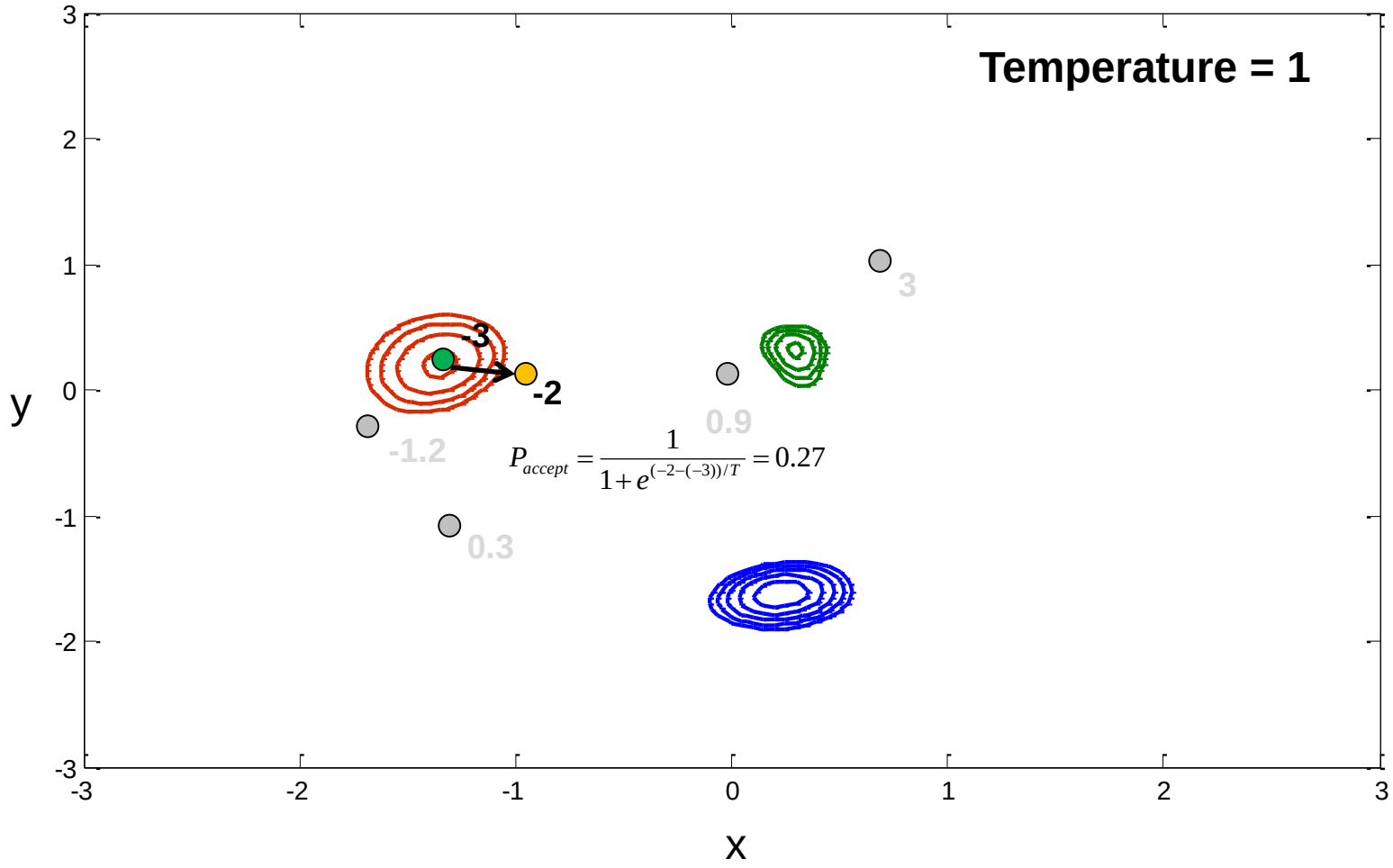
Simulated Annealing Overview – Iteration N

Reannealing



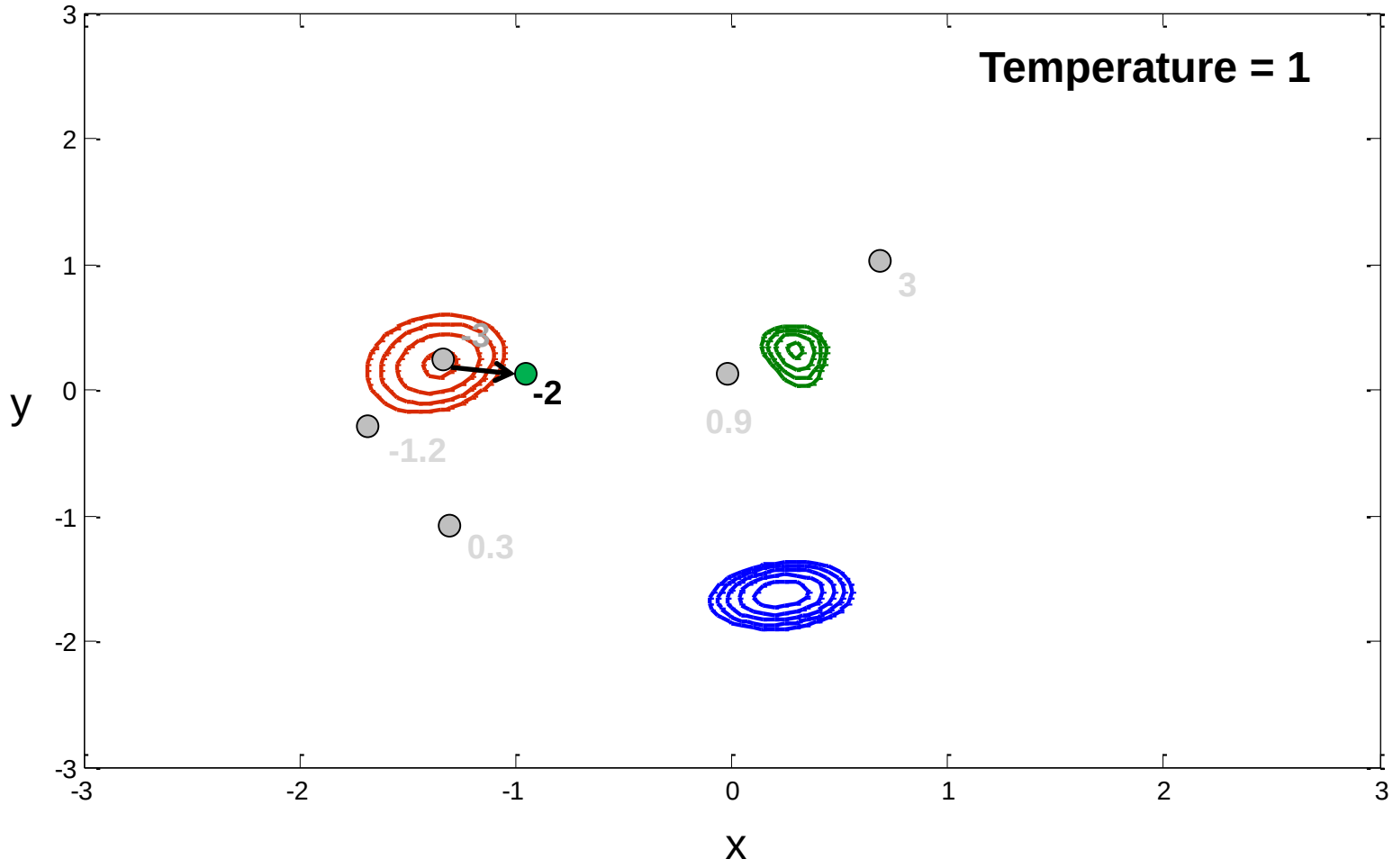
Simulated Annealing Overview – Iteration N

Reannealing

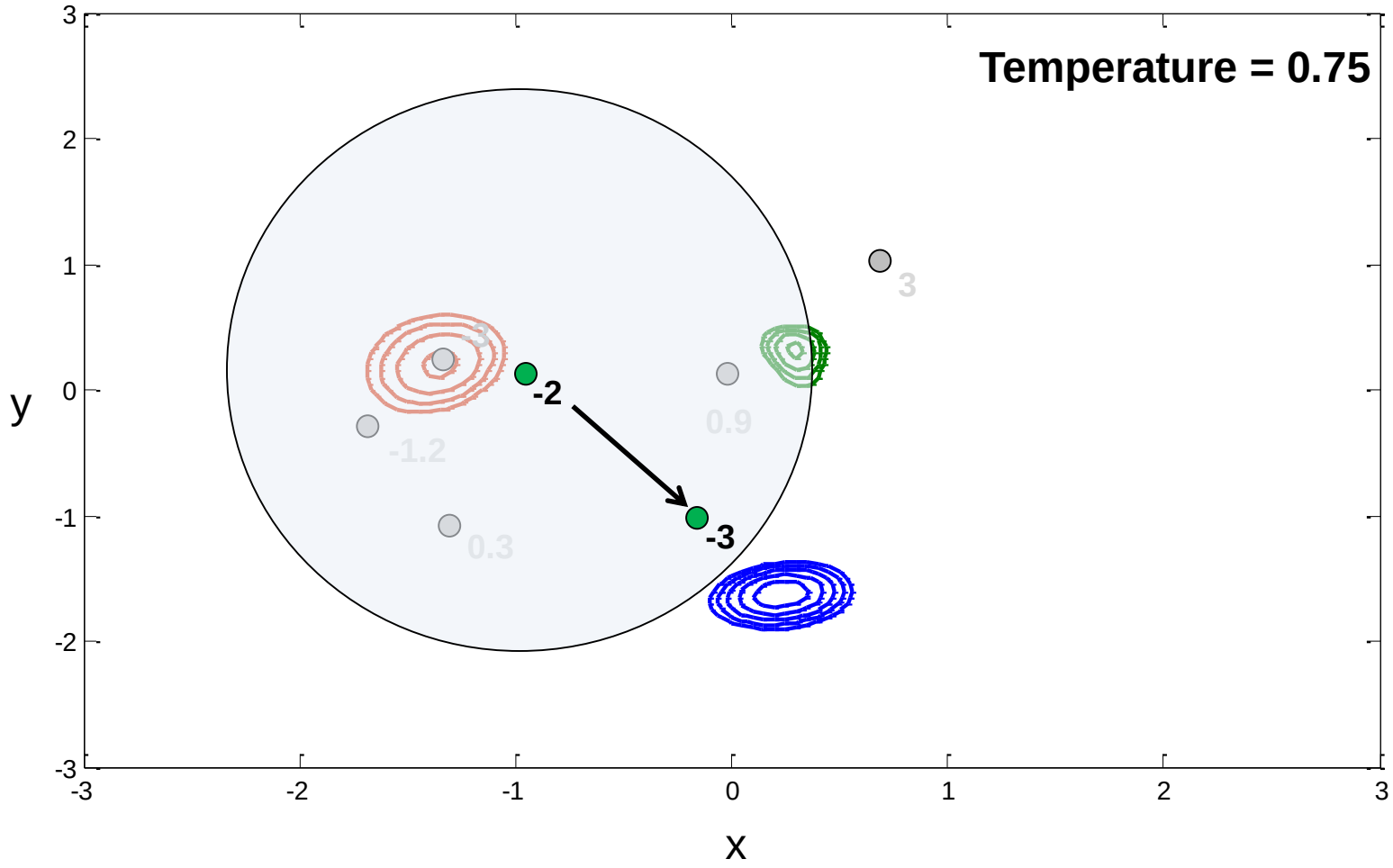


Simulated Annealing Overview – Iteration N

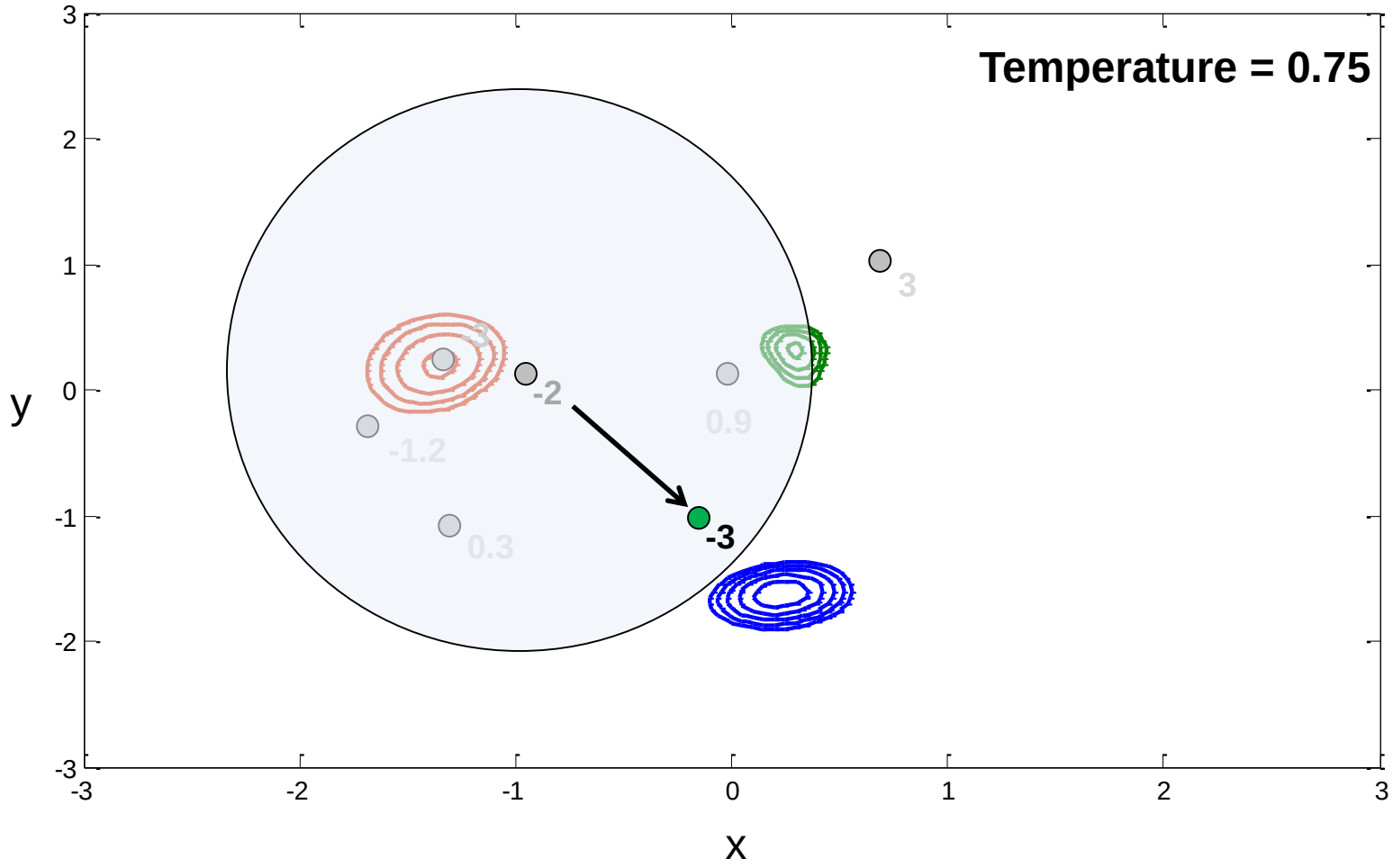
Reannealing



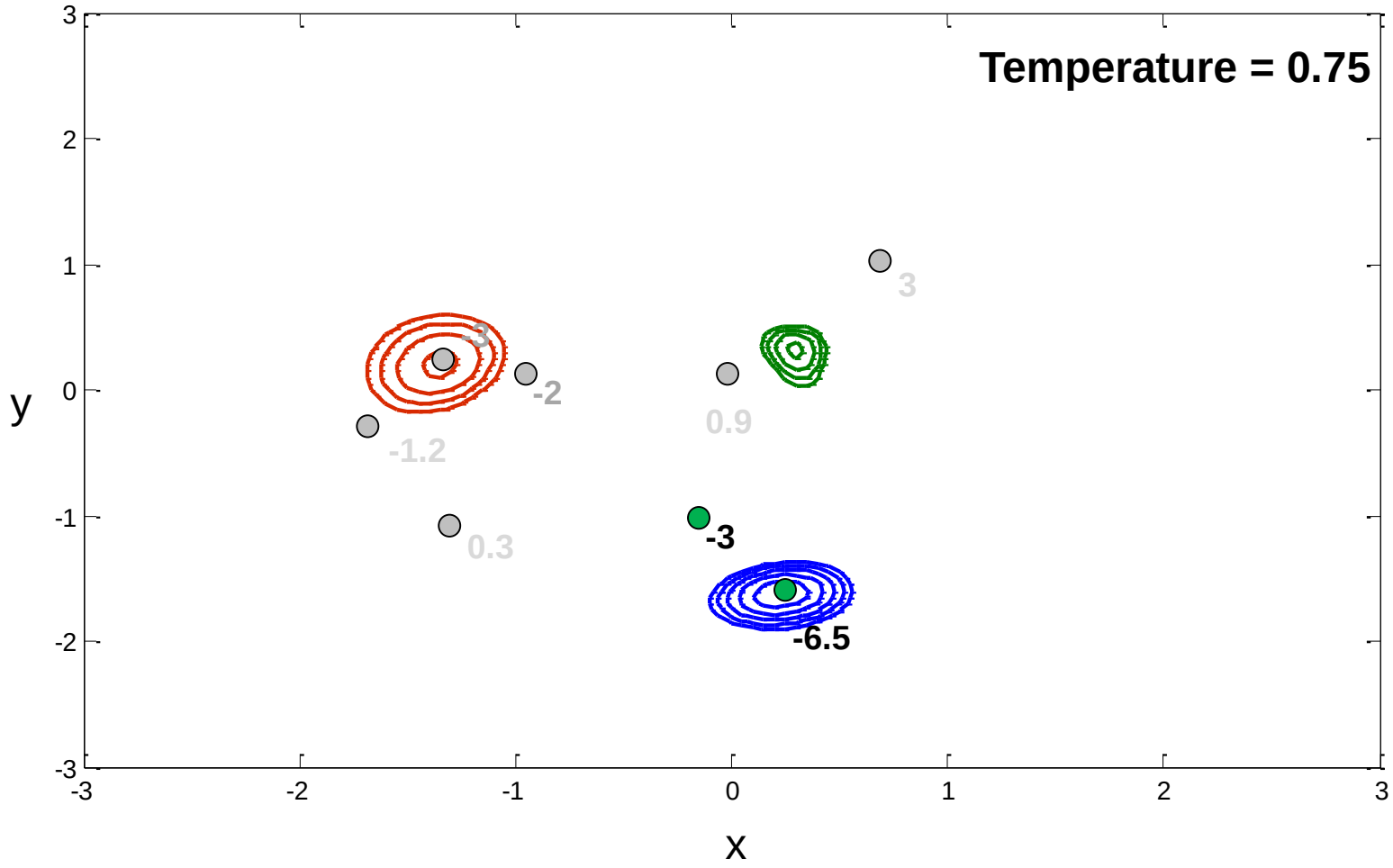
Simulated Annealing Overview – Iteration N+1



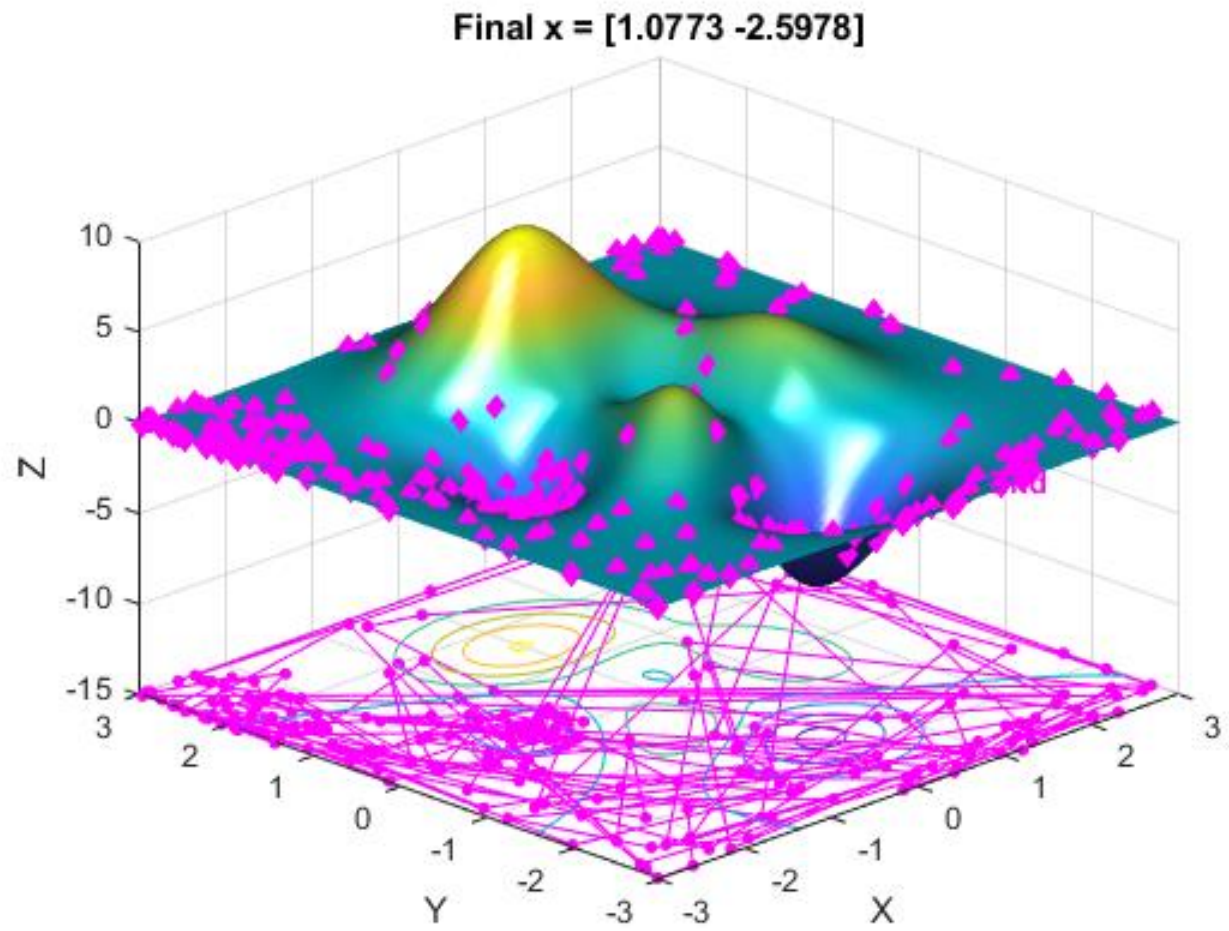
Simulated Annealing Overview – Iteration N+1



Simulated Annealing Overview – Iteration …



Simulated Annealing – Peaks Function



Global Optimization Toolbox Solvers

- `GlobalSearch`, `MultiStart`
  - Well suited for smooth objective and constraints
  - Relies on gradient calculations
  - Return the location of local and global minima

- `ga`, `simulannealbnd`, `particleswarm`
  - **Many function evaluations** to sample the search space
  - Work on both smooth and nonsmooth problems

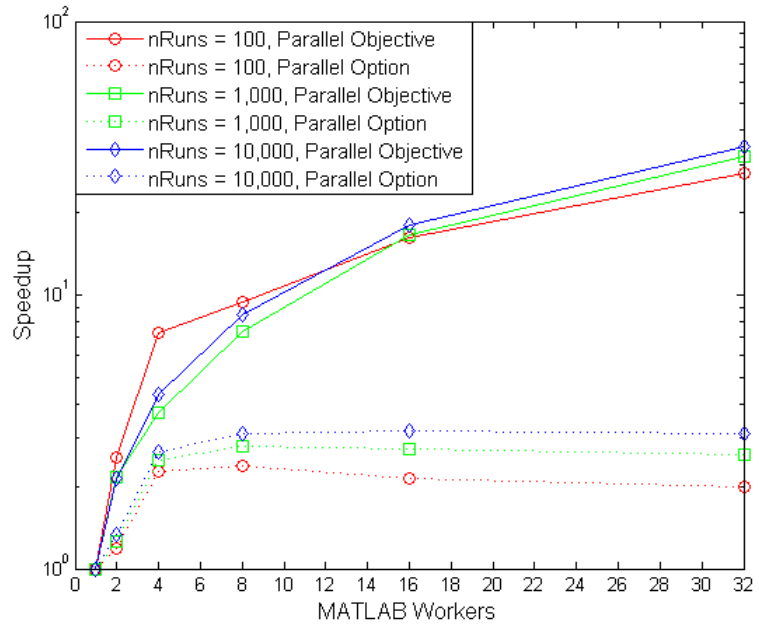
- `patternsearch`, `surrogateopt`
  - **Fewer function evaluations** than `ga`, `simulannealbnd`, `particlewarm`
  - Work on both smooth and nonsmooth problems

Optimization Toolbox Solvers

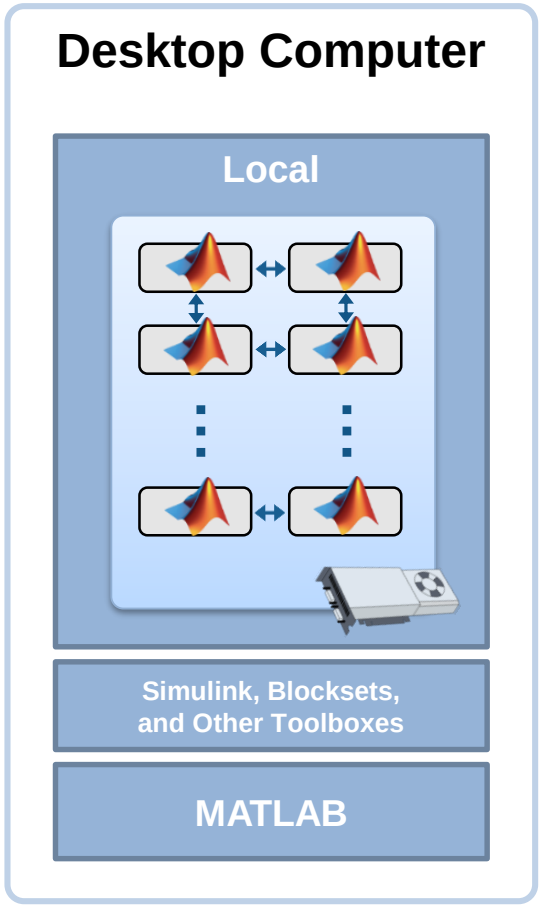
- `fmincon`, `fminbnd`, `fminunc`, `fgoalattain`, `fminimax`
  - Nonlinear constraints and objectives
  - Gradient-based methods for smooth objectives and constraints
- `quadprog`, `linprog`
  - Linear constraints and quadratic or linear objective, respectively
- `intlinprog`
  - Linear constraints and objective and integer variables
- `lsqlin`, `lsqnonneg`
  - Constrained linear least squares
- `lsqnonlin`, `lsqcurvefit`
  - Nonlinear least squares
- `fsolve`
  - Nonlinear equations

Speeding-up with Parallel Computing

- Global Optimization Toolbox
  - `patternsearch`, `surrogateopt`, `ga`, `gamultiobj`, `particleswarm`: Points evaluated in parallel at each iteration
  - `MultiStart`: Start points evaluated in parallel
- Optimization Toolbox
  - `fmincon`, `fminunc`, `fminimax`, `fgoalattain`, `fsolve`, `lsqcurvefit`, `lsqnonlin`: Parallel evaluation of objective function for finite differences
- Parallel Computing can also be used in the Objective Function
  - `parfor`

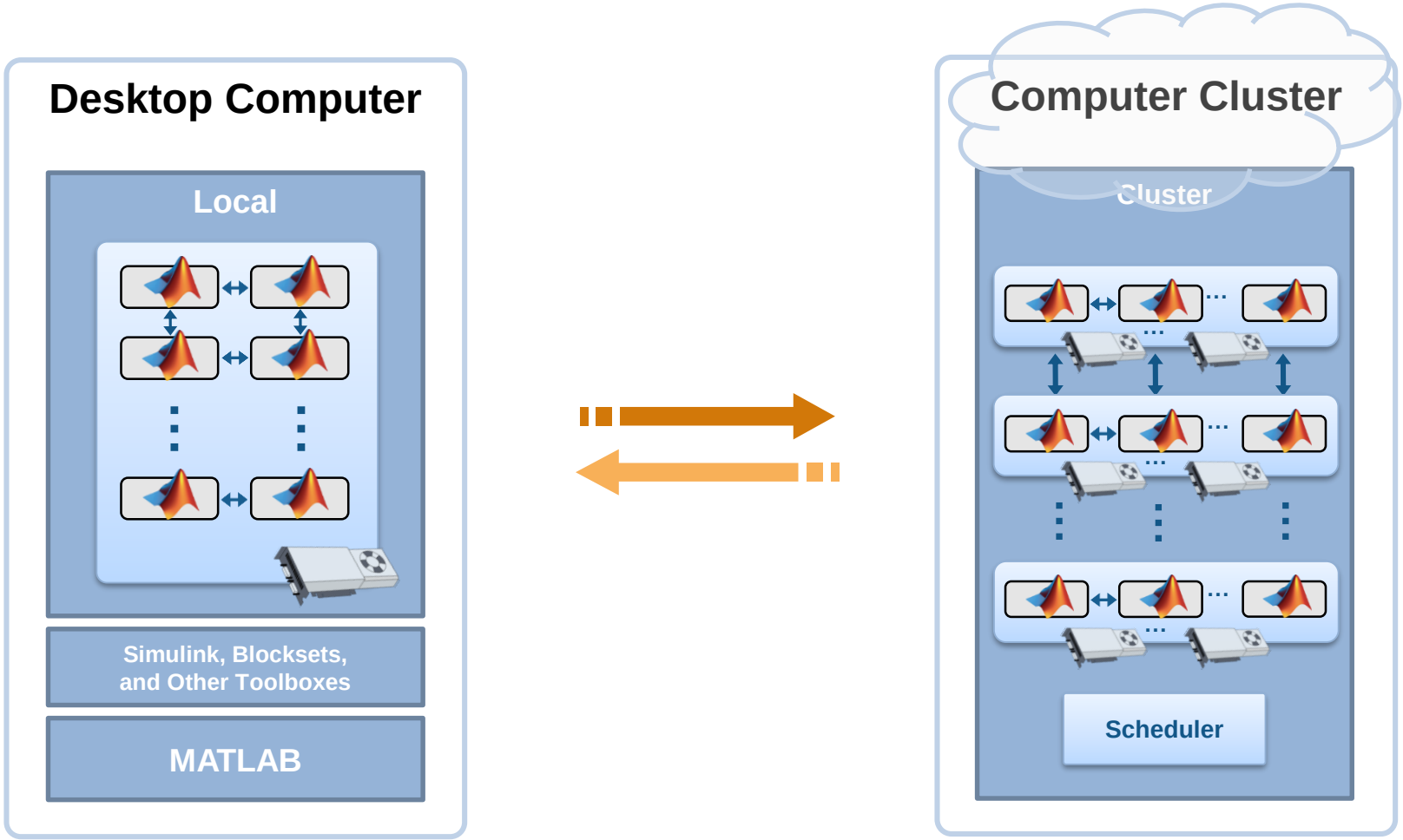


Parallel Computing Toolbox for the Desktop



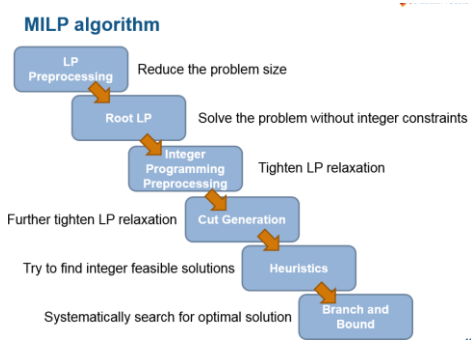
- Speed up parallel applications
- Take advantage of GPUs
- Prototype code for your cluster

Scale Up to Clusters and Clouds

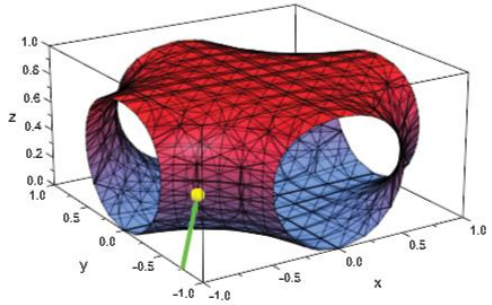


Learn More about Optimization with MATLAB

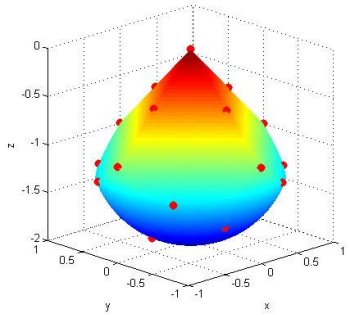
Recorded webinar: *Linear and Mixed Integer Linear Programming in MATLAB*



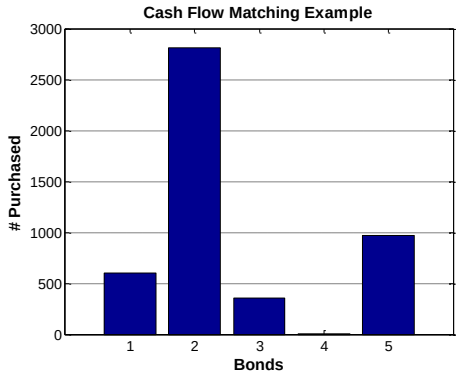
MATLAB Digest: *Using Symbolic Gradients for Optimization*



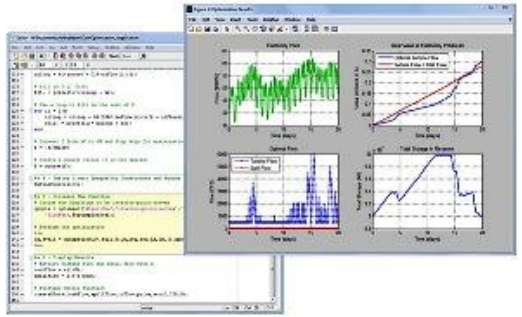
MATLAB Digest: *Improving Optimization Performance with Parallel Computing*



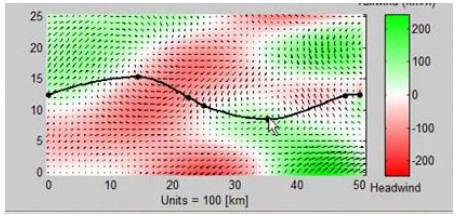
Recorded webinar: *Optimization in MATLAB for Financial Applications*



Recorded webinar: *Optimization in MATLAB: An Introduction to Quadratic Programming*



Optimization Toolbox Web demo: *Finding an Optimal Path using MATLAB and Optimization Toolbox*



Key Takeaways

- Solve a wide variety of optimization problems in MATLAB
  - Linear and Nonlinear
  - Continuous and mixed-integer
  - Smooth and Nonsmooth
- Find better solutions to multiple minima and non-smooth problems using global optimization
- Use symbolic math for setting up problems and automatically calculating gradients
- Using parallel computing to speed up optimization problems

MATLAB Central Community

Every month, over **2 million** MATLAB & Simulink users visit MATLAB Central to get questions answered, download code and improve programming skills.

MATLAB Central

Learn | Contribute | Connect

MATLAB Answers: Q&A forum; most questions get answered in only **60 minutes**

File Exchange: Download code from a huge repository of free code including **tens of thousands** of open source community files

Cody: Sharpen programming skills while having fun

Blogs: Get the inside view from Engineers who build and support MATLAB & Simulink

ThingSpeak: Explore IoT Data

And more for you to explore…

Training: *Optimization Techniques in MATLAB*

After this 1-day course you will be able to:

- Write objective function files and pass extra parameters
- Add different types of constraints
- Select an appropriate solver and algorithm
- Interpret the output from the solver and diagnose the progress of an optimization