Politecnico di Bari

Dipartimento di Ingegneria Elettrica e dell'Informazione

Corso di Laurea Magistrale in Ingegneria dei Sistemi Medicali

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

Bioinformatica Avanzata

Solving Numerical Optimization Problems with Genetic Algorithms (MATLAB and C)

Dr. Nicola **Altini**, Ph.D. Student

Prof. Eng. Vitoantonio **Bevilacqua**, Ph.D.

Anno Accademico 2019/2020

Optimization problem

- Optimization problem: find X^* so as to optimize $f(X^*)$
- $X^* = (x_1^*, x_2^*, \dots, x_n^*) \in \mathbb{R}^n$
- $X^* \in F \subseteq S$
- $S \subseteq \mathbb{R}^n$ is the search space
  - $l(i) \leq x_i \leq u(i)\ ,\ \ 1 \leq i \leq n$
- $F \subseteq S$ is the feasible search space, with $m \geq 0$ constraints:
  - $g_j(X^*) \leq 0, \quad for\ j = 1, \dots, q$
  - $h_j(X^*) = 0, \quad for\ j = q+1, \dots, m$
- Type of constraints:
  - **LE**: Linear Equations
  - **LI**: Linear Inequalities
  - **NE**: Nonlinear Equations
  - **NI**: Nonlinear Inequalities

MATLAB Global Optimization Toolbox

Global Optimization Toolbox provides functions that search for global solutions to problems that contain multiple maxima or minima.

- Direct Search
  - Pattern search solver for derivative-free optimization, constrained or unconstrained
- **Genetic Algorithm**
  - **Genetic algorithm solver for mixed-integer or continuous-variable optimization, constrained or unconstrained**
- Particle Swarm
  - Particle swarm solver for derivative-free unconstrained optimization or optimization with bounds
- Surrogate Optimization
  - Surrogate optimization solver for expensive objective functions, with bounds and optional integer constraints
- Simulated Annealing
  - Simulated annealing solver for derivative-free unconstrained optimization or optimization with bounds
- Multiobjective Optimization
  - Pareto sets via genetic or pattern search algorithms, with or without constraints

Genetic Algorithms in MATLAB

- Function: `ga()`
- Parameters:
  - `fun`
    - The $f(X)$ to optimize
  - `nvars`
    - $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$
  - `A, b`
    - Linear inequalities, in matricial form
  - `Aeq, beq`
    - Linear equalities, in matricial form
  - `lb, ub`
    - Lower and upper bounds
  - `nonlcon`
    - Nonlinear equalities and inequalities
  - `IntCon`
    - Integer constraints
  - `options`
    - Hyperparameters of the algorithm

The fitness function

- `fun` – Objective function
- Objective function, specified as a function handle or function name. Write the objective function to accept a row vector of length `nvars` and return a scalar value.
- MATLAB attempts to find X^* as to minimize $f(X^*)$
- **Exercise 1**: implement a MATLAB function which realizes $f(X)$:

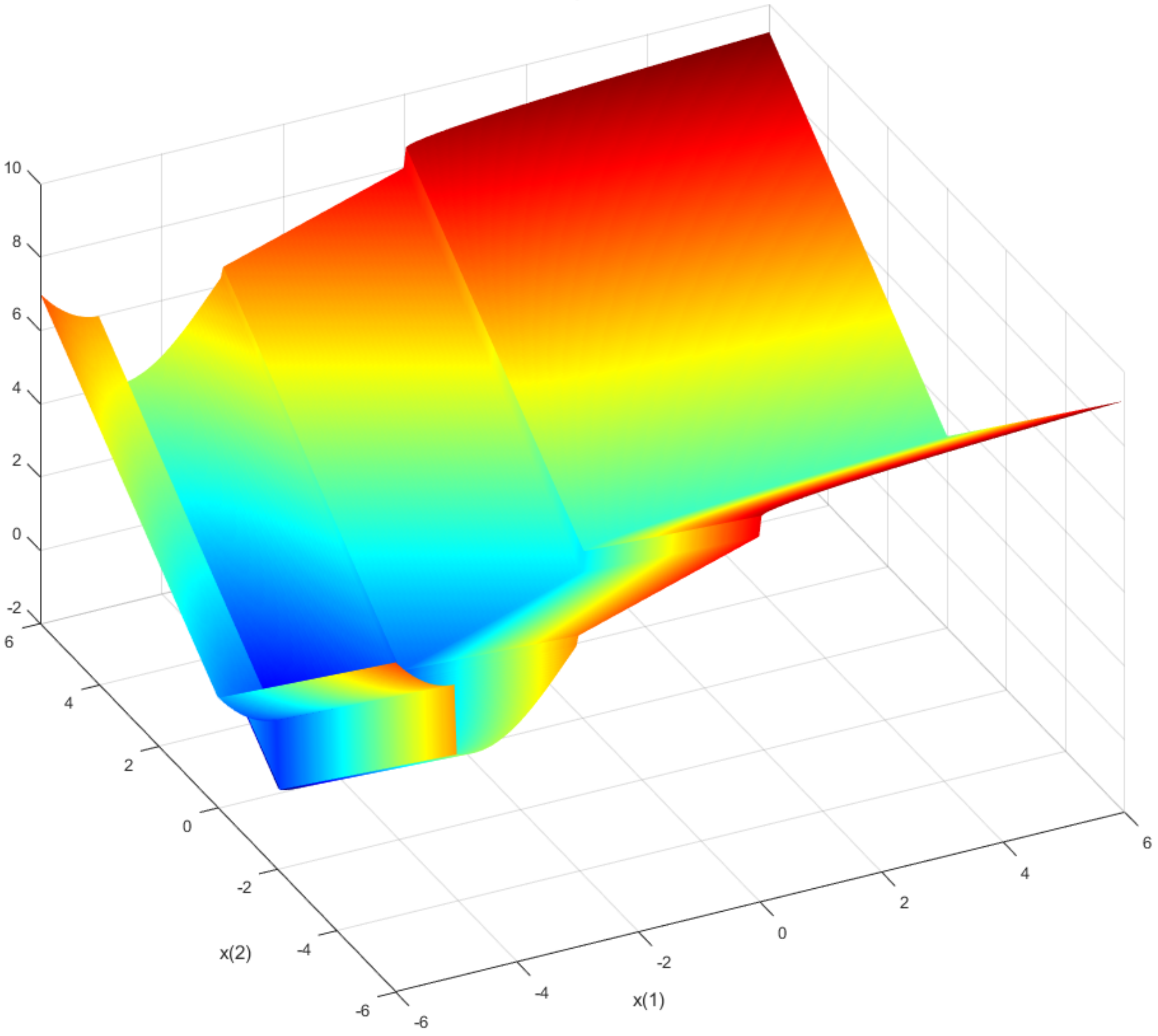
$$f(X) = f(x_1, x_2) = \begin{cases} (x_1 + 5)^2 + |x_2| & if\ x_1 \leq -5 \\ -2\sin(x_1) + |x_2| & if -5 \leq x_1 \leq -3 \\ \frac{1}{2}x_1 + 2 + |x_2| & if -3 \leq x_1 \leq 0 \\ \frac{3}{10}\sqrt{x_1} + \frac{5}{2} + |x_2| & if\ x_1 > 0 \end{cases}$$

The fitness function

- **Exercise 1 solution**:

```
function f = ps_example(x)
for i = 1:size(x,1)
    if  x(i,1) < -5
        f(i) = (x(i,1)+5)^2 + abs(x(i,2));
    elseif x(i,1) < -3
        f(i) = -2*sin(x(i,1)) + abs(x(i,2));
    elseif x(i,1) < 0
        f(i) = 0.5*x(i,1) + 2 + abs(x(i,2));
    elseif x(i,1) >= 0
        f(i) = .3*sqrt(x(i,1)) + 5/2 +abs(x(i,2));
    end
end
```

ps_example(x)



Optimization

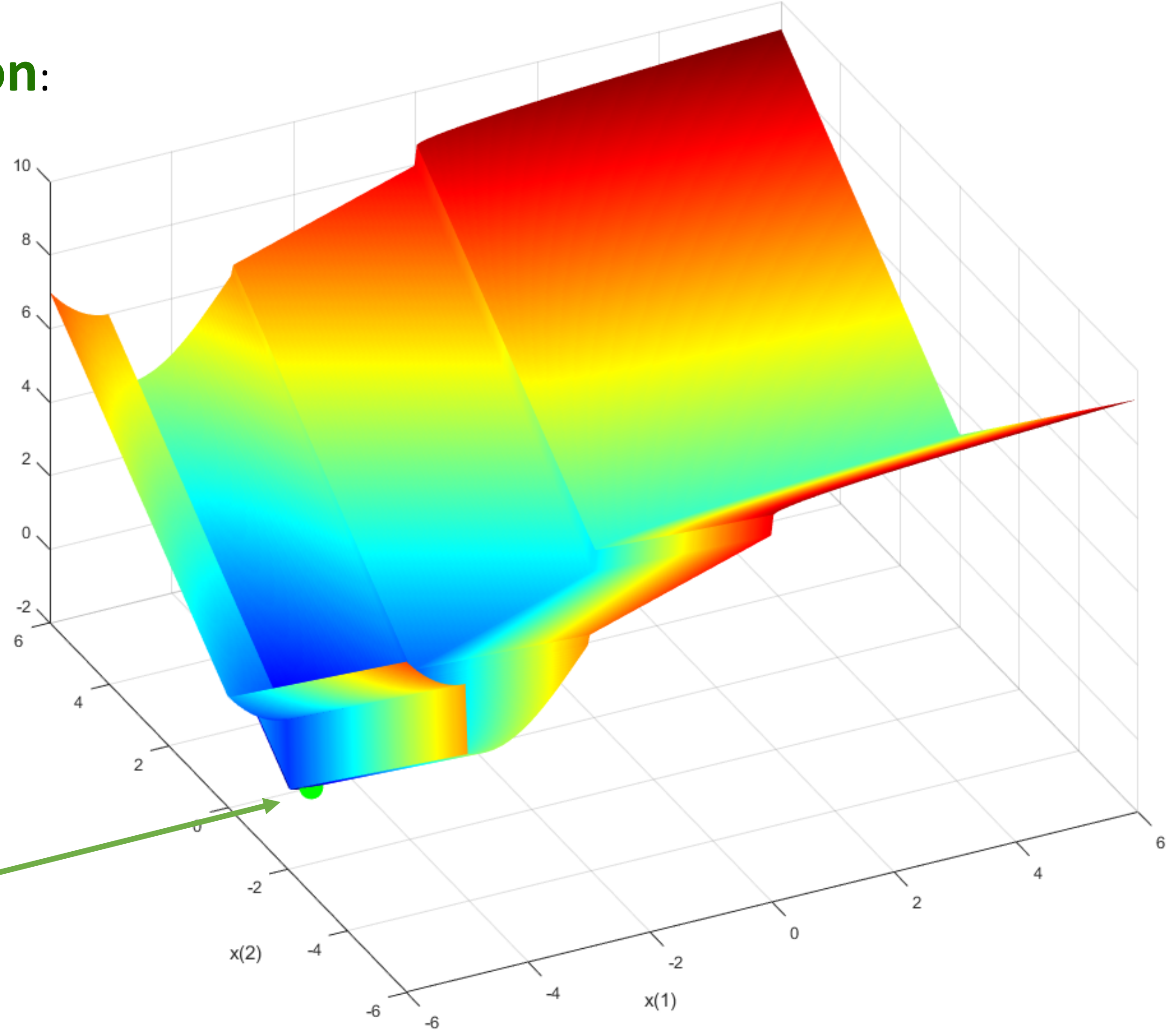
- `x = ga(@ps_example,2)`
  - `@ps_example` is the handle of the fitness function
  - `2` is the number of variables in the fitness function
- x is the optimum point found by genetic algorithms
- $ps\_example(x)$ is the value of the fitness function in the optimum
- **Exercise 2**: use your MATLAB to find x and $ps\_example(x)$. Then make a 3D plot of the optimum point.

Optimum point

Exercise 2 solution:

x = [-4.6793, -0.0860]
f(x) = -1.9129

Minimum
found by GA



Linear inequalities

- Linear inequalities:

$$\begin{cases} -x_1 - x_2 \le -1 \\ -x_1 + x_2 \le 5 \end{cases}$$

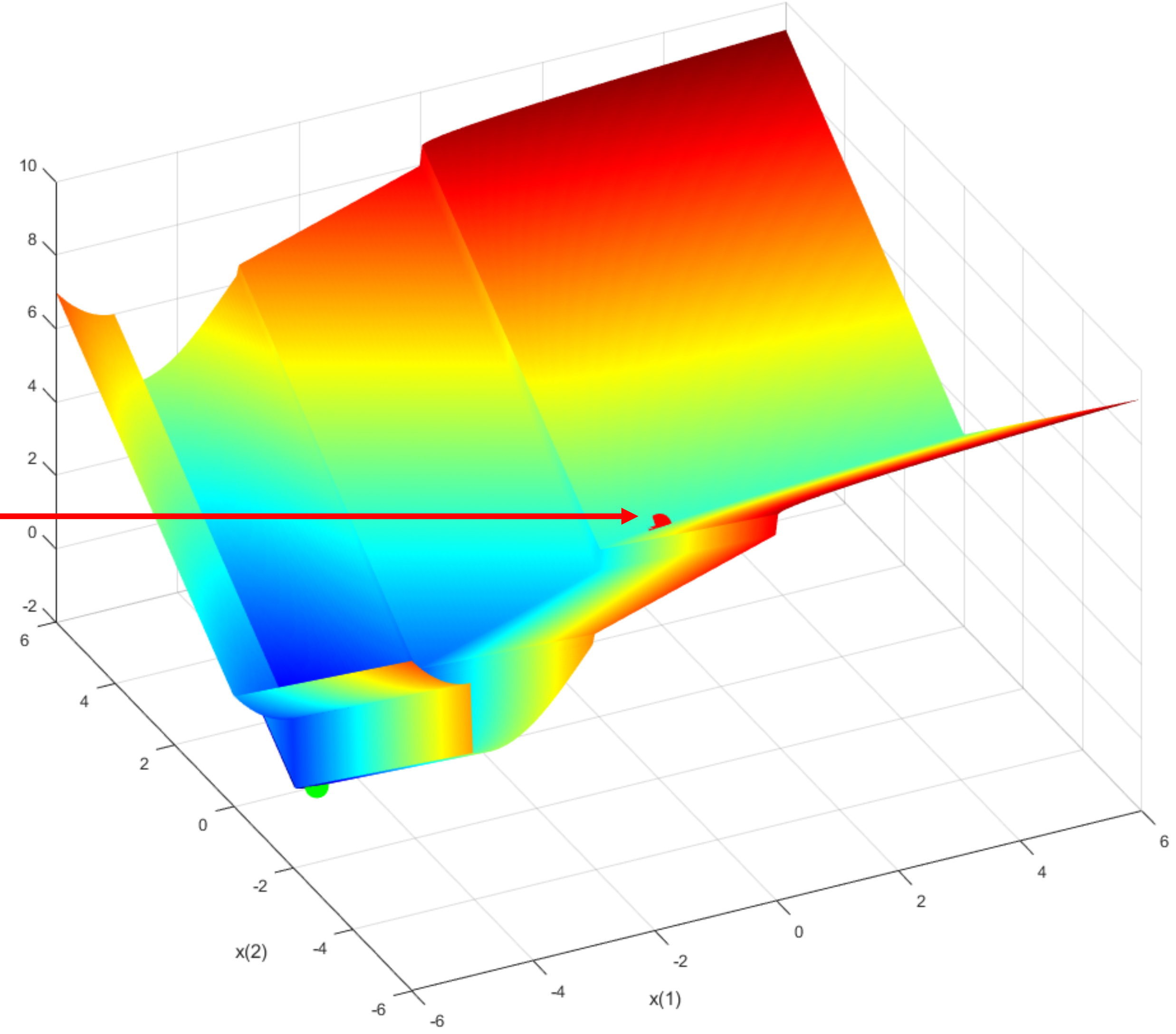
- MATLAB matricial form:

```
A = [-1, -1;
     -1,  1]
b = [-1;  5]
```

- Optimization:
  - `x = ga(@ps_example, 2, A, b)`

Optimum with LI

Minimum found by GA with LI



x = [0.9992, -0.0000]
f(x) = 2.7999

Linear equalities

- Linear equalities and inequalities:

$$\begin{cases} -x_1 - x_2 \leq -1 \\ -x_1 + x_2 = 5 \end{cases}$$

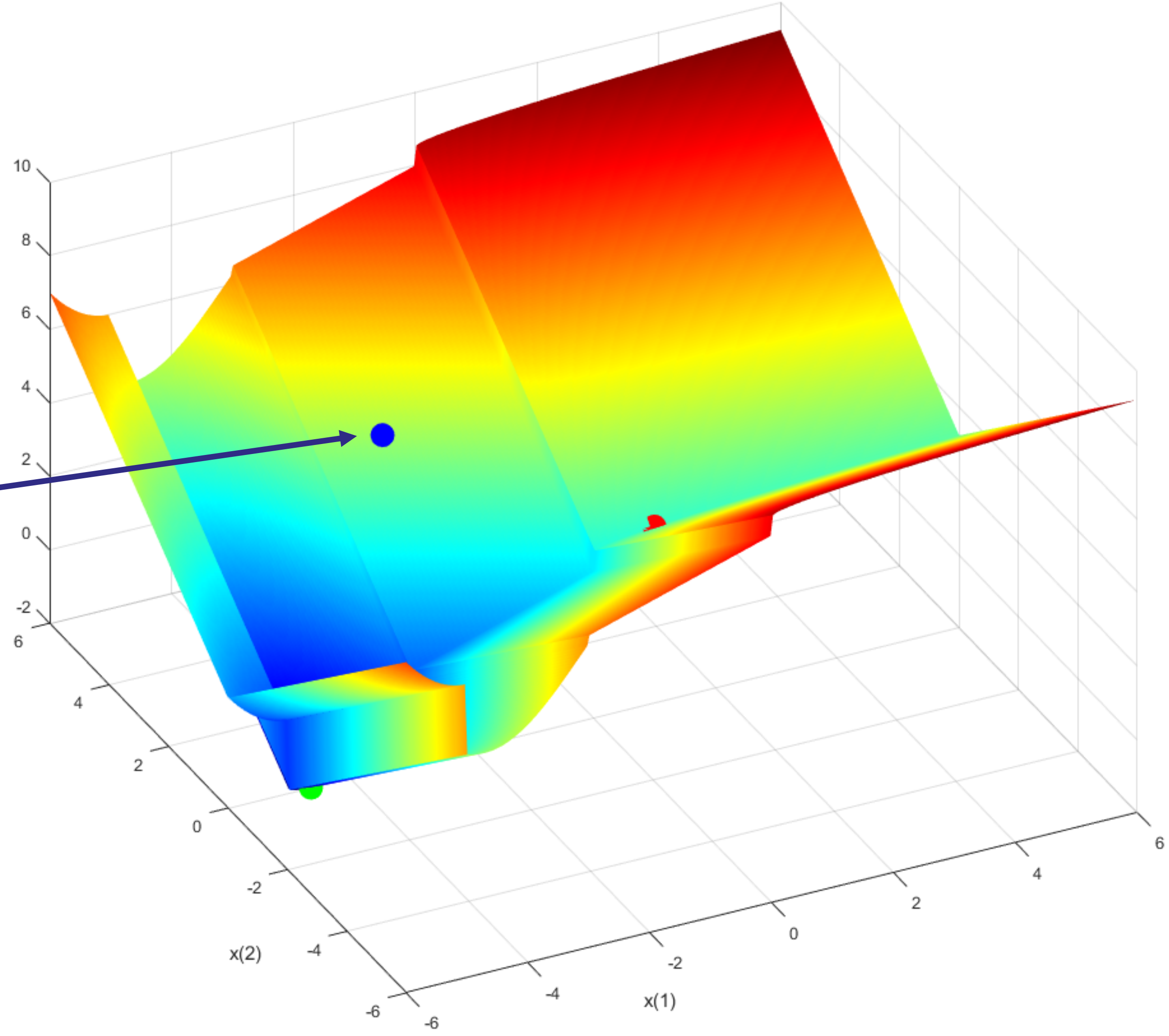
- MATLAB matricial form:

```
A = [-1 -1]
b = -1
Aeq = [-1 1]
beq = 5
```

- Optimization:
  - `x = ga(@ps_example, 2, A, b, Aeq, beq)`

Optimum with LI and LE

Minimum found by GA with LI and LE



x = [-2.0000, 2.9990]
f(x) = 3.9990

LI and LE

- **Exercise 3**: implement in MATLAB the following conditions:

$$
\begin{cases}
-2x_1 \le -1 + 4x_2 \\
+x_2 = 5 - 2x_1 \\
x_2 \ge 2 + 3x_1 \\
x_1 + x_2 \ge 0
\end{cases}
$$

```
A = ?
b = ?
Aeq = ?
beq = ?
```

LI and LE

- **Exercise 3 solution**:
- Remember to put
  - LI in the form $Ax \leq b$
  - LE in the form $A_{eq}x = b_{eq}$

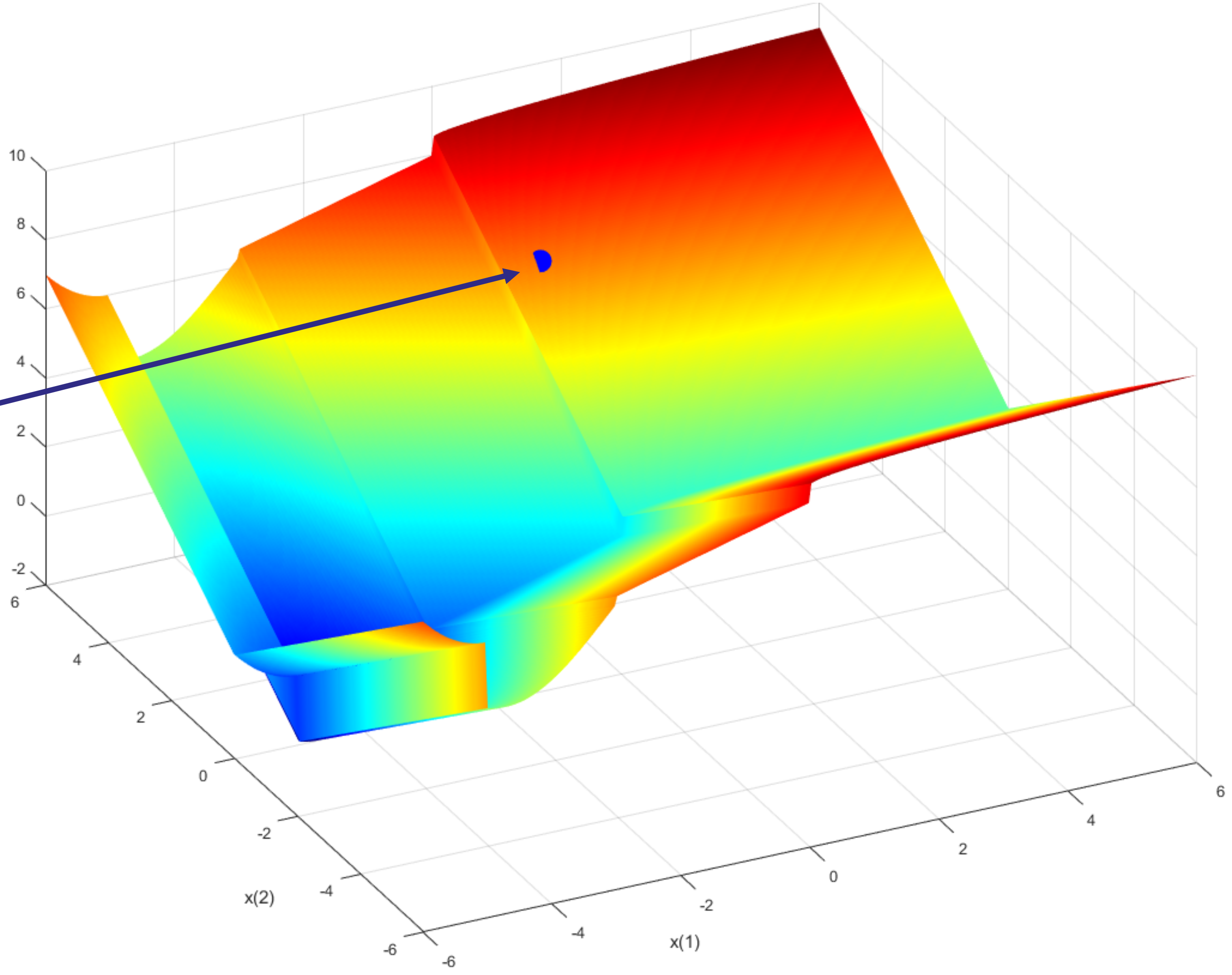
$$\begin{cases} -2x_1 - 4x_2 \leq -1 \\ +3x_1 - x_2 \leq -2 \\ -x_1 - x_2 \leq 0 \\ 2x_1 + x_2 = 5 \end{cases}$$

```
A = [-2 -4;
      3 -1;
     -1 -1]
b = [-1; -2; 0]
Aeq = [2 1]
beq = [5]
```

LI and LE

Exercise 3 solution:

Minimum found by GA with LI and LE



x = [0.6000, 3.7990]
f(x) = 6.5314

Lower and upper bounds

- Lower and upper bounds:

$$\begin{cases} 1 \leq -x_1 \leq 6 \\ -3 \leq \quad x_2 \leq 8 \end{cases}$$

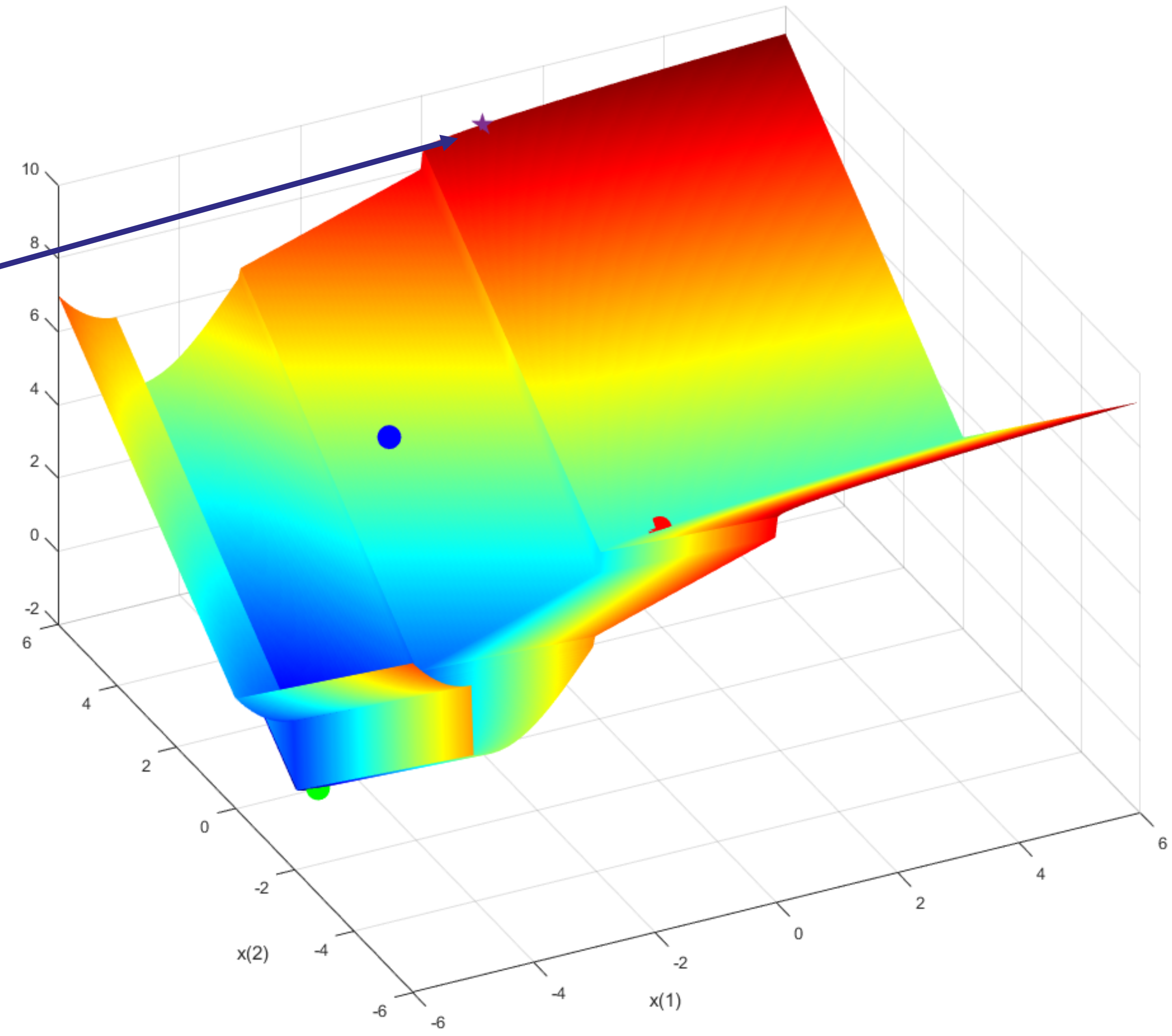
- MATLAB form:

```
lb = [1 -3]
ub = [6  8]
```

- Optimization:
  - `x = ga(@ps_example, 2, A, b, Aeq, beq, lb, ub)`

Optimization with LI, LE and bounds

Minimum found by GA with LI, LE and bounds



x = [1.0000, 5.9991]
f(x) = 8.7991

Nonlinear constraints

- Nonlinear constraints:

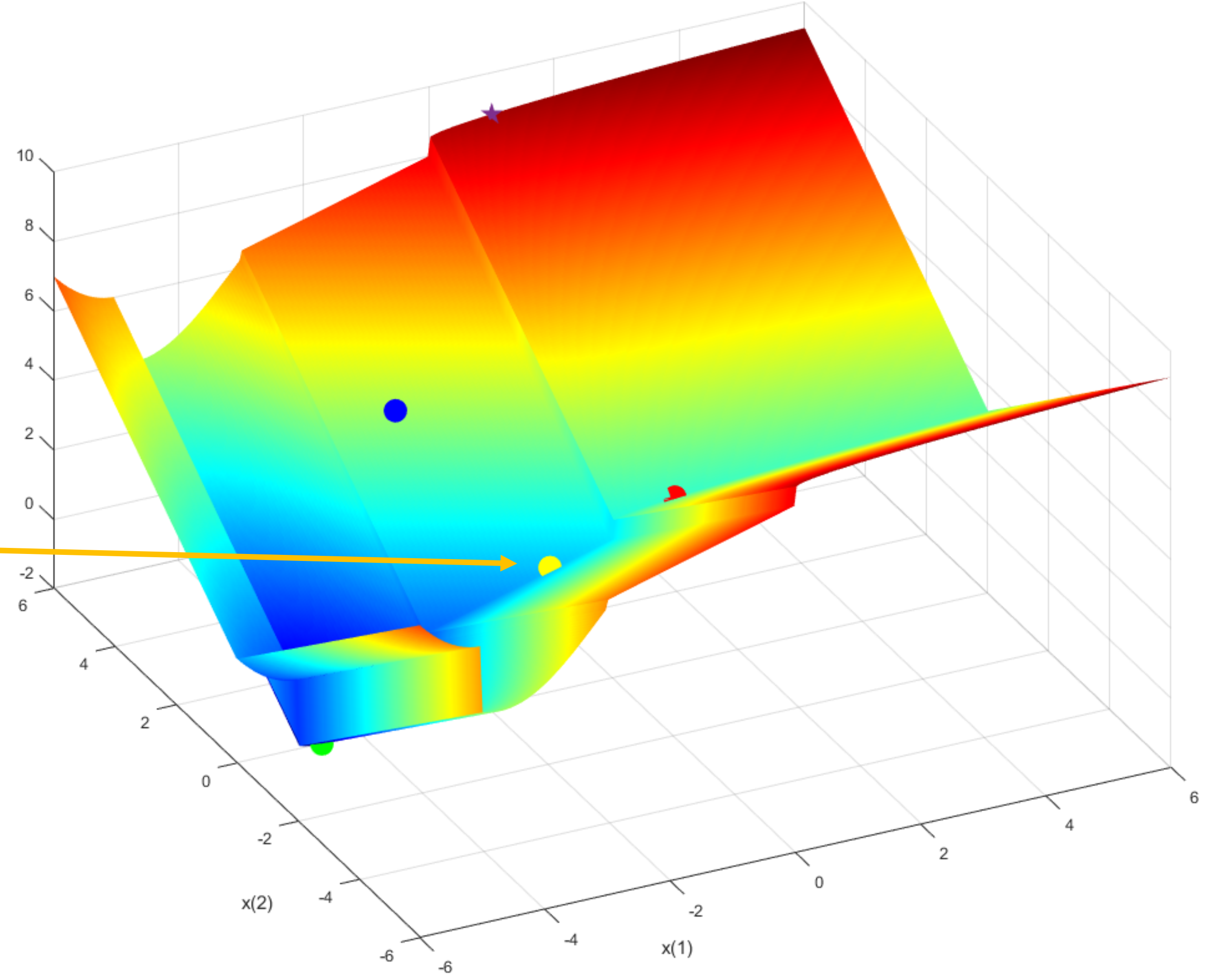
$$\begin{cases} 2x_1^2 + x_2^2 \leq 3 \\ (x_1+1)^2 = \left(\frac{x_2}{2}\right)^4 \end{cases}$$

- Define a function

```
function [c, ceq] = ellipsecons(x)
c = 2*x(1)^2 + x(2)^2 - 3;
ceq = (x(1)+1)^2 - (x(2)/2)^4;
end
```

- Optimization
  - `x = ga(@ps_example,2,[],[],[],[],[],[],@ellipsecons)`

Optimization with NI, NE



Minimum found by GA with NI, NE

x = [-0.9766, 0.0362]
f(x) = 1.5479

NI, NE

- **Exercise 4**: implement in MATLAB a function with 2 nonlinear equalities and 2 nonlinear inequalities

$$\begin{cases} g_1(X^*) \leq 0 \\ g_2(X^*) \leq 0 \\ h_1(X^*) = 0 \\ h_2(X^*) = 0 \end{cases}$$

```
function [c, ceq] = myConstraints(x)
c = ?
ceq = ?
end
```

Exercise 5

- **Exercise 5**: solve the following optimization problem:
- Minimize the squared norm of a point in $\mathbb{R}^2$, with the constraints that:
  - The point must be inside the rectangle with vertices (2,2), (2,8), (7,2), (7,8)
  - The point must belong to the circonference with center in (8,6) and radius 3

Exercise 5

- **Exercise 5**: solve the following optimization problem:
- Minimize the squared norm of a point in $\mathbb{R}^2$, with the constraints that:
  - The point must be inside the rectangle with vertices (2,2), (2,8), (7,2), (7,8)
  - The point must belong to the circonference with center in (8,6) and radius 3

$$\min f(X) = \min f(x_1, x_2) = \min x_1^2 + x_2^2$$

$$\begin{cases} 2 \leq x_1 \leq 7 \\ 2 \leq x_2 \leq 8 \\ (x_1 - 8)^2 + (x_2 - 6)^2 = 3^2 \end{cases}$$

Exercise 5 - MATLAB solution

- The fitness function is the squared norm:

```
function f = squarednorm(x)
f = x(1)^2 + x(2)^2;
end
```

- The following is a nonlinear equality constraint (NE):

$$(x_1 - 8)^2 + (x_2 - 6)^2 = 3^2$$

```
function [c,ceq] = circlecons(x)
c = [];
ceq = (x(1)-8)^2 +(x(2)-6)^2 -9;
end
```

  - Note that we do not have a NI constraint, so `c` must be empty.

- We have lower and upper bounds:

```
lb = [2, 2]
ub = [7, 8]
```

Exercise 5 - MATLAB solution

- Optimization

```
[x,fval] = ga(@squarednorm,2,[],[],[],[],lb,ub,@circlecons);

   x = [5.6789     4.0994]
   fval = 49.0546
```

Exercise 5 – C solution

$$\min(f(X) + penalty(X)) = \max \frac{1}{f(X) + penalty(X)} = \max f^*(X)$$

$$f(X) = x_1^2 + x_2^2$$

$$penalty(X) = w\left((x_1 - 8)^2 + (x_2 - 6)^2 - 9\right)^2$$

$$f^*(X) = \frac{1}{x_1^2 + x_2^2 + w\left((x_1 - 8)^2 + (x_2 - 6)^2 - 9\right)^2}$$

With this C implementation, you have to embody the constraints in the fitness function! But remember that, at the end, we are still interested in the value of the original fitness function!

Exercise 5 – C solution

$$f^*(X) = \frac{1}{x_1^2 + x_2^2 + w\left((x_1-8)^2 + (x_2-6)^2 - 9\right)^2}$$

$w = 1000$

```
#define NVARS 2
…
void evaluate(void) {
    int mem;
    for (mem = 0; mem < POPSIZE; mem++) {
        valutation(mem);
        population[mem].fitness = 1 / (    x[0]*x[0]+x[1]*x[1] + 1000 * ( (x[0]-
    8)*(x[0]-8)+(x[1]-6)*(x[1]-6) - 9) *( (x[0]-8)*(x[0]-8)+(x[1]-6)*(x[1]-6) - 9) );
    }
}
```

Exercise 5 – C solution

After the optimization we get the following results.

$$\max f^*(X) = max \frac{1}{x_1^2 + x_2^2 + w\left((x_1 - 8)^2 + (x_2 - 6)^2 - 9\right)^2} \qquad w = 1000$$

$$X = [5.5875, 4.2170]$$

$$f(X) = f(x_1, x_2) = x_1^2 + x_2^2 = 49.0032$$

$$f^*(X) = 0.020407$$

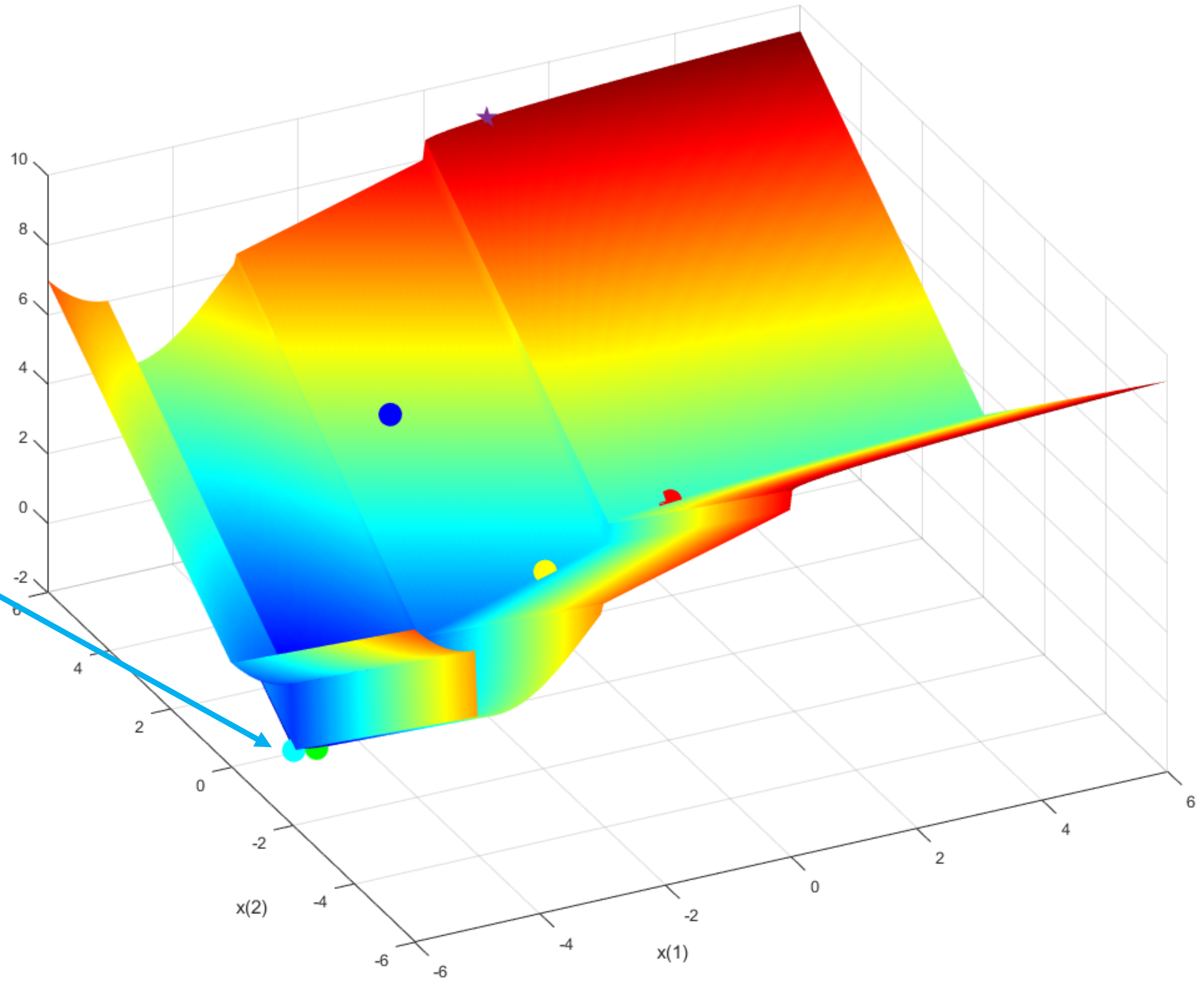
Integer Constraints

- `IntCon` – Integer Variables
- Integer variables, specified as a vector of positive integers taking values from 1 to `nvars`. Each value in `IntCon` represents an x component that is integer-valued.
- Example: we want x_1 to be constrained to be an integer-valued variable.
- Optimization
  - `x = ga(@ps_example,2,[],[],[],[],[],[],[],1)`

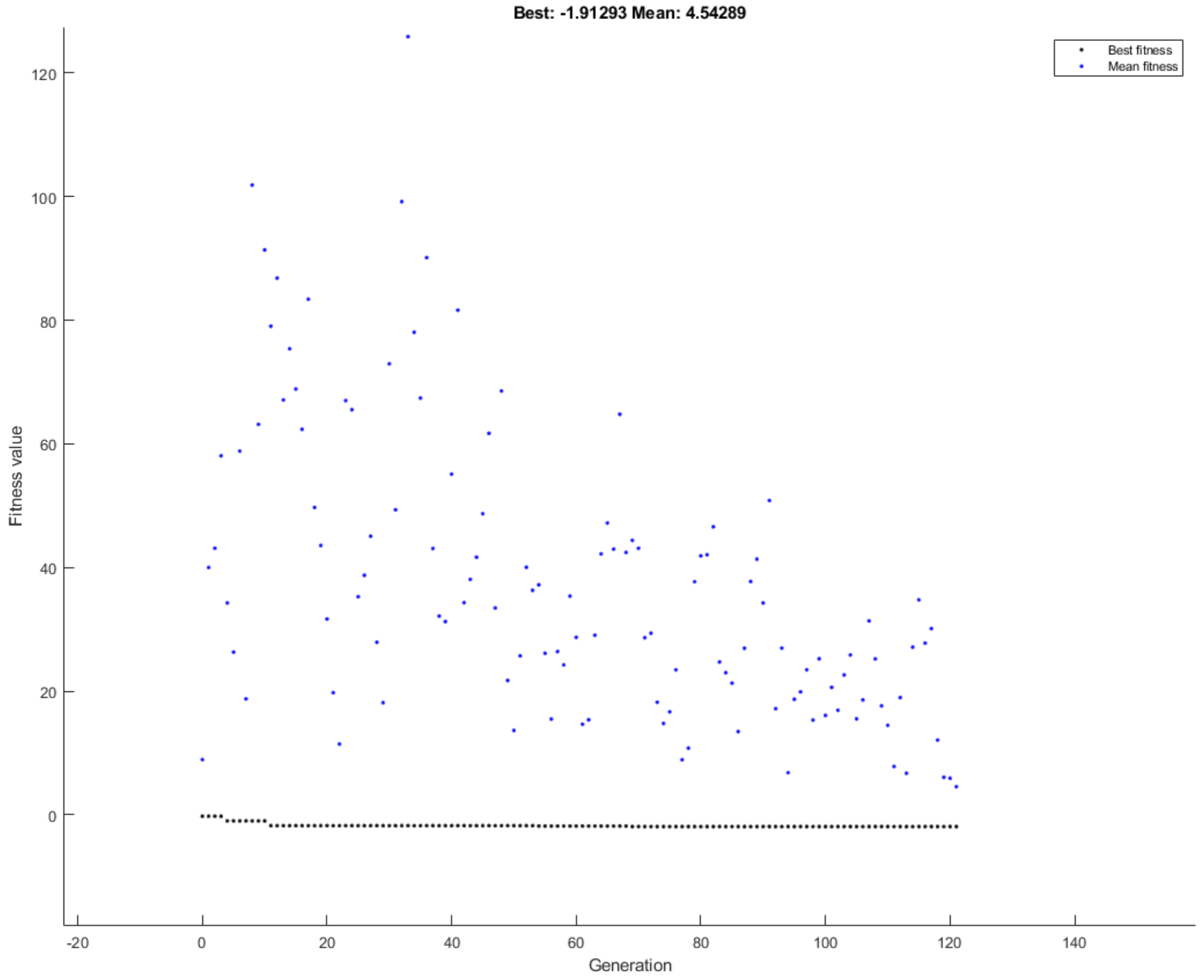
Optimization with Integer Constraints

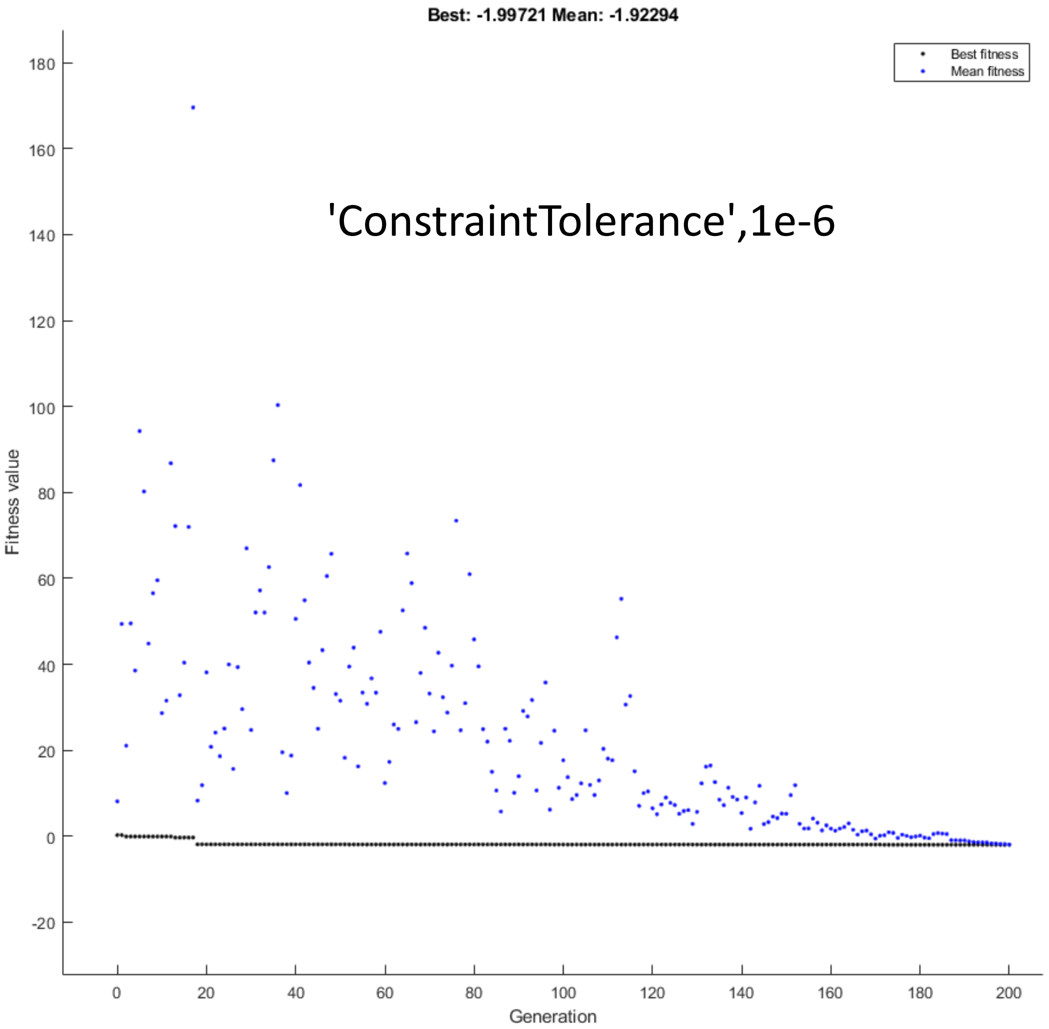
Minimum found by GA with Integer Constraints

x = [-5.0000, -0.0000]
f(x) = -1.9178



Monitor solver progress





Algorithm hyperparameters tuning

- **Some** of the hyperparameters you can tune using MATLAB `ga()`:
  - `EliteCount`
    - Positive integer specifying how many individuals in the current generation are guaranteed to survive to the next generation.
  - `FitnessLimit`
    - If the fitness function attains the value of `FitnessLimit`, the algorithm halts.
  - `ConstraintTolerance`
    - Determines the feasibility with respect to constraints.
  - `MaxGenerations`
    - Maximum number of iterations before the algorithm halts.
  - `PopulationSize`
    - Size of the population.

Hyperparameters Tuning

- **Exercise 6**: try to tune options and see how it affects the optimization process.

Multiobjective Optimization Problem

- A multi-objective optimization problem is an optimization problem that involves multiple objective functions.

$$\min(f(X)) = \min(f_1(X), f_2(X), \dots, f_k(X))$$

$$f(X) = (f_1(X), f_2(X), \dots, f_k(X)) \in \mathbb{R}^k$$

$$X \in F \subseteq S$$

$$f(X) \in \mathbb{R}^k$$

- In multi-objective optimization, there does not typically exist a feasible solution that minimizes all objective functions simultaneously.
- Therefore, attention is paid to Pareto optimal solutions; that is, solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives.

Multiobjective Optimization with GA in MATLAB

- Function: `gamultiobj()`
- Parameters:
  - `fun`
    - The $f(X) = \left(f_1(X), f_2(X), \dots, f_k(X)\right) \in \mathbb{R}^k$ to optimize
  - `nvars`
    - $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$
  - `A, b`
    - Linear inequalities, in matricial form
  - `Aeq, beq`
    - Linear equalities, in matricial form
  - `lb, ub`
    - Lower and upper bounds
  - `nonlcon`
    - Nonlinear equalities and inequalities
  - `options`
    - Hyperparameters of the algorithm

Multiobjective function

- Define a vector-valued objective function:

$$f(X) = (f_1(X), f_2(X)) \in \mathbb{R}^2$$

$$f_1(X) = \|X\|^2 = x_1^2 + x_2^2$$

$$f_2(X) = \frac{1}{2}((x_1 - 2)^2 + (x_2 + 1)^2) + 2$$

- You can use a MATLAB lambda expression or use a dedicate file.

```
fitnessfcn = @(x)[norm(x)^2 , 0.5*norm(x(:)-[2;-1])^2+2];
```

Constrained optimization with MATLAB

- We want to impose the following linear inequality:

$$x_1 + x_2 \leq \frac{1}{2}$$

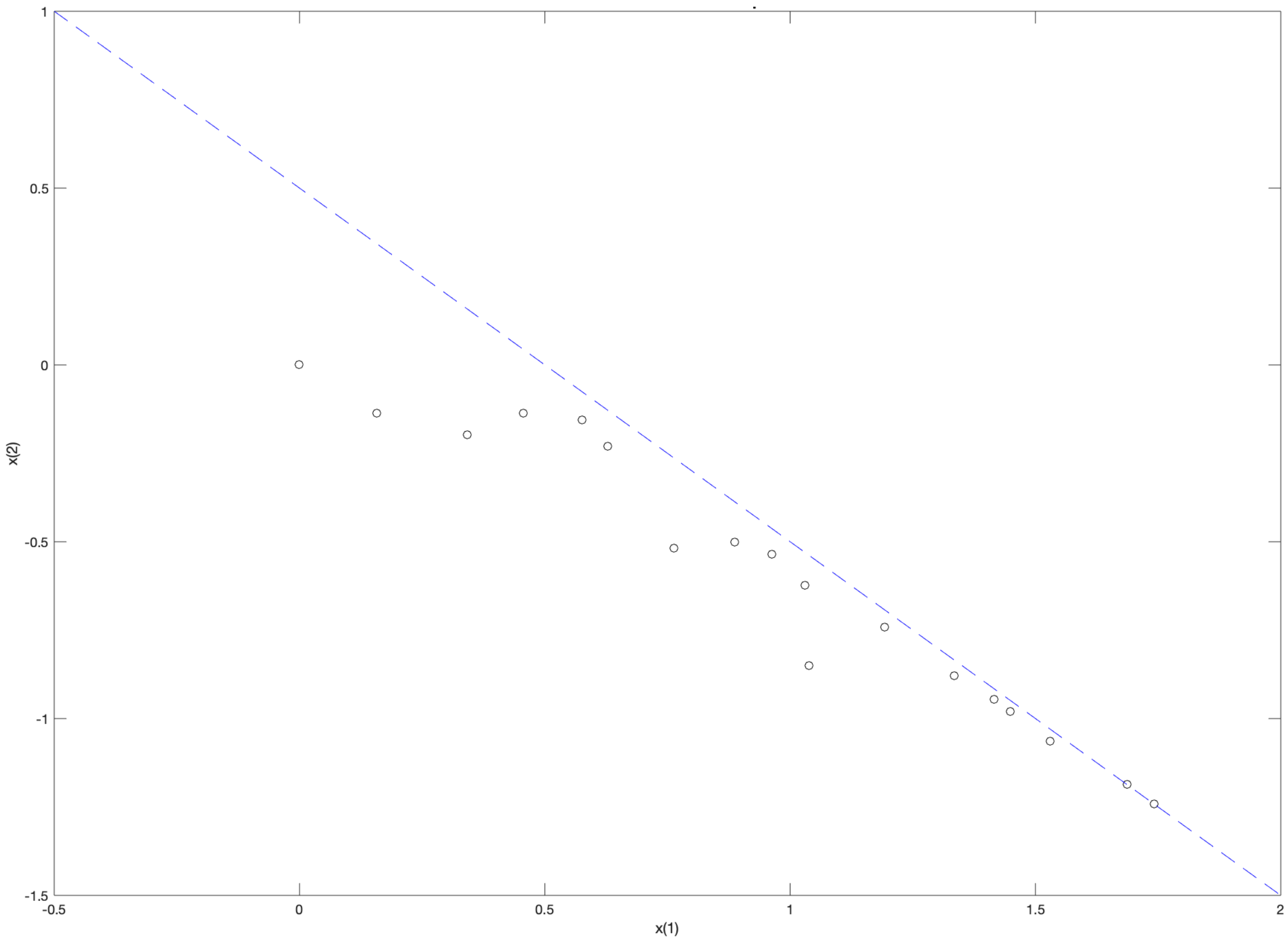
- As usual, we define A and b:

```
A = [1,1]
b = 1/2
```

- Optimization:

```
[x, fval] = gamultiobj(fitnessfcn,2,A,b);
```

Pareto Points in Parameter Space



Bound constraints

Optimize:

$$f(x) = \left(f_1(x), f_2(x)\right) = (\sin(x), \cos(x))$$

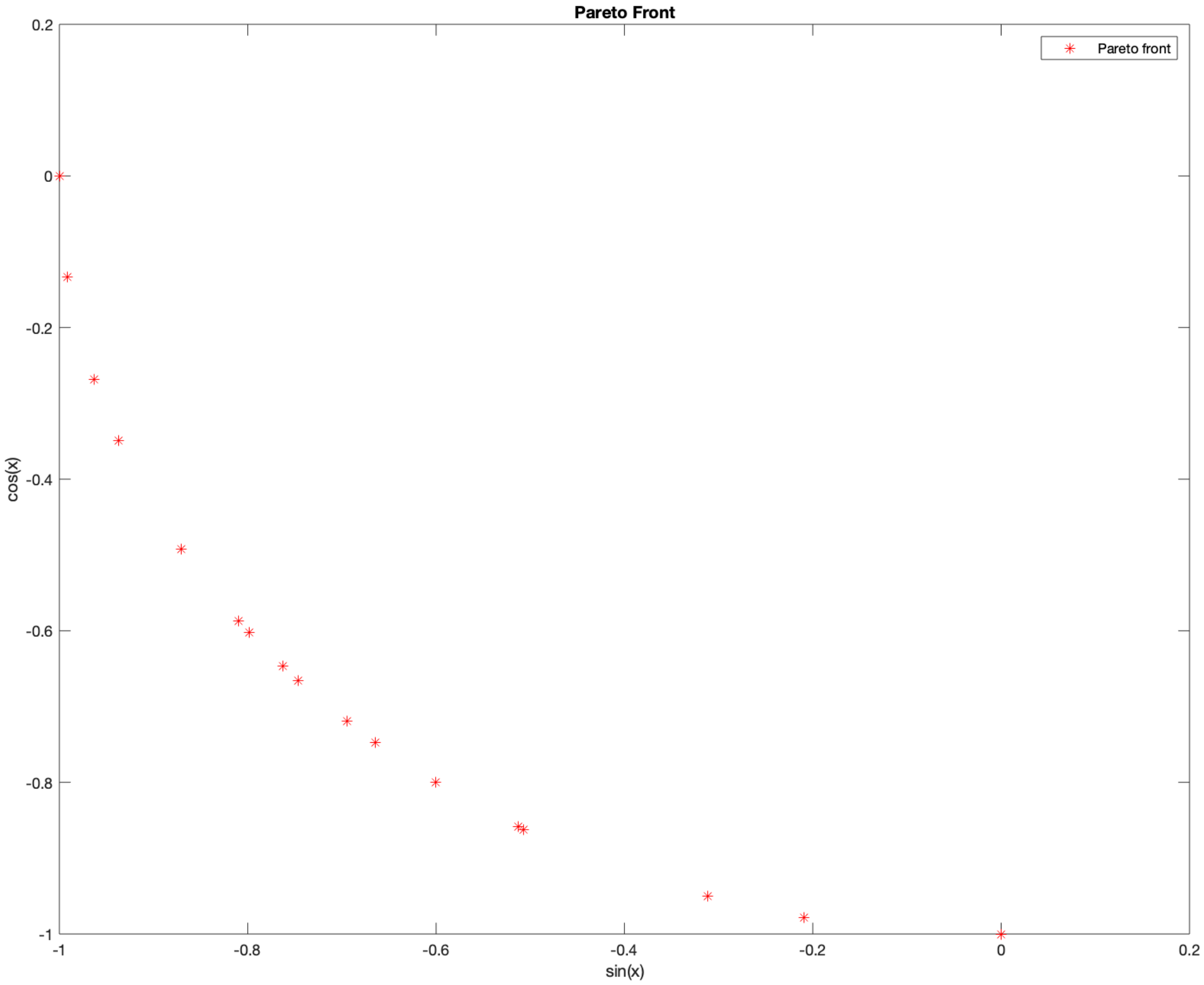
With the constraint:

$$0 \leq x \leq 2\pi$$

In MATLAB:

```
fitnessfcn = @(x)[sin(x),cos(x)];
nvars = 1;
lb = 0;
ub = 2*pi;
[x, fval] = gamultiobj(fitnessfcn,nvars,[],[],[],[],lb,ub);
```

Pareto Front



References

- MATLAB Documentation
  - *Global Optimization Toolbox*.
    URL: https://it.mathworks.com/help/gads/
  - *Genetic Algorithm for Mono-objective Optimization*.
    URL: https://it.mathworks.com/help/gads/ga.html
  - *Genetic Algorithm for Multi-objective Optimization*.
    URL: https://it.mathworks.com/help/gads/gamultiobj.html
- Zbigniew Michalewicz and Girish Nazhiyath, *"Genocop III: A Co-evolutionary Algorithm for Numerical Optimization Problems with Nonlinear Constraints"*.
  URL: https://cs.adelaide.edu.au/~zbyszek/Papers/p24.pdf
- Deb, Kalyanmoy. *"Multi-Objective Optimization Using Evolutionary Algorithms"*. Chichester, England: John Wiley & Sons, 2001.