Using Evolutionary Algorithms for Finding Optimal Initial Conditions in a Zero-Player Game

CL643

Guided by Prakash Kotecha

Rohit Priyanka Nandwani

180121035

Agenda

- Background
- Project Description
- Objective Function – 1
- Objective Function – 2 [Objective 1 + time-step minimization]
- Brute Force (First Objective)
- Comparison of a few Existing Techniques (First Objective)
- Comparison of a few Existing Techniques (Second Objective)
- Conclusion

Background

- **Zero-Player Game**

A game that evolves as *determined by its initial state*, requiring no further input from humans is considered a zero-player game.

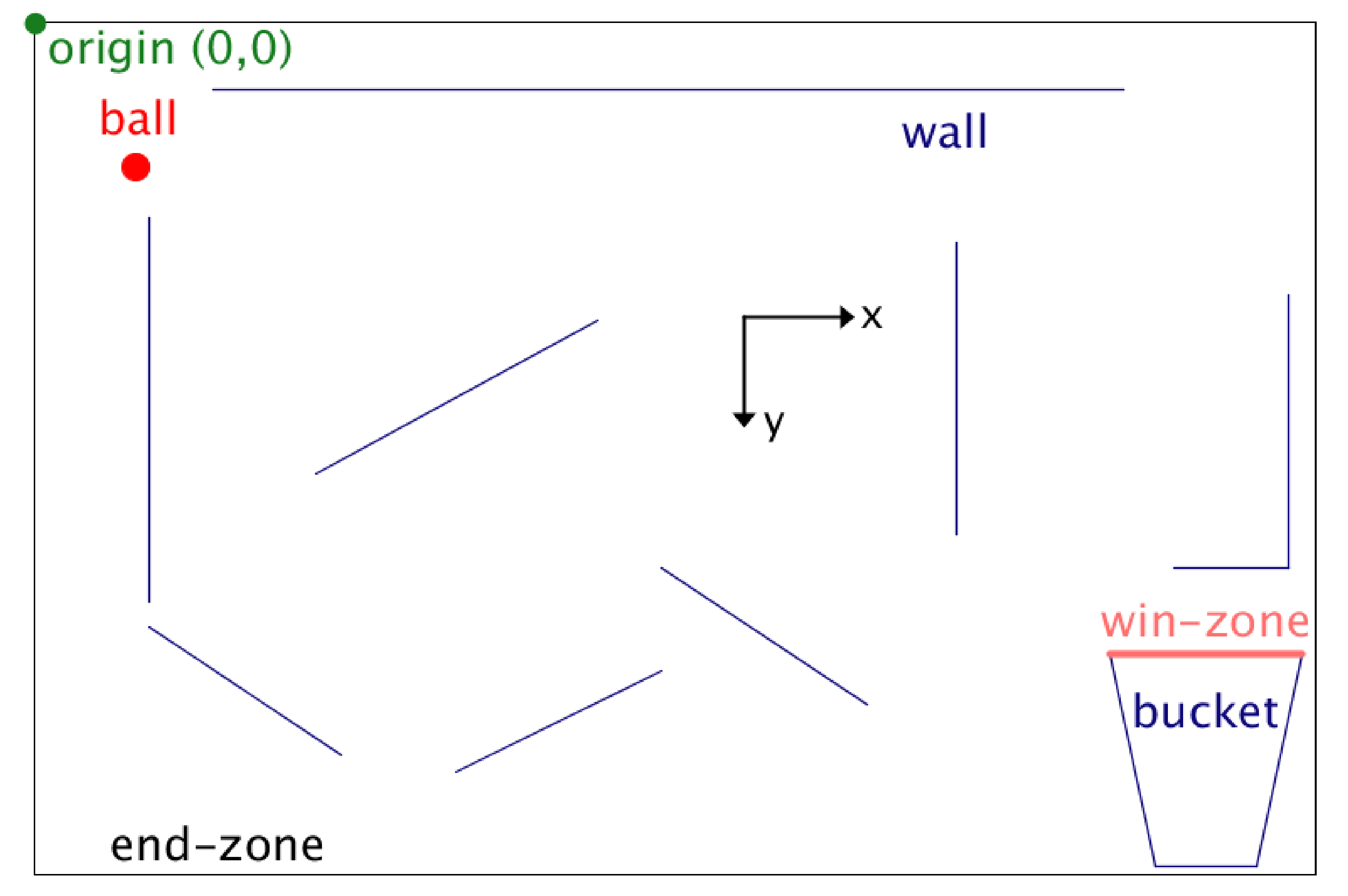
Cellular automaton games that are determined by initial conditions including Conway's Game of Life are examples of this.

Here, we deal with such a physics based zero-player maze game where a ball shooted from a fixed points needs to be given initial velocity. After that, the ball bounces off the walls in the environment. The game is “won” if the ball goes inside a bucket instead of escaping to infinity.

Project Description

Assumptions:

- Coefficient of restitution, $e = 1$
- Total of 12 walls as shown in the figure
- Ball departure point is fixed for all throws
- Fixed boundary as a end-zone
- Lid of the bucket as win-zone
- Ball disappears after reaching end-zone or win-zone; and the simulation ends



Project Description

Objectives

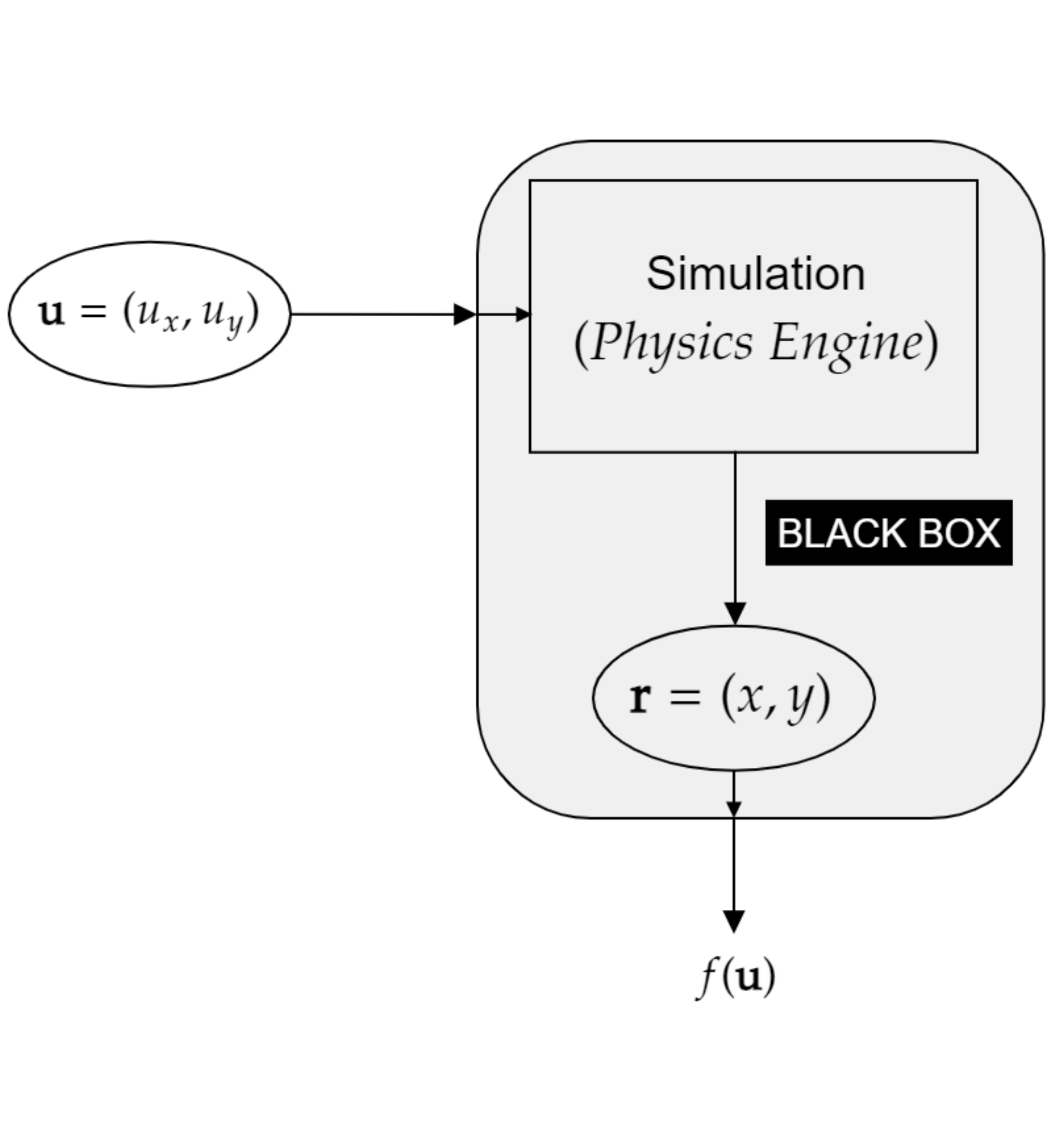
- Objective-1: The ball should reach the win-zone (i.e- the bucket).
- Objective-2: The ball should reach the win-zone in the least amount of time possible.

Objective Function-1

- Decision variables : $\mathbf{u} = (u_x, u_y)$
- Final coordinate of ball before ball disappears $(\boldsymbol{r})$
- Midpoint of win-zone $(\boldsymbol{r_c})$
- Objective function

$$f(\mathbf{u}) = \begin{cases} \mathrm{d}(\mathbf{r},\mathbf{r}_c) & ,\ \mathbf{r} \in \text{win-zone} \\ \lambda \cdot \mathrm{d}(\mathbf{r},\mathbf{r}_c) & ,\ \mathbf{r} \notin \text{win-zone} \end{cases}$$

, where the factor of $\lambda (= 10)$ is introduced to create a bias towards the win-zone, in contrast to the remaining end-zone.



Objective Function-2

- Decision variables : $\mathbf{u} = (u_x, u_y)$
- Final coordinate of ball before ball disappears $(\boldsymbol{r})$
- Midpoint of win-zone $(\boldsymbol{r}_c)$
- Fitness function

$$fit(\mathbf{u}) = \begin{cases} \mathrm{d}(\mathbf{r},\mathbf{r}_c) + \gamma_0 (ts)^{\beta_0} & , \mathbf{r} \in \text{win-zone} \\ \lambda \cdot \mathrm{d}(\mathbf{r},\mathbf{r}_c) + \gamma_1 (ts)^{\beta_1} & , \mathbf{r} \notin \text{win-zone} \end{cases}$$

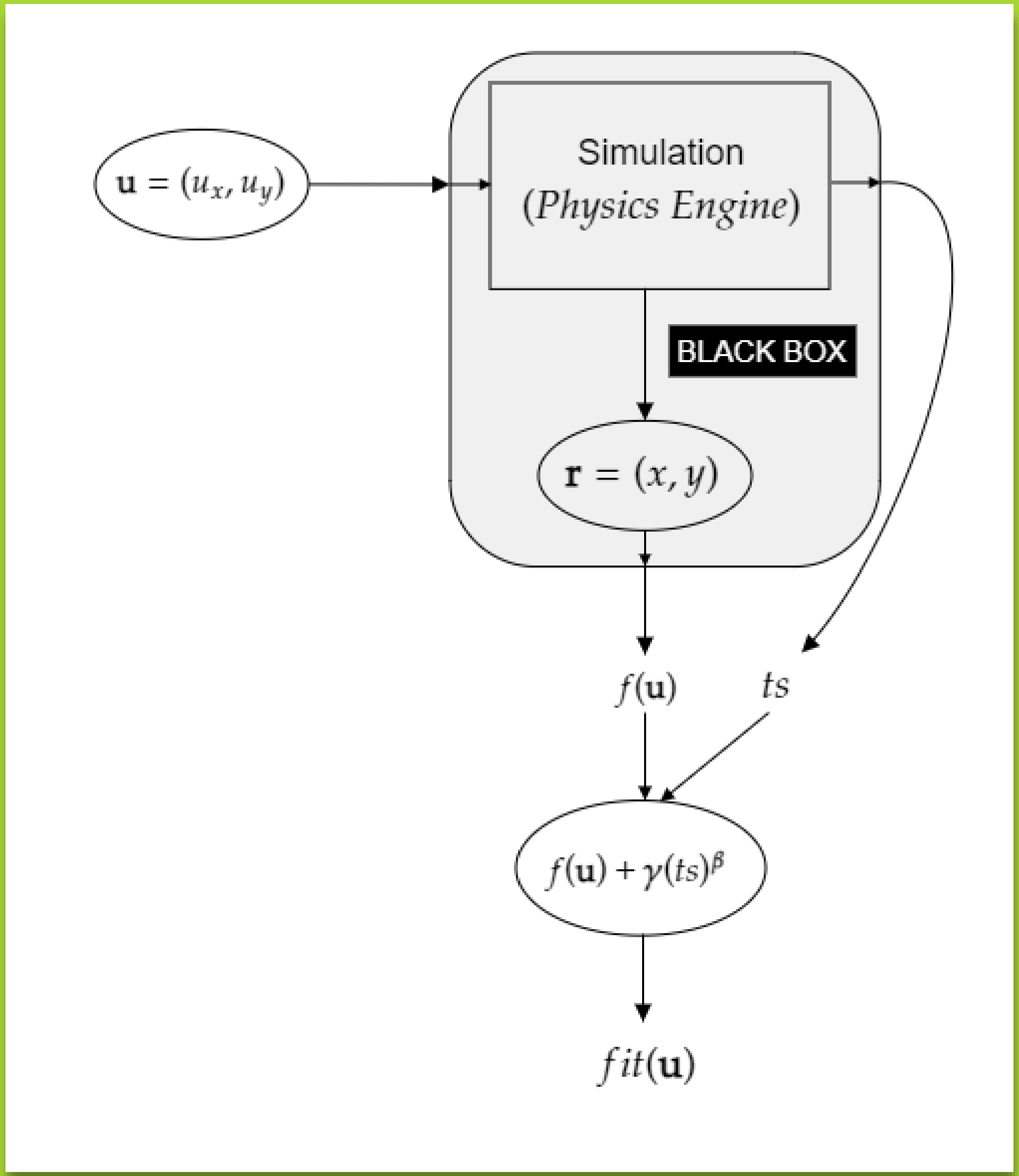
$$= f(\mathbf{u}) + \boldsymbol{\gamma}(ts)^{\boldsymbol{\beta}}$$

, *where*

$$ts = \text{number-of-time-steps}$$

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_0 \\ \gamma_1 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



First Approach

Brute force shallow search

Brute Force Shallow Search

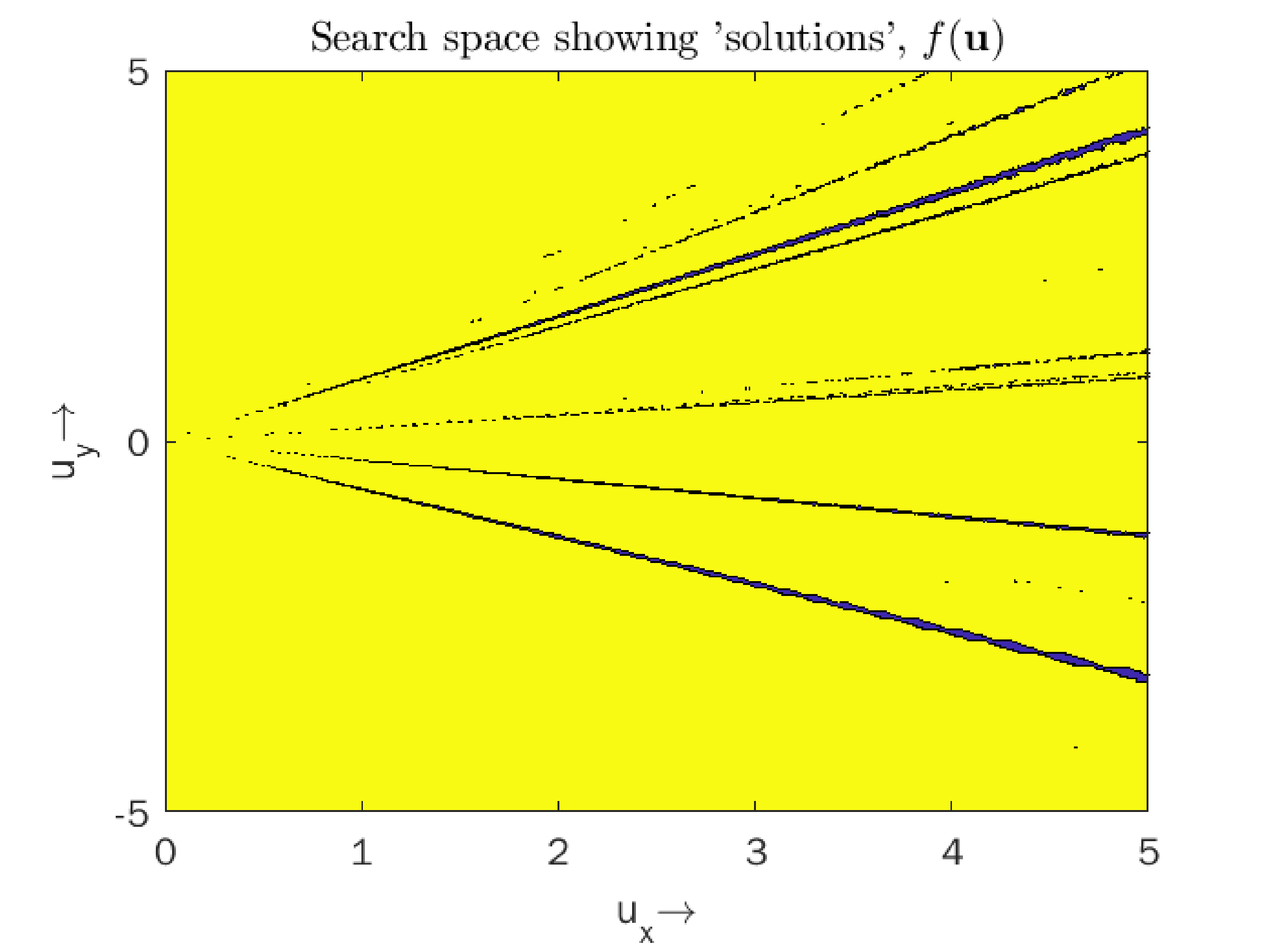
Assumptions

- Range of $u_x = [0,5]$
- Range of $u_y = [-5,5]$
- Step-Size = 0.01

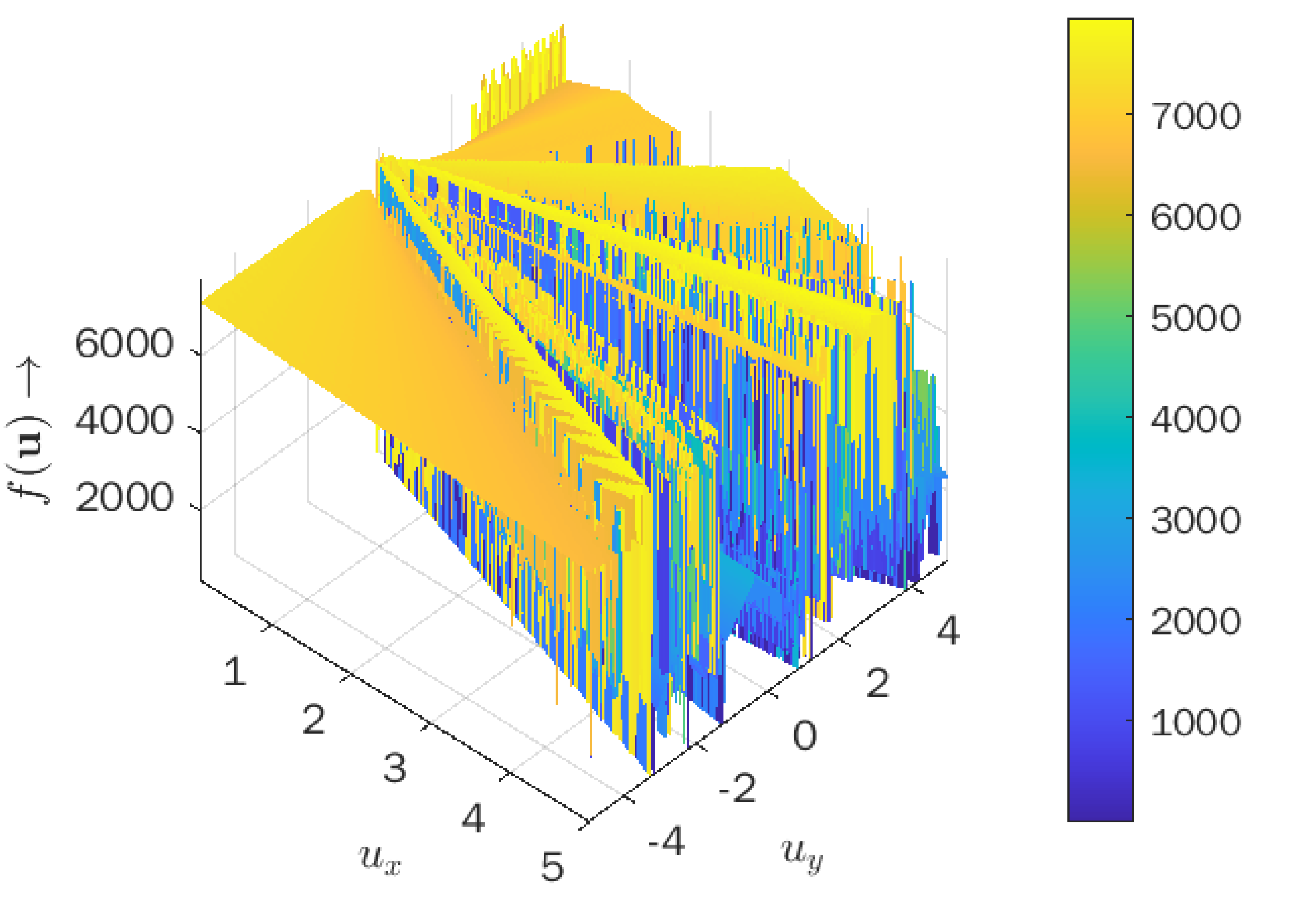
Requirement

u is a solution if and only if $f(\mathbf{u})$ is atmost 57 as per the given boundaries.

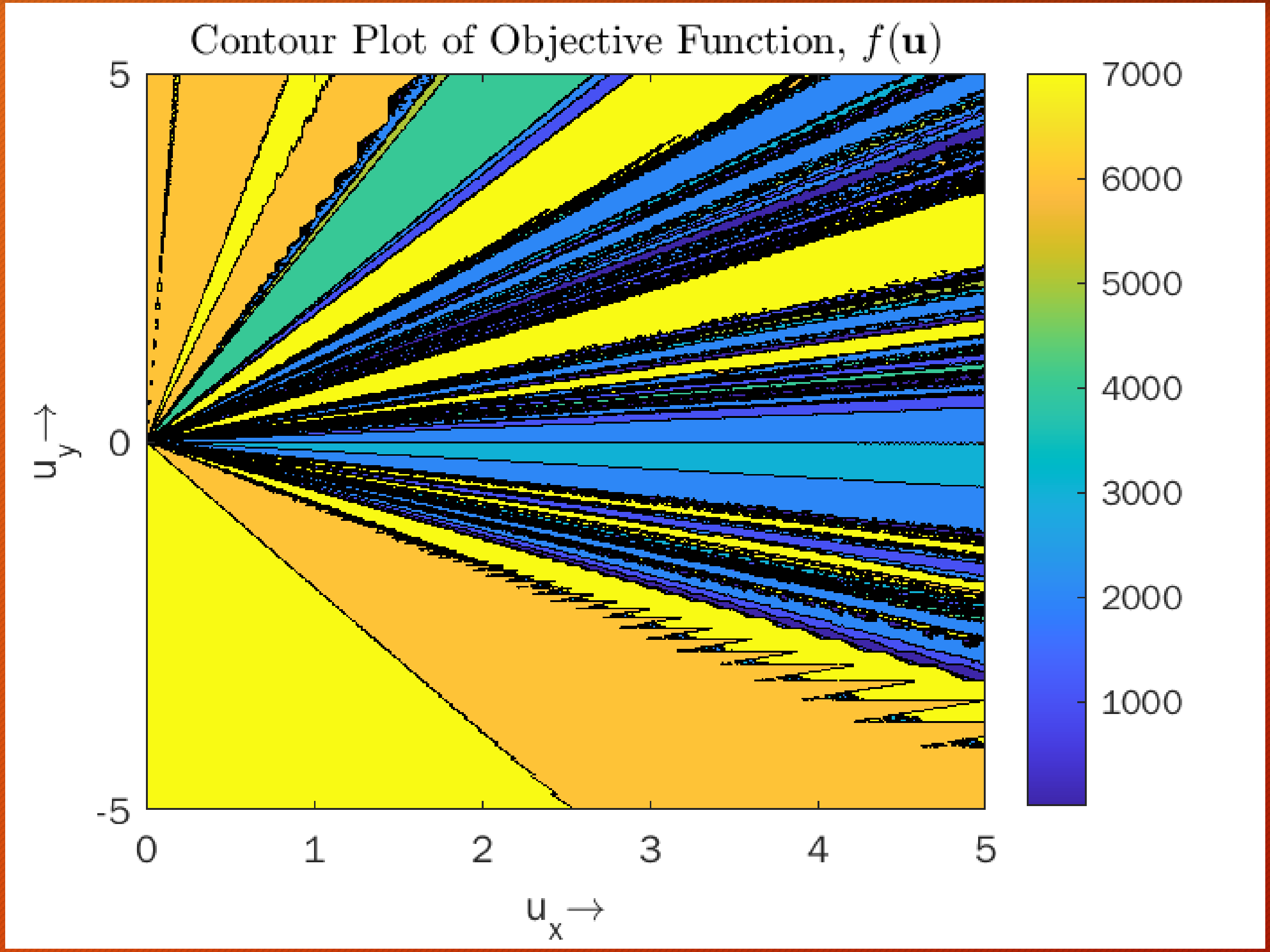
Search space showing 'solutions', $f(\mathbf{u}) \leq 57$



Nature of Objective Function (3D)



Nature of Objective Function (2D Contour Plot)



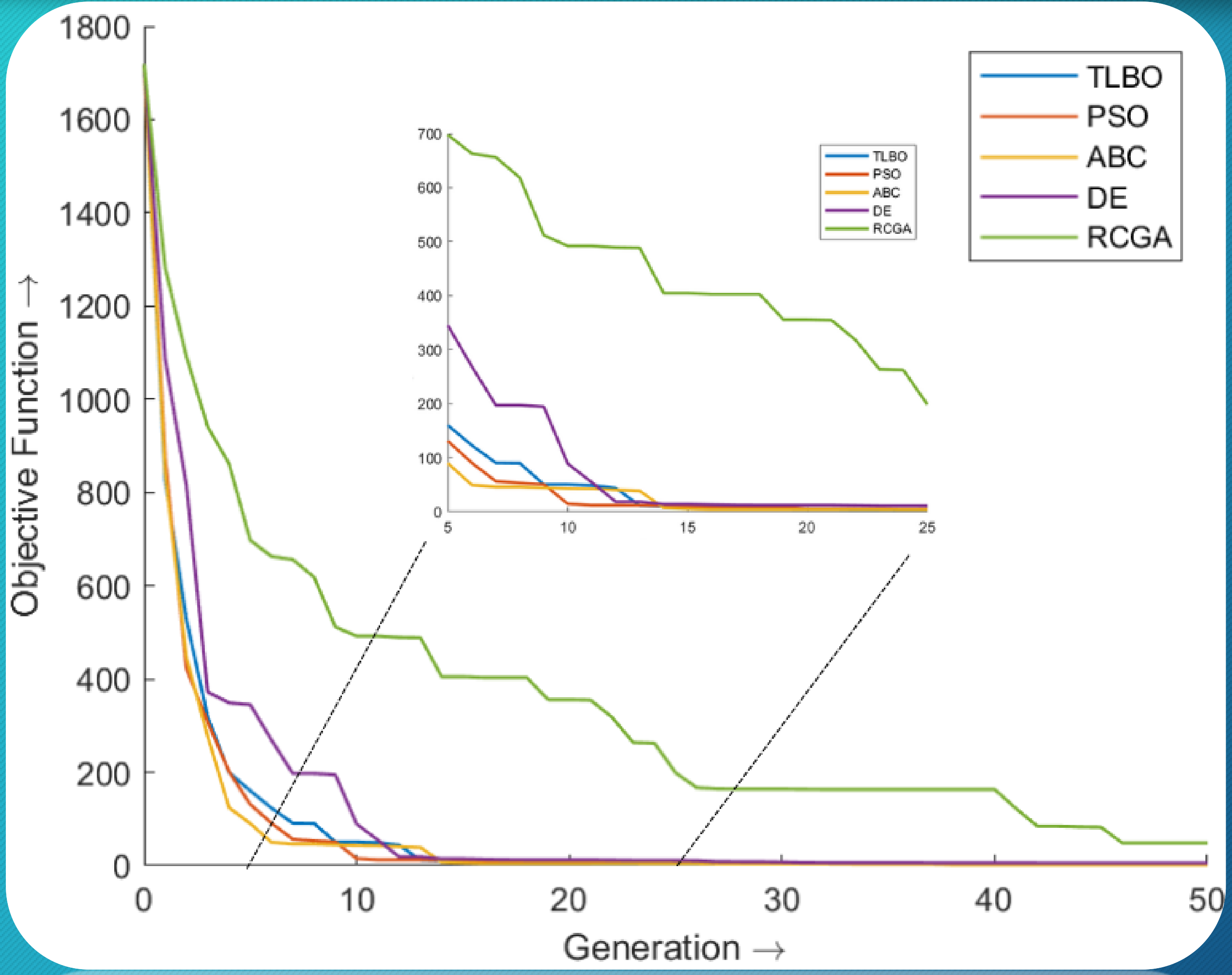
Comparison of Algorithms

With Objective Function - 1

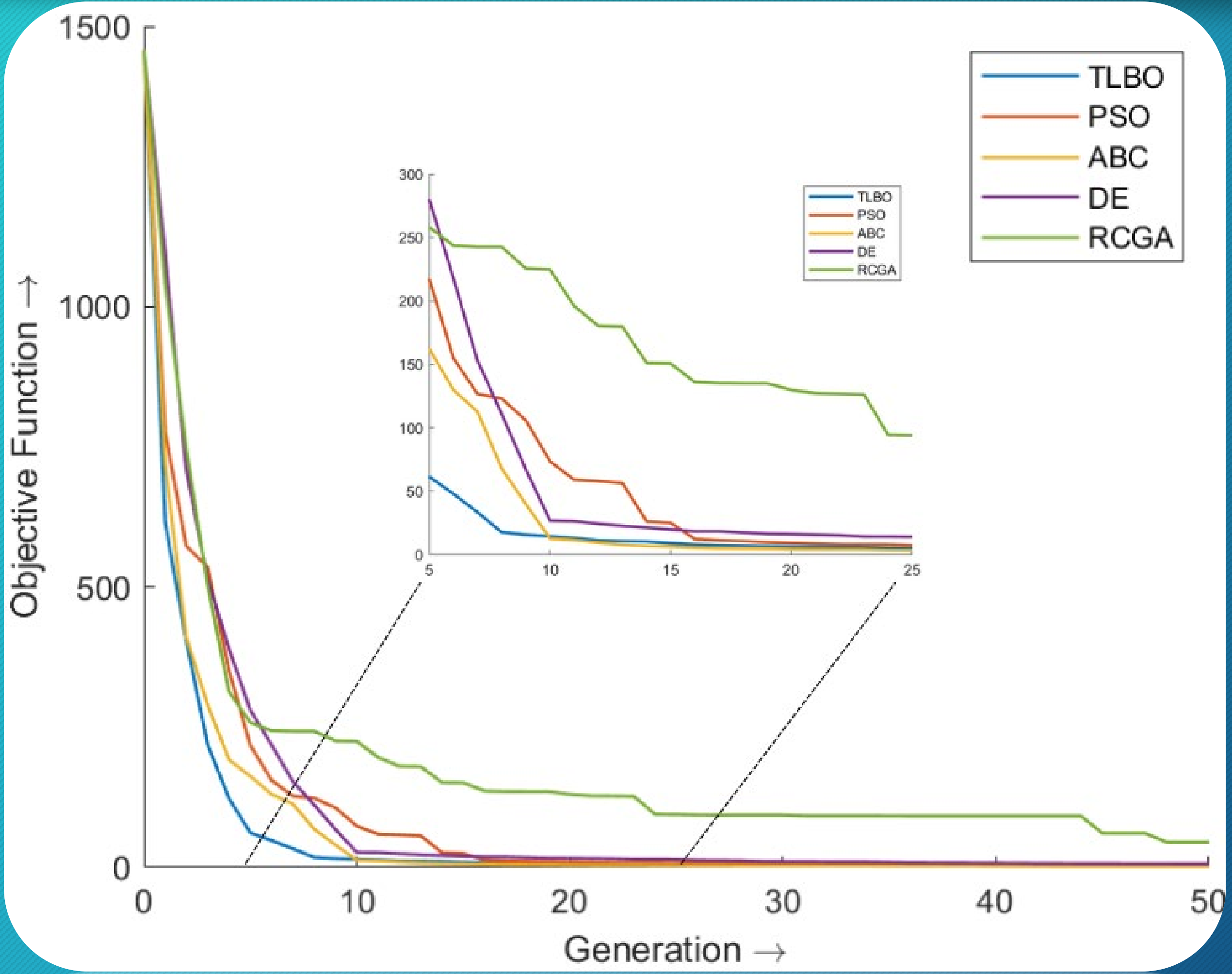
Used Metaheuristic Techniques

Algorithm	Parameters
TLBO (Teacher and Learning based Optimization)	$N_p = 10;\ \ T = 50;\ \ \#FE = 1010$
PSO (Particle Swarm Optimization)	$N_p = 10;\ \ T = 50;\ \ w = 0.7298;$ $c_1 = 1.496;\ \ c_2 = 1.496;\ \ \#FE = 510$
DE (rand/1/binomial)	$N_p = 10;\ \ T = 50;\ \ p_c = 0.8;$ $F = 0.8;\ \ \#FE = 510$
RCGA (Real Coded Genetic Algorithm)	$N_p = 10;\ \ T = 50;\ \ \eta_c = 20;\ \ \eta_m = 20;$ $p_c = 0.8;\ \ p_m = 0.2;\ \ \max \#FE = 510$
ABC (Artifical Bee Colony)	$N_p = 10;\ \ T = 50;\ \ limit = 50;$ $\max \#FE = 1060$

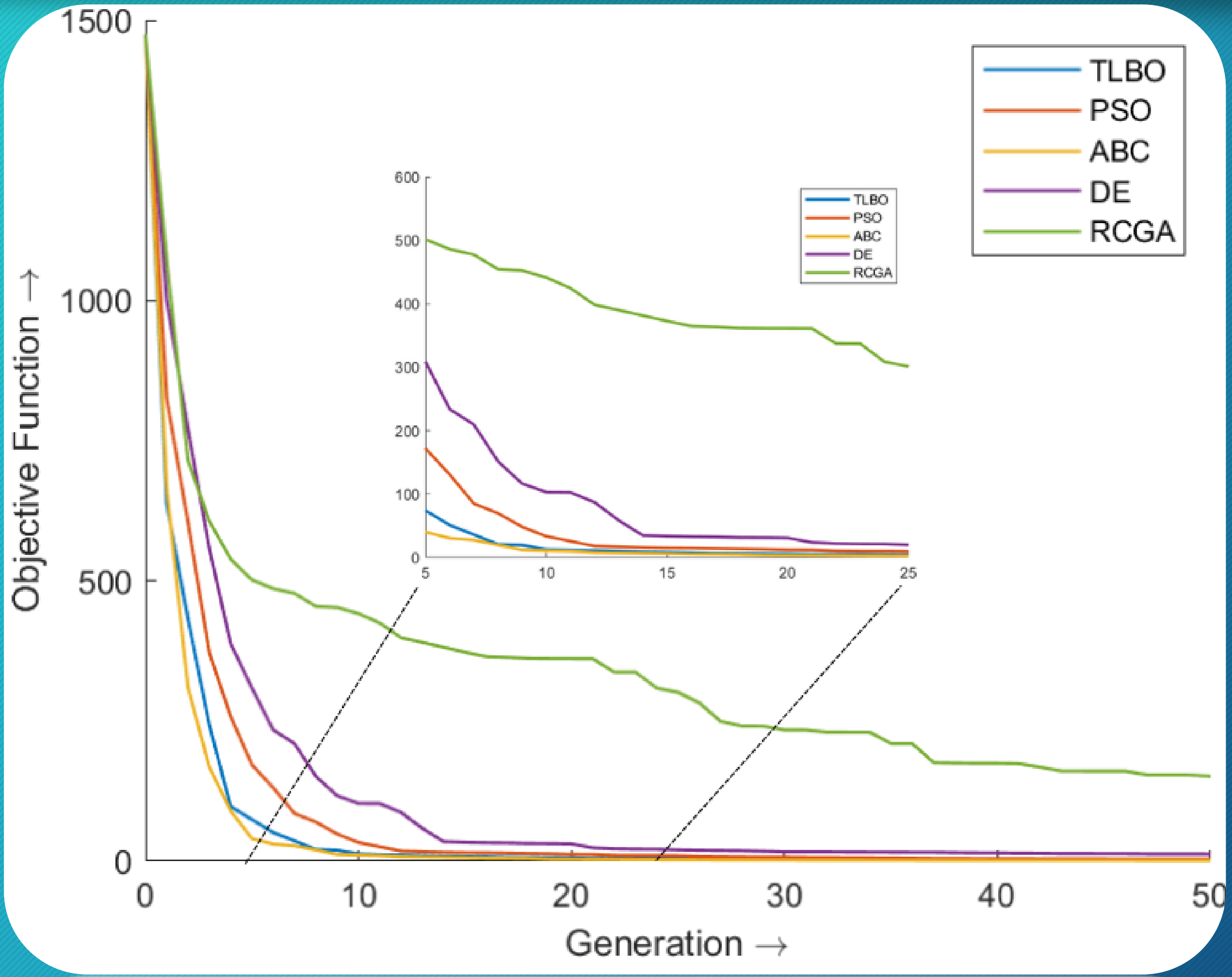
Mean of best objective function across 20 runs



Mean of best objective function across 50 runs

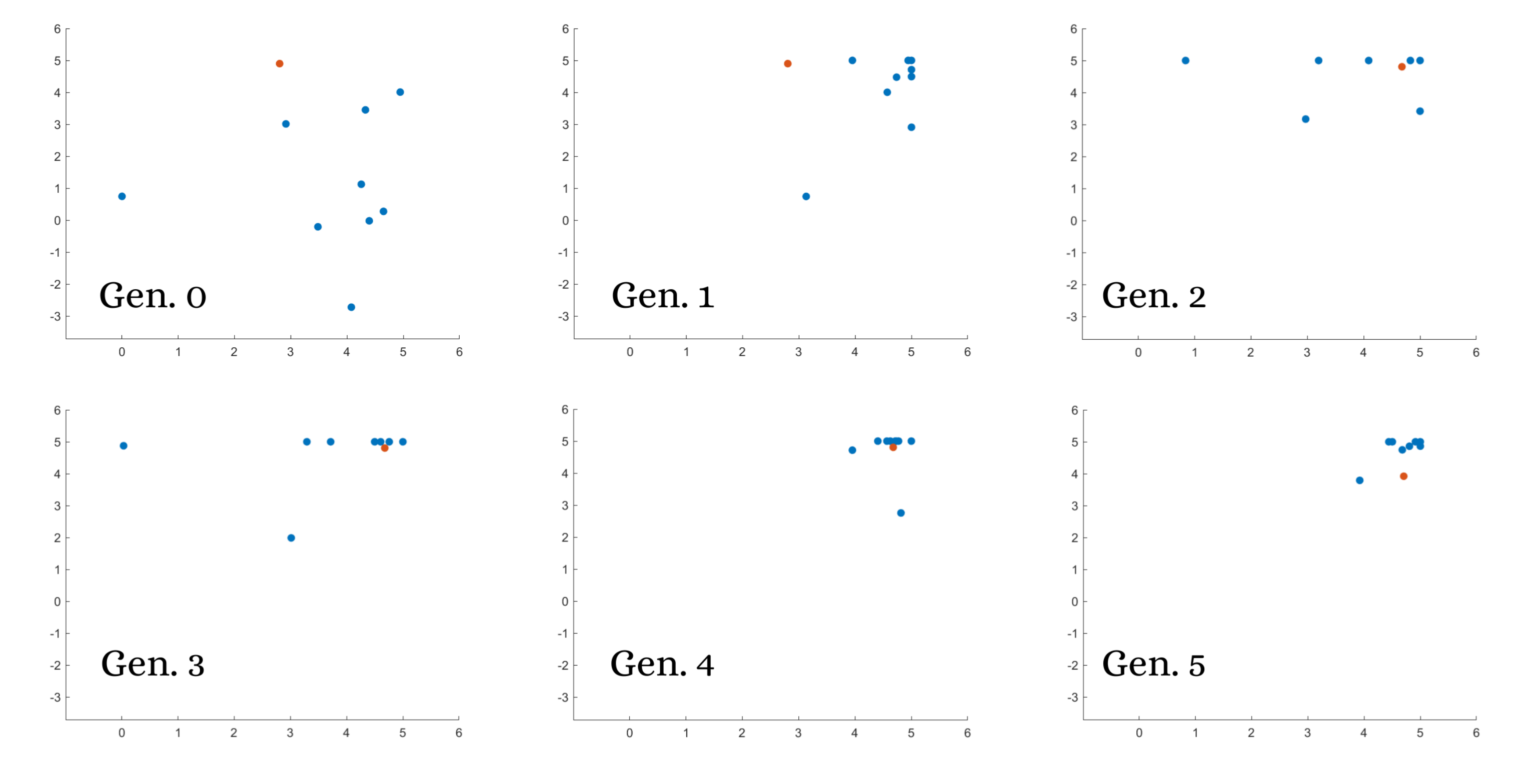


Mean of best objective function across 100 runs



Summary

- Mean behavior of RCGA (for chosen parameters) gave the worst result.
- TLBO, PSO and ABC perform the best.
- Considering #FE, the "best" algorithm for this problem would be PSO.



Actual Evolution of Simulation (animated)

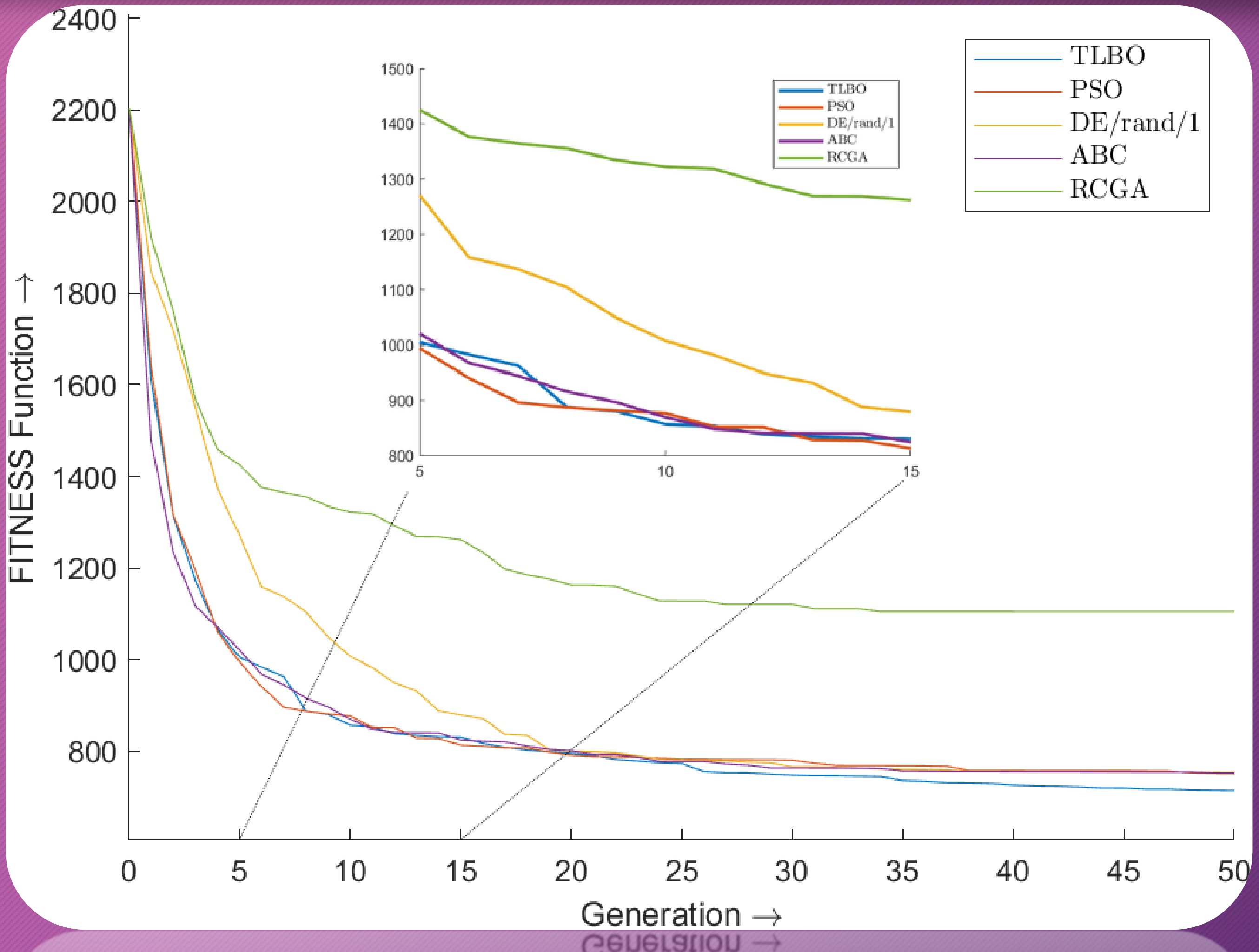
Comparison of Algorithms

With Objective Function - 2

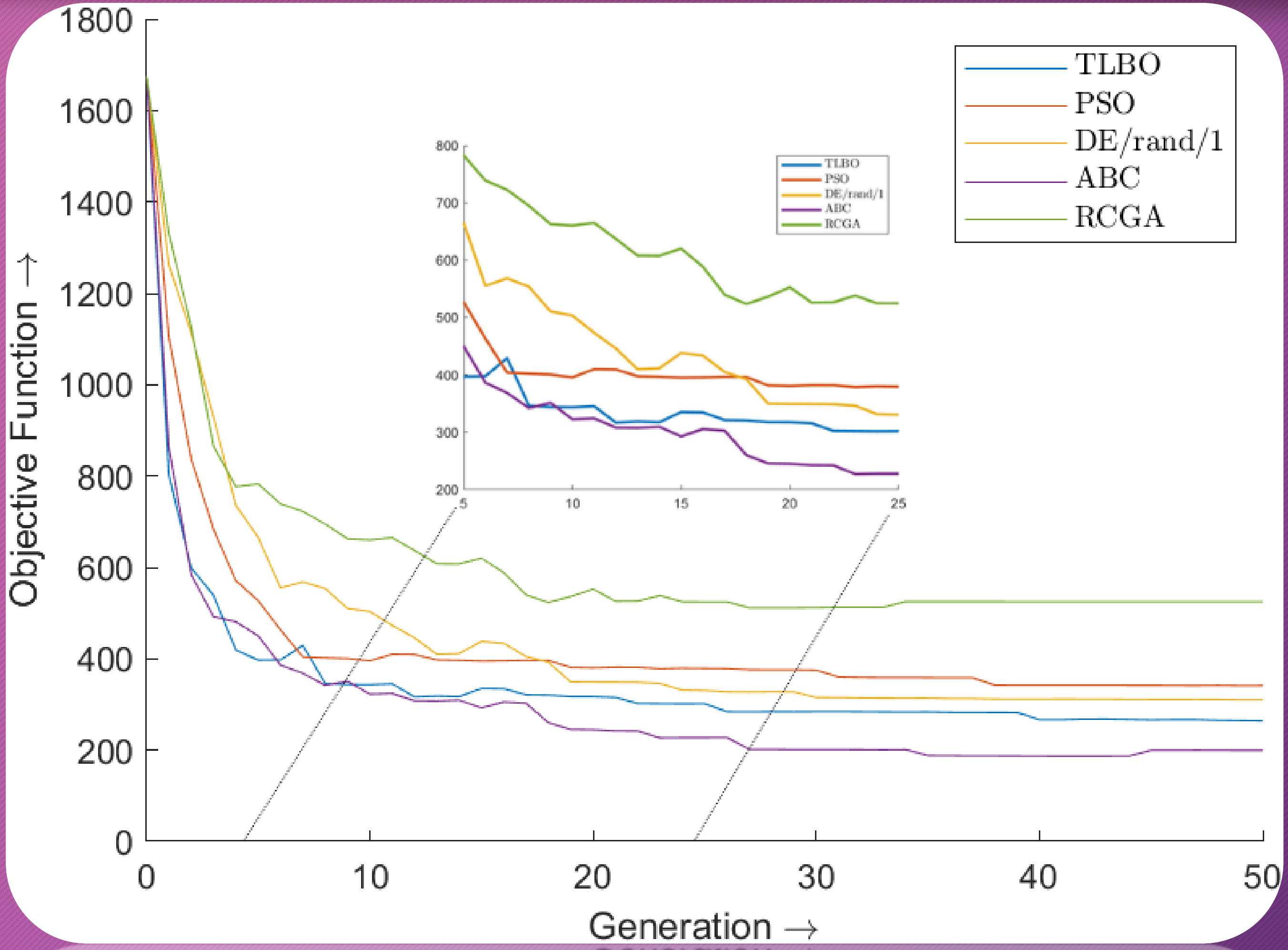
Used Metaheuristic Techniques

Algorithm	Parameters
TLBO (Teacher and Learning based Optimization)	$N_p = 10;\ \ T = 50;\ \ \#FE = 1010$
PSO (Particle Swarm Optimization)	$N_p = 10;\ \ T = 50;\ \ w = 0.7298;$ $c_1 = 1.496;\ \ c_2 = 1.496;\ \ \#FE = 510$
DE (rand/1/binomial)	$N_p = 10;\ \ T = 50;\ \ p_c = 0.8;$ $F = 0.8;\ \ \#FE = 510$
RCGA (Real Coded Genetic Algorithm)	$N_p = 10;\ \ T = 50;\ \ \eta_c = 20;\ \ \eta_m = 20;$ $p_c = 0.8;\ \ p_m = 0.2;\ \ \max \#FE = 510$
ABC (Artifical Bee Colony)	$N_p = 10;\ \ T = 50;\ \ limit = 50;$ $\max \#FE = 1060$

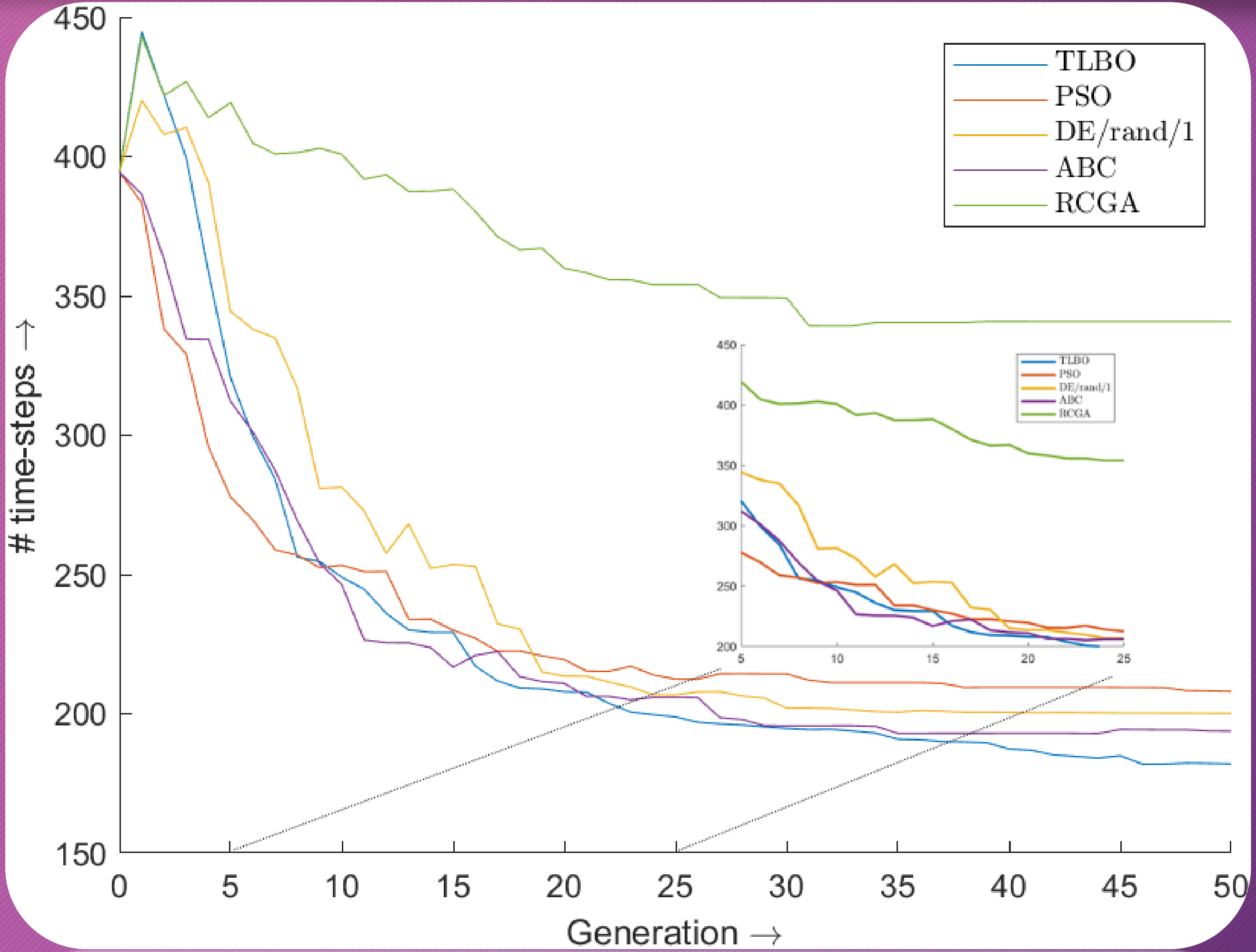
Mean of Best Fitness Function Values, $fit(\mathbf{u})$ across 50 runs



Mean of Corresponding Objective Function Values, $f(\mathbf{u})$ across 50 runs



Mean of Corresponding “time-step” Values, $ts(\mathbf{u})$ across 50 runs



Project Summary

Conclusion

- Assigned an objective function to an abstract and complicated physics-based zero-player game.
- Applied and Compared existing metaheuristic techniques to this objective function.
- The algorithms performed well, and the solution was reached within a few generations.
- Added an additional objective of minimization of the time taken for the ball to end up in the end-zone.
- Used weighted-sum method (albeit in a piece-wise sense) for this “multi-objective” optimization, choosing the weights (representing the relative importance of objectives).
- Possible future endeavour – try to add complex problems (*inelastic walls, massy finite-sized ”planets” with gravitational fields, massless finite-size ”charges” with electrostatic field and viscous mediums)* and deploy a GUI on the web to add to existing rich text of “applicability of metaheuristic techniques” another beautiful (and definitely esoteric!) example.

References

- L. Zadeh, ”Optimality and non-scalar-valued performance criteria,” in IEEE Transactions on Automatic Control, vol. 8, no. 1, pp. 59-60, January 1963, doi: 10.1109/TAC.1963.1105511.
- Martin Gardner (October 1970), ”Mathematical games: The fantastic combinations of John Conway’s new solitaire game ’Life’”, Scientific American
- Reas, C. and Fry, B. Processing: programming for the media arts (2006). Journal AI Society, volume 20(4), pages 526-538, Springer