Game Theory in Differential and Genetic Algorithm

CL643 - Computer Aided Applied Optimization

1st Chitrabanu T
Electrical and Electronics Department
Indian Institute of Technology
Guwahati, India
t.chitrabanu@iitg.ac.in

2nd Mridul Garg
Department ofMathematics
Indian Institute of Technology
Guwahati, India
mridul18@iitg.ac.in

***Abstract*—Applying Game theory in evolutionary algorithms like Differential Evolution and Genetic Algorithm to obtain a Pareto optimal set and a single compromised optimal solution respectively.**

I. Introduction

Given the omni-present nature of optimisation, most of these real-world engineering systems can be formulated as complex models that consist of challenging characteristics such as non-linearity, multi-modality, discontinuousness and non-differentiability. To solve these complicated models successfully, optimisation is a decision-making process that seeks for the optimal combinations of decision variables that maximize or minimize the stated objective functions while satisfying all technical and non-technical restrictions in an acceptable amount of time. Traditional optimizer like Linear programming have a number of flaws, including low global strength, poor initial solution guessing, and a high reliance on gradient information, all of which can limit their ability to solve a variety of modern engineering optimization problems of growing complexity. Meta-heuristic search algorithms (MSAs) are here to replace the conventional algorithms by inspiring search mechanism process from nature.

Given its advantages DE and GA still lacks in providing optimal solution to complex multi-objective problems. This paper tries to combine DE and game theory to optimize multi-objective optimization problems.

II. Differential Evolution

Differential Evolution algorithm is an algorithm based on natural selection process that mimics biological evolution. It has been studied extensively to solve different areas of optimisation and engineering applications since its introduction by Storn in 1997. DE is one of the most adaptable and stable population-based search algorithms available, with multi-modal problem resilience.The DE finds the real global minimum of a multi-modal search space regardless of the initial parameter values, it is quick, and it only uses a few control parameters.The approach is ideally suited to non-differential nonlinear objective functions since it does not employ gradient information in the search. Added to that fact is that the implementation of DE is simpler and more straightforward.

The performance of DE is sensitive to the choice of the mutation strategy and associated control parameters.

DE is very similar to genetic algorithm by the fact that they both involve crossing over and mutating their chromosomes to generate new population. In addition Differential Evolution algorithm provides local search. Further the old population have equal chances to be selected unlike genetic algorithm. Many research have gone through in multi-objective space using Differential evolution to solve real world problems.They involve mutating the population followed by crossing over, It uses two types of crossovers, binomial and exponential.

In an iterative process, the population of each generation contains individuals. Suppose that the individual of generation is represented as

$$X_i = (x^1_{i,G}, x^2_{i,G}, ...) \tag{1}$$

A. Mutation

Differential Evolution uses the mutation operator for producing the donor vector vi for each individual xi in the current population. For each target vector xi = xi, 1, . . . , xi, D at generation G, the associated mutant vector vi = vi, 1, . . . , vi, D can be produced using a specific mutation scheme.

B. Crossover

In this phase, both the mutant and target vectors cross their components together in a probabilistic manner to produce a trial vector (offspring). This crossover process allows the target solution to inherit the attributes of the donor solution or mutant. Two commonly used crossover operators are known as uniform crossover and exponential crossover. The uniform crossover scheme is controlled by a crossover rate (CR) that has a value between [0,1].

It employs greedy selection to choose the solution to be added in the new population.

$$x^{new}_{i.G} = \begin{cases} U_{i,G} & if(f(U_{i,G} > f(x^{new}_{i.G}) \\ x^{new}_{i.G} & if(f(U_{i,G} \leq f(x^{new}_{i.G}) \end{cases} \tag{2}$$

III. Genetic Algorithm

Genetic Algorithm is subset of Evolutionary Algorithm. It is very similar to differential algorithm. It is one of the most used and researched heuristic algorithm. It also follows

Crossover and Mutation in the respective order like DE.The genetic algorithm (GA), was developed by John Holland and his collaborators in the 1960s and 1970s. There are two ways we could implement GA namely Binary and Real Ga.

It involves creating an initial random population each called chromosomes. Better descendants are tried starting from a random initial solution in an attempt to identify one that is the best under certain criteria and conditions. In chromosomes or solution representations, swapping sections of the solution with another is done and called crossover. The basic function is to enable mixing and convergence of solutions in a subspace.Next we do mutation that changes parts of one solution randomly, which increases the diversity of the population and provides a mechanism for escaping from a local optimum. A good solution to a given problem is an individual who is likely to succeed in a given environment.

These procedures eventually result in a population in the next generation that differs from the first. Because only the best creatures from the first generation are picked for breeding, along with a small fraction of less fit solutions, the population's average fitness will have increased. These less suitable strategies assure genetic variation within the parental genetic pool, and thus genetic diversity in following generations of children.

IV. Multi Objective Problems

Multi-objective optimization involves different functions being optimized on the same decision region. Oftentimes it involves large calculations which do not essentially converge to an optimal point after many iterations. A multi-objective can have an ideal solution or efficient solution. An ideal solution is the best of all objective functions combined. An efficient solution degrades in other objective functions if one of the functions is improved any further. We can scalarize the solution of a multi-objective problem by adding weights or introducing epsilon constraints. In all cases, we strive to traverse through the Pareto points (non-dominating points).

V. Game Theory

Game theory helps us in obtaining a mathematical behaviour of a system in game based manner. It generally involves modeling systems into players who participate in a game (target function) equipped with different strategies to attain Nash equilibrium where no players condition can be improved further.Each player will be given a payoff as part of the game. It can be classified into cooperative and non - cooperative games. The former has players cooperating with each other to attain equilibrium. The latter together reaches an equilibrium where everyone are benefited.

A. Jargon in Game theory [1]

- Game : Game is a model of an interactive state between entities or groups of entities.
- Player : Players are the basic entities of the games. This entity is the decision maker of the game that can be a person, a group, a concept, and so on.
- State : States are the possible situations of the game that the players can be in.
- Action : The set of all possible works that players of the game can do in different states is called actions.
- Payoff : The score that awarded to the action of a player in a game (or in one step of the game) is called payoff which can be positive or negative.
- Strategy : A player's strategy is the complete set of actions that player can do in each state of the game. Each player has a number of strategies that can be selected based on the conditions and the objective(s).
- Equilibrium : Equilibrium is the point of a game at which no player tends to change, and any change leads to worsening the payoffs of all players.

B. Nash Equilibrium

The equilibrium point is the point at which none of the game's players tend to alter. The Nash equilibrium is a game action profile wherein, assuming consistent actions by other players, any modification worsens the condition of each player. In other words, it's an action profile that doesn't encourage players to change their circumstances while presuming that other players' actions are consistent . The Nash equilibrium idea is used to examine the outcomes of numerous decision makers' strategic interactions.

C. Game theory and DE procedure

Multi-objective optimisation is modelled as cooperative game where each objective function is considered as a player. The optimal solution of each objectives function is found separately and a pay-off matrix is formed at each of these solutions.

$$P_0 = \begin{pmatrix} f_1(x_1) & f_2(x_1) & .... & f_i(x_1) & .... & f_m(x_1) \\ f_1(x_2) & f_2(x_2) & .... & f_i(x_2) & .... & f_m(x_2) \\ & & .... & .... & & \\ & & .... & .... & & \\ f_1(x_m) & f_2(x_m) & .... & f_i(x_m) & .... & f_m(x_m) \end{pmatrix} \tag{3}$$

From this matrix the maximum and minimum values are found and a substitute equation is formed.

$$S = \prod_{i=1}^{m} \frac{[f_{i,max} - f_i(x)]}{[f_{i,max} - f_{i,min}]} \tag{4}$$

Maximising the substitute equation gives us a compromise solution which has values of the objective function not more than the maximum in the payoff matrix and at best the optimal solution.

$$\text{Maximize } S \text{ , subject to } g(x) \leq 0$$

The algorithm has two operators Niche and Pareto set filter. **Niche** forces individuals to share available resources and maintain appropriate diversity.Because the non-dominated region –Pareto optimal set — is the optimum goal of a Pareto DE, effective niche technique is critical to its success.

Cavicchio's notion (Cavicchio, 1972) is used to create a specialty technique. The parents are either replaced or passed on to the following generation after producing a pair of new children. Replacement occurs only when one offspring's rank in the parent population is no lower than their parents' best rank. If not, the parents pass on to the next generation.

The **Pareto-set filter** lowers the impacts of genetic drift and improves the robustness of a Pareto DE. The best features of the parents are not necessarily inherited by their offspring when a new generation is reproduced. Because of the small population size, some of these features may never arise in a subsequent phase of evolution. Many points arise once or twice in evolutionary processes before disappearing forever. Some of them could be the sought-after optimum goals, or Pareto optimum points. A new idea known as the Pareto-set filter is developed to prevent the loss of Pareto optimal points. At each generation, a Pareto-set filter pools non-dominated points ranked 1 and drops dominated points. Points ranked 1 are filtered at the end of each generation.

At each iteration the maximum and minimum values of the Pareto set in the filter are obtained and added in 3. If the 4 remains same we assume convergence and stop the process.

D. Game theory and GA procedure

A vector (multi-objective) optimization issue is reduced to a scalar optimization problem using this method, which uses the game theory methodology to transform the problem. A basic GA is used to optimise the substitution problem once it has been formed. There are four steps to the method:

- optimize each objective function using GA separately
- construct the pay-off matrix (3)
- calculate the best and worst values of each objective function , and construct the substitution function S
- apply GA to optimise the scalar objective optimization problem

m+1 scalar optimization analyses are required by this algorithm. As previously stated, the offered optimum position from the fourth phase is a sensible choice. As previously stated, the supplied optimum point from the fourth step is a sensible compromise that falls within the Pareto optimal set.

VI. Benchmark problems used for DE and GA

Two benchmark problems were were used to test the algorithm.

Problem 1:

$$Minimize \begin{cases} f_1(x) = \begin{cases} -x & x \leq 1 \\ x-2 & 1 < x \leq 3 \\ 4-x & 3 < x \leq 4 \\ x-4 & x > 4 \end{cases} \\ f_2(x) = (x-5)^2 \end{cases} \tag{5}$$

Constraints:

$$5 \leq x \leq 10 \tag{6}$$

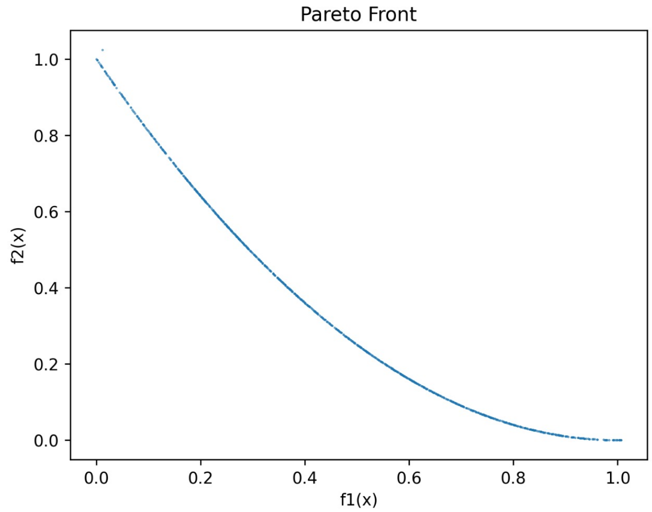

Fig. 1. Function 1

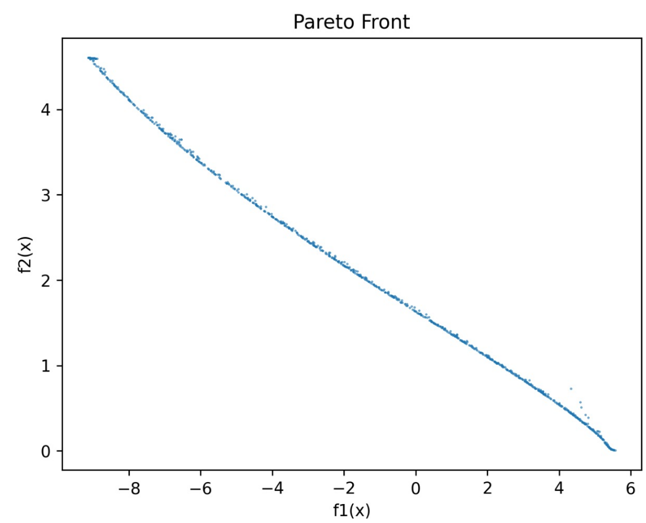

Fig. 2. Function 2

Problem 2:

$$Minimize = \begin{cases} f_1(x,y) = [1 + (A_1 - B_1(x,y))^2 + (A_2 - B_2(x,y))^2] \\ f_2(x,y) = (x+3)^2 + (y+1)^2 \end{cases} \tag{7}$$

$$where = \begin{cases} A_1 = 0.5sin(1) - 2cos(1) + sin(2) - 1.5cos(2) \\ A_2 = 1.5sin(1) - cos(1) + 2sin(2) - 0.5cos(2) \\ B_1(x,y) = 0.5sin(x) - 2cos(x) + sin(y) - 1.5cos(y) \\ B_2 = 1.5sin(x) - cos(x) + 2sin(y) - 0.5cos(y) \end{cases} \tag{8}$$

Constraints:

$$-\pi \leq x, \pi \geq y \tag{9}$$

VII. Result

We were able to obtain a payoff-matrix from which we derived a single solution for GA and a Pareto front for DE.

```
[[-0.9996979726410249, 15.997583872348725],
 [0.9997511810564053, 6.191086669160334e-08
]]
(array([-4.99998783]), 10.50537221953014, 0
)
```

Fig. 3. Ga payoff matrix and result

References

[1] FRANKLIN Y. CHENG and DAN LI , "GENETIC ALGORITHM AND GAME THEORY FOR MULTIOBJECTIVE OPTIMIZATION OF SEISMIC STRUCTURES WITH/WITHOUT CONTROL " ,Eleventh World Conference on Earthquake Engineering.

[2] Mohammad Karim Sohrabi, Hossein Azgomi, A Survey on the Combined Use of Optimization Methods and Game Theory,2018,22.

[3] Zhehuang Huang, Yidong Chen, "An Improved Differential Evolution Algorithm Based on Adaptive Parameter", Journal of Control Science and Engineering, vol. 2013, Article ID 462706, 5 pages, 2013.

[4] Georgioudakis Manolis and Plevris Vagelis,A Comparative Study of Differential Evolution Variants in Constrained Structural Optimization, Frontiers in Built Environment,6,2020.