Constraint-Handling in EAs

Motivation

- Traditional mathematical programming techniques used to solve constrained optimization problems have several limitations when dealing with the general nonlinear programming problem:

$$\text{Min } f(x)$$

subject to:

$$g_i(x) \leq 0, \; i = 1, \ldots, m$$

$$h_j(x) = 0, \; j = 1, \ldots, p$$

Where x is the vector of decision variables

$$x = [x_1, x_2, \ldots, x_n]^T,$$

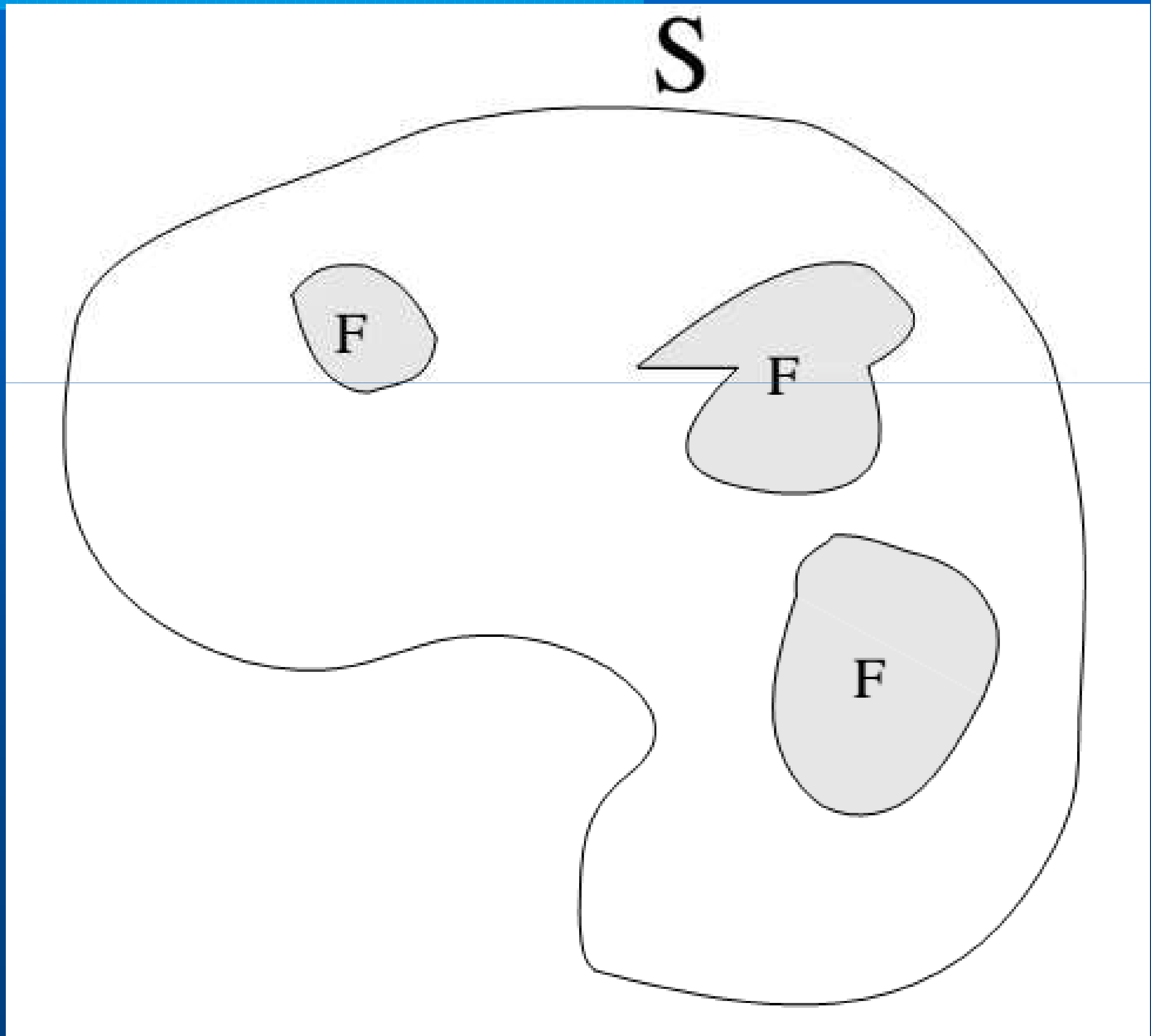
m is the number of inequality constraints and

p is the number of equality constraints (in both cases, constraints can be either linear or non-linear

Motivation

- Evolutionary Algorithms (EAs) have been found successful in the solution of a wide variety of optimization problems.
- EAs are unconstrained search techniques.
- There is a considerable amount of research regarding mechanisms that allow EAs to deal with equality and inequality constraints;
- Both type of constraints can be linear or nonlinear.

Search Space



Constraint-Handling Approaches

- Penalty Functions
- Special representations and operators

- Separation of constraints and objectives
- Hybrid Methods

Penalty Functions

- The most common approach in the EA community to handle constraints (particularly, inequality constraints) is to use penalties.
- Penalty functions were originally proposed by Richard Courant in the 1940s
- The idea of penalty functions is to transform a constrained optimization problem into an unconstrained one by adding (or subtracting) a certain value to/from the objective function based on the amount of constraint violation present in a certain solution.

Penalty Function

- In mathematical programming, two kinds of penalty functions are considered: exterior and interior.
- Exterior methods: We start with an infeasible solution and from there we move towards the feasible region.
- Interior methods: The penalty term is chosen such that its value will be small at points away from the constraint boundaries
- If we start from a feasible point, the subsequent points generated will always lie within the feasible region since the constraint boundaries act as barriers during the optimization process.

Penalty Function

- Penalty functions can deal both with equality and inequality constraints, and the normal approach is to transform an equality to an inequality of the form:

$$|h_j(\vec{x})| - \epsilon \leq 0$$

where ϵ is the tolerance allowed (a very small value).

Special representations and operators

- Some researchers have decided to develop special representation schemes to tackle certain difficult problem for which a generic representation scheme (e.g., the binary representation used in the traditional genetic algorithm) might not be appropriate.
- Due to the change of representation, it is necessary to design special genetic operators that work in a similar way than the traditional operators used with a binary representation.

Special representations and operators

- The emphasis of these approaches is to map chromosomes from the infeasible region into the feasible region of the problem to solve.
- In some cases, special operators have also been designed in order to produce offspring that lie on the boundary between the feasible and the infeasible region.
- A more intriguing idea is to transform the whole feasible region into a different shape that is easier to explore.

Separation of constraints and objectives

- Unlike penalty functions which combine the value of the objective function and the constraints of a problem to assign fitness, these approaches handle constraints and objectives separately.
- In multiobjective optimization concepts: The main idea is to redefine the single-objective optimization of f (x) as a multiobjective optimization problem in which we will have m + 1 objectives, where m is the total number of constraints.
- Some examples:
  - Coevolution
  - Superiority of feasible points
  - Behavioral memory

Hybrid Method

- Within this category, we consider methods that are coupled with another technique (either another heuristic or a mathematical programming approach).
- Examples:
  - Simulated Annealing (SA): Wah & Chen [2001] proposed a hybrid of SA and a genetic algorithm (GA). The first part of the search is guided by SA. After that, the best solution is refined using a GA. To deal with constraints, Wah & Chen use Lagrangian Multipliers.

Hybrid Method

- Artificial Immune System (AIS): Hajela and Lee [1996] proposed a GA hybridized with an AIS (based on the negative selection approach). The idea is to adopt as antigens some feasible solutions and evolve (in an inner GA) the antibodies (i.e., the infeasible solutions) so that they are "similar" (at a genotypic level) to the antigens.
- Ant System (AS): The main AS algorithm is a multi-agent system where low level interactions between single agents (i.e., artificial ants) result in a complex behavior of the whole ant colony. Although mainly used for combinatorial optimization, AS has also been successfully applied to numerical optimization [Bilchev & Parmee, 1995; Leguizamon, 2004].

Hybrid Method

- Cultural Algorithms: In this sort of approach, the main idea is to preserve beliefs that are socially accepted and discard (or prune) unacceptable beliefs. The acceptable beliefs can be seen as constraints that direct the population at the micro-evolutionary level. Therefore, constraints can influence directly the search process, leading to an efficient optimization process.
- Constrained optimization by random evolution (CORE): This is an approach proposed by Belur [1997] which combines a random evolution search with Nelder and Mead's method [1965].

Hybrid Method

- Adeli and Cheng [1994] proposed a hybrid EA that integrates the penalty function method with the primal-dual method. This approach is based on sequential minimization of the Lagrangian method
- Kim and Myung [1997] proposed the use of an evolutionary optimization method combined with an augmented Lagrangian function that guarantees the generation of feasible solutions during the search process.