Constraint-Handling in Nature-Inspired Optimization

Efrén Mezura-Montes

Artificial Intelligence Research Center
University of Veracruz, MEXICO
emezura@uv.mx
http://www.uv.mx/personal/emezura

Universidad Veracruzana

2017 IEEE Congress on Evolutionary Computation
Donostia - San Sebastián, SPAIN, June 5th, 2017

Outline

Universidad Veracruzana

Outline

Universidad Veracruzana

Outline

Universidad Veracruzana

Outline

Universidad Veracruzana

Outline

Universidad Veracruzana

Problem statement

Constrained numerical optimization problem (CNOP)

Find $\vec{x}$ which minimizes $f(\vec{x})$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, m$$
$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p$$

- $\vec{x} \in \mathbf{R}^n$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$.
- Each $x_k, k = 1, ..., n$ is bounded by lower and upper limits $L_k \leq x_k \leq U_k$ which define the search space $\mathcal{S}$.
- $\mathcal{F}$ comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region.
- To handle equality constraints they are transformed into inequality constraints as follows: $|h_j(\vec{x})| - \varepsilon \leq 0$.

Problem statement

Constrained numerical optimization problem (CNOP)

Find $\vec{x}$ which minimizes $f(\vec{x})$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, m$$
$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p$$

- $\vec{x} \in \mathbf{R}^n$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$.
- Each $x_k, k = 1, ..., n$ is bounded by lower and upper limits $L_k \leq x_k \leq U_k$ which define the search space $\mathcal{S}$.
- $\mathcal{F}$ comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region.
- To handle equality constraints they are transformed into inequality constraints as follows: $|h_j(\vec{x})| - \varepsilon \leq 0$.

Problem statement

Constrained numerical optimization problem (CNOP)

Find $\vec{x}$ which minimizes $f(\vec{x})$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, m$$
$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p$$

- $\vec{x} \in \mathbf{R}^n$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$.
- Each $x_k, k = 1, ..., n$ is bounded by lower and upper limits $L_k \leq x_k \leq U_k$ which define the search space $\mathcal{S}$.
- $\mathcal{F}$ comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region.
- To handle equality constraints they are transformed into inequality constraints as follows: $\left|h_j(\vec{x})\right| - \varepsilon \leq 0$.

Problem statement

Constrained numerical optimization problem (CNOP)

Find $\vec{x}$ which minimizes $f(\vec{x})$

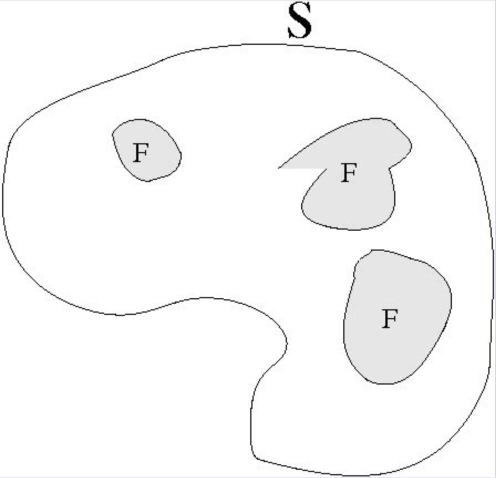
subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, m$$
$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p$$

- $\vec{x} \in \mathbf{R}^n$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$.
- Each $x_k, k = 1, ..., n$ is bounded by lower and upper limits $L_k \leq x_k \leq U_k$ which define the search space $\mathcal{S}$.
- $\mathcal{F}$ comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region.
- To handle equality constraints they are transformed into inequality constraints as follows: $\left|h_j(\vec{x})\right| - \varepsilon \leq 0$.

Constrained numerical optimization problem

Constrained search space



Outline

Feasible global optimum

In the following definitions we will assume minimization (without loss of generality). $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ refers to the feasible optimum point and its corresponding value of the objective function $f(\vec{x}^*)$ is called the feasible optimum value. The pair $\vec{x}^*$ and $f(\vec{x}^*)$ is called feasible optimum solution.

Feasible global minimum

A function $f(\vec{x})$ defined on a set S attains its feasible global minimum at a point $\vec{x}^* \in \mathcal{F} \subseteq S$ if and only if: $f(\vec{x}^*) \leq f(\vec{x}), \forall \vec{x} \in \mathcal{F} \subseteq S$.

Kuhn-Tucker Conditions

- Kuhn and Tucker developed the necessary and sufficient optimality conditions for the CNOP assuming that the functions f, g_i, and h_j, are differentiable or twice-differentiable.
- These optimality conditions, commonly known as the *Kuhn-Tucker conditions* (KTC) consist of finding a solution to a system of nonlinear equations.
- However, it is quite difficult that KTC hold for real-world problems. Therefore, the CNOP is an open-problem.

Universidad Veracruzana

Kuhn-Tucker Conditions

- Kuhn and Tucker developed the necessary and sufficient optimality conditions for the CNOP assuming that the functions f, g_i, and h_j, are differentiable or twice-differentiable.
- These optimality conditions, commonly known as the *Kuhn-Tucker conditions* (KTC) consist of finding a solution to a system of nonlinear equations.
- However, it is quite difficult that KTC hold for real-world problems. Therefore, the CNOP is an open-problem.

Universidad Veracruzana

Kuhn-Tucker Conditions

- Kuhn and Tucker developed the necessary and sufficient optimality conditions for the CNOP assuming that the functions f, g_i, and h_j, are differentiable or twice-differentiable.
- These optimality conditions, commonly known as the *Kuhn-Tucker conditions* (KTC) consist of finding a solution to a system of nonlinear equations.
- However, it is quite difficult that KTC hold for real-world problems. Therefore, the CNOP is an open-problem.

Outline

Two categories

- They usually require just one solution which is improved during the process.
- Two categories:
  - Direct Methods.
  - Indirect Methods.

Two categories

- They usually require just one solution which is improved during the process.
- Two categories:
  - Direct Methods.
  - Indirect Methods.

Universidad Veracruzana

Direct methods

These methods use only the information of the objective function to find search directions.

Indirect methods

These methods require that the objective function is differentiable or twice differentiable so as to use such information to guide the search.

Outline

Universidad Veracruzana

Motivation

Despite the large number of mathematical programming methods developed, several optimization problems present characteristics that make them difficult to solve using this kind of algorithms.

Difficulties found

- Problems with non-differentiable objective functions and/or non-differentiable constraints.
- Problems with disjoint feasible regions
- Problems with objective function and/or constraints not available in algebraic form.
- Problems in which the Kuhn-Tucker conditions for optimality do not hold.
- Problems where no mathematical programming technique can guarantee convergence to the global optimum.
- Huge search spaces.

Difficulties found

- Problems with non-differentiable objective functions and/or non-differentiable constraints.
- Problems with disjoint feasible regions
- Problems with objective function and/or constraints not available in algebraic form.
- Problems in which the Kuhn-Tucker conditions for optimality do not hold.
- Problems where no mathematical programming technique can guarantee convergence to the global optimum.
- Huge search spaces.

Difficulties found

- Problems with non-differentiable objective functions and/or non-differentiable constraints.
- Problems with disjoint feasible regions
- Problems with objective function and/or constraints not available in algebraic form.
- Problems in which the Kuhn-Tucker conditions for optimality do not hold.
- Problems where no mathematical programming technique can guarantee convergence to the global optimum.
- Huge search spaces.

Universidad Veracruzana

Difficulties found

- Problems with non-differentiable objective functions and/or non-differentiable constraints.
- Problems with disjoint feasible regions
- Problems with objective function and/or constraints not available in algebraic form.
- Problems in which the Kuhn-Tucker conditions for optimality do not hold.
- Problems where no mathematical programming technique can guarantee convergence to the global optimum.
- Huge search spaces.

Difficulties found

- Problems with non-differentiable objective functions and/or non-differentiable constraints.
- Problems with disjoint feasible regions
- Problems with objective function and/or constraints not available in algebraic form.
- Problems in which the Kuhn-Tucker conditions for optimality do not hold.
- Problems where no mathematical programming technique can guarantee convergence to the global optimum.
- Huge search spaces.

Universidad Veracruzana

Difficulties found

- Problems with non-differentiable objective functions and/or non-differentiable constraints.
- Problems with disjoint feasible regions
- Problems with objective function and/or constraints not available in algebraic form.
- Problems in which the Kuhn-Tucker conditions for optimality do not hold.
- Problems where no mathematical programming technique can guarantee convergence to the global optimum.
- Huge search spaces.

Nature-inspired algorithms (NIAs)

- Evolutionary algorithms (EAs) and swarm intelligence algorithms (SIAs) (grouped as NIAs) are popular meta-heuristics approaches used to solve complex optimization problems.
- NIAs are designed to deal with unconstrained search spaces.
- The design and addition of a constraint-handling techniques into a NIA to deal with a constrained search space is an open problem.

Universidad Veracruzana

Nature-inspired algorithms (NIAs)

- Evolutionary algorithms (EAs) and swarm intelligence algorithms (SIAs) (grouped as NIAs) are popular meta-heuristics approaches used to solve complex optimization problems.
- NIAs are designed to deal with unconstrained search spaces.
- The design and addition of a constraint-handling techniques into a NIA to deal with a constrained search space is an open problem.

Universidad Veracruzana

Nature-inspired algorithms (NIAs)

- Evolutionary algorithms (EAs) and swarm intelligence algorithms (SIAs) (grouped as NIAs) are popular meta-heuristics approaches used to solve complex optimization problems.
- NIAs are designed to deal with unconstrained search spaces.
- The design and addition of a constraint-handling techniques into a NIA to deal with a constrained search space is an open problem.

Universidad Veracruzana

Main components of a nature-inspired algorithm

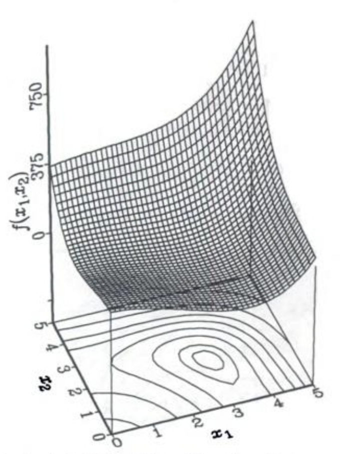
1. Solution encoding.
2. Fitness function.
3. Initial population.
4. Parent selection.
5. Variation operators (crossover & mutation).
6. Replacement.

Universidad Veracruzana

Why the search must change?

Unconstrained optimization problem

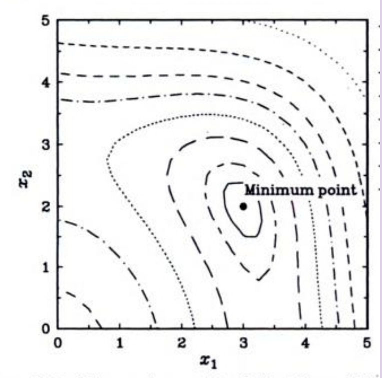
Min: $f(\vec{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2)^2$



Taken from Deb, K., Opt. for Eng. Design, Algorithms and Examples, Prentice-Hall, 1995.

Why the search must change?

Unconstrained optimization problem

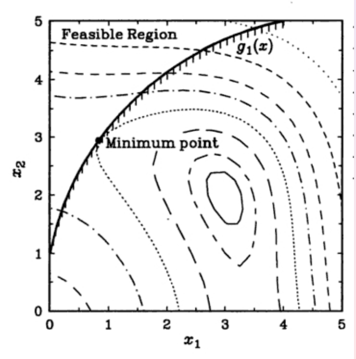


Taken from Deb, K., Opt. for Eng. Design, Algorithms and Examples, Prentice-Hall, 1995.

Why the search must change?

Constrained optimization problem

Min: $f(\vec{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2)^2$
subject to:
$(x_1 - 5)^2 + x_2^2 - 26 \geq 0$



Taken from Deb, K., Opt. for Eng. Design, Algorithms and Examples, Prentice-Hall, 1995.

Why a constraint-handling technique?

- The initial population (usually generated at random) may contain several (if not all) infeasible solutions, and it may be difficult to generate only feasible solutions from the beginning.
- The information about feasibility must be incorporated into the fitness function to bias the search to the feasible region.
- The parent selection and/or replacement must distinguish between feasible and infeasible solutions.
- The variation operators are blind with respect to the constraints of the optimization problem.

Universidad Veracruzana

Why a constraint-handling technique?

- The initial population (usually generated at random) may contain several (if not all) infeasible solutions, and it may be difficult to generate only feasible solutions from the beginning.
- The information about feasibility must be incorporated into the fitness function to bias the search to the feasible region.
- The parent selection and/or replacement must distinguish between feasible and infeasible solutions.
- The variation operators are blind with respect to the constraints of the optimization problem.

Universidad Veracruzana

Why a constraint-handling technique?

- The initial population (usually generated at random) may contain several (if not all) infeasible solutions, and it may be difficult to generate only feasible solutions from the beginning.
- The information about feasibility must be incorporated into the fitness function to bias the search to the feasible region.
- The parent selection and/or replacement must distinguish between feasible and infeasible solutions.
- The variation operators are blind with respect to the constraints of the optimization problem.

Universidad Veracruzana

Why a constraint-handling technique?

- The initial population (usually generated at random) may contain several (if not all) infeasible solutions, and it may be difficult to generate only feasible solutions from the beginning.
- The information about feasibility must be incorporated into the fitness function to bias the search to the feasible region.
- The parent selection and/or replacement must distinguish between feasible and infeasible solutions.
- The variation operators are blind with respect to the constraints of the optimization problem.

Universidad Veracruzana

Constraint-handling over the years

- Two classifications were proposed: one by Michalewicz and Schoenauer [96] and another one by Coello [18].
- Both taxonomies agreed on penalty functions as a particular class.
- This new classification for earlier methods is based on constraint-handling mechanisms, whereas the search algorithm employed is discussed as a separate issue.

Universidad Veracruzana

Constraint-handling over the years

- Two classifications were proposed: one by Michalewicz and Schoenauer [96] and another one by Coello [18].
- Both taxonomies agreed on penalty functions as a particular class.
- This new classification for earlier methods is based on constraint-handling mechanisms, whereas the search algorithm employed is discussed as a separate issue.

Universidad Veracruzana

Constraint-handling over the years

- Two classifications were proposed: one by Michalewicz and Schoenauer [96] and another one by Coello [18].
- Both taxonomies agreed on penalty functions as a particular class.
- This new classification for earlier methods is based on constraint-handling mechanisms, whereas the search algorithm employed is discussed as a separate issue.

Outline

Universidad Veracruzana

Penalty functions

Definition

Based on mathematical programming approaches, where a CNOP is transformed into an unconstrained numerical optimization problem, NIAs have adopted penalty functions, whose general formula is the following:

$$\phi(\vec{x}) = f(\vec{x}) + p(\vec{x})$$

where $\phi(\vec{x})$ is the expanded objective function to be optimized, and $p(\vec{x})$ is the penalty value that can be calculated as follows:

$$p(\vec{x}) = \sum_{i=1}^{m} r_i \cdot \max(0, g_i(\vec{x}))^2 + \sum_{j=1}^{p} c_j \cdot |h_j(\vec{x})|$$

where r_i and c_j are positive constants called “penalty factors”.

Penalty functions

Pros and cons

- The aim is to decrease the fitness of infeasible solutions.
- Unlike mathematical programming approaches, where interior and exterior penalty functions are employed, NIAs have mainly focused on the last ones.
- Their implementation is quite simple ... but,
- Penalty functions require a careful fine-tuning of their penalty factors.
- Such values usually are highly problem-dependent.
- Different approaches have been proposed to tackle this shortcoming.

Universidad Veracruzana

Penalty functions

Pros and cons

- The aim is to decrease the fitness of infeasible solutions.
- Unlike mathematical programming approaches, where interior and exterior penalty functions are employed, NIAs have mainly focused on the last ones.
- Their implementation is quite simple ... but,
- Penalty functions require a careful fine-tuning of their penalty factors.
- Such values usually are highly problem-dependent.
- Different approaches have been proposed to tackle this shortcoming.

Universidad Veracruzana

Penalty functions

Pros and cons

- The aim is to decrease the fitness of infeasible solutions.
- Unlike mathematical programming approaches, where interior and exterior penalty functions are employed, NIAs have mainly focused on the last ones.
- Their implementation is quite simple ... but,
- Penalty functions require a careful fine-tuning of their penalty factors.
- Such values usually are highly problem-dependent.
- Different approaches have been proposed to tackle this shortcoming.

Penalty functions

Pros and cons

- The aim is to decrease the fitness of infeasible solutions.
- Unlike mathematical programming approaches, where interior and exterior penalty functions are employed, NIAs have mainly focused on the last ones.
- Their implementation is quite simple ... but,
- Penalty functions require a careful fine-tuning of their penalty factors.
- Such values usually are highly problem-dependent.
- Different approaches have been proposed to tackle this shortcoming.

Penalty functions

Pros and cons

- The aim is to decrease the fitness of infeasible solutions.
- Unlike mathematical programming approaches, where interior and exterior penalty functions are employed, NIAs have mainly focused on the last ones.
- Their implementation is quite simple ... but,
- Penalty functions require a careful fine-tuning of their penalty factors.
- Such values usually are highly problem-dependent.
- Different approaches have been proposed to tackle this shortcoming.

Universidad Veracruzana

Penalty functions

Pros and cons

- The aim is to decrease the fitness of infeasible solutions.
- Unlike mathematical programming approaches, where interior and exterior penalty functions are employed, NIAs have mainly focused on the last ones.
- Their implementation is quite simple ... but,
- Penalty functions require a careful fine-tuning of their penalty factors.
- Such values usually are highly problem-dependent.
- Different approaches have been proposed to tackle this shortcoming.

Universidad Veracruzana

Penalty functions

Death penalty

- The most simple penalty function.
- Infeasible solutions are assigned the worst possible fitness value or are simply eliminated from the optimization process.
- Keeps the search from using valuable information from infeasible solutions.
- Not suitable for very small feasible region with respect to the whole search space.

Universidad Veracruzana

Penalty functions

Death penalty

- The most simple penalty function.
- Infeasible solutions are assigned the worst possible fitness value or are simply eliminated from the optimization process.
- Keeps the search from using valuable information from infeasible solutions.
- Not suitable for very small feasible region with respect to the whole search space.

Universidad Veracruzana

Penalty functions

Death penalty

- The most simple penalty function.
- Infeasible solutions are assigned the worst possible fitness value or are simply eliminated from the optimization process.
- Keeps the search from using valuable information from infeasible solutions.
- Not suitable for very small feasible region with respect to the whole search space.

Universidad Veracruzana

Death penalty

- The most simple penalty function.
- Infeasible solutions are assigned the worst possible fitness value or are simply eliminated from the optimization process.
- Keeps the search from using valuable information from infeasible solutions.
- Not suitable for very small feasible region with respect to the whole search space.

Penalty functions

Static penalty functions

- Those whose penalty factor values (r_i and c_j, $i = 1, \ldots, m$ and $j = 1 \ldots, m$) remain fixed during all the process.
  - Kuri and Villegas-Quezada [59].
  - Homaifar et al. [43]. Hoffmeister and Sprave [42].
  - Le Riche et al. [112]).
- The main drawback is the generalization of such type of approach, i.e., the values that may be suitable for one problem are normally unsuitable for another one.

Universidad Veracruzana

Penalty functions

Static penalty functions

- Those whose penalty factor values (r_i and c_j, $i = 1, \ldots, m$ and $j = 1 \ldots, m$) remain fixed during all the process.
    - Kuri and Villegas-Quezada [59].
    - Homaifar et al. [43]. Hoffmeister and Sprave [42].
    - Le Riche et al. [112]).
- The main drawback is the generalization of such type of approach, i.e., the values that may be suitable for one problem are normally unsuitable for another one.

Penalty functions

Dynamic penalty functions

- Time (usually the generation counter in a NIA) is used to affect the penalty factors.
- Considering the usage of exterior penalty functions, soft penalties are expected first, while severe penalties are adopted in the last part of the search.
- Examples:
  - Joines and Houck [48].
  - Kazarlis and Petridis[51].
  - Crossley and Williams [21].
- The cooling factor of the simulated annealing algorithm has been employed to vary the penalty factors by Michalewicz and Attia [94].
- The main disadvantages of dynamic penalty functions are the parameters for their dynamic tuning and the difficulty to generalize them.

Penalty functions

Dynamic penalty functions

- Time (usually the generation counter in a NIA) is used to affect the penalty factors.
- Considering the usage of exterior penalty functions, soft penalties are expected first, while severe penalties are adopted in the last part of the search.
- Examples:
  - Joines and Houck [48].
  - Kazarlis and Petridis[51].
  - Crossley and Williams [21].
- The cooling factor of the simulated annealing algorithm has been employed to vary the penalty factors by Michalewicz and Attia [94].
- The main disadvantages of dynamic penalty functions are the parameters for their dynamic tuning and the difficulty to generalize them.

Penalty functions

Dynamic penalty functions

- Time (usually the generation counter in a NIA) is used to affect the penalty factors.
- Considering the usage of exterior penalty functions, soft penalties are expected first, while severe penalties are adopted in the last part of the search.
- Examples:
  - Joines and Houck [48].
  - Kazarlis and Petridis[51].
  - Crossley and Williams [21].
- The cooling factor of the simulated annealing algorithm has been employed to vary the penalty factors by Michalewicz and Attia [94].
- The main disadvantages of dynamic penalty functions are the parameters for their dynamic tuning and the difficulty to generalize them.

Dynamic penalty functions

- Time (usually the generation counter in a NIA) is used to affect the penalty factors.
- Considering the usage of exterior penalty functions, soft penalties are expected first, while severe penalties are adopted in the last part of the search.
- Examples:
  - Joines and Houck [48].
  - Kazarlis and Petridis[51].
  - Crossley and Williams [21].
- The cooling factor of the simulated annealing algorithm has been employed to vary the penalty factors by Michalewicz and Attia [94].
- The main disadvantages of dynamic penalty functions are the parameters for their dynamic tuning and the difficulty to generalize them.

Dynamic penalty functions

- Time (usually the generation counter in a NIA) is used to affect the penalty factors.
- Considering the usage of exterior penalty functions, soft penalties are expected first, while severe penalties are adopted in the last part of the search.
- Examples:
  - Joines and Houck [48].
  - Kazarlis and Petridis[51].
  - Crossley and Williams [21].
- The cooling factor of the simulated annealing algorithm has been employed to vary the penalty factors by Michalewicz and Attia [94].
- The main disadvantages of dynamic penalty functions are the parameters for their dynamic tuning and the difficulty to generalize them.

Penalty functions

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Penalty functions

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Penalty functions

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Penalty functions

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Penalty functions

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Penalty functions

Adaptive penalty functions

- The behavior of the NIA is used to update the penalty factors.
- Feasibility of the best solution in a number of generations by Hadj-Alouane and Bean [34].
- The fitness of the best feasible solution by Rasheed [105].
- The balance between feasible and infeasible solutions by Hamda and Schoenauer [35] and Hamida and Schoenauer [36].
- The average of the objective function and the level of violation of each constraint by Barbosa and Lemonge [11].
- Co-evolution by Coello [20].
- Fuzzy logic by Wu and Yu [150].
- Their main drawback lies in the following: there is no guarantee that the values defined based on the current behavior will be indeed useful later.

Penalty functions

Discussion

- Diverse ways to define penalty factors (static, dynamic, adaptive, co-evolved, fuzzy-adapted, etc.).
- Not clear which approach was more competitive.
- Most of the time, additional parameters were required.

Penalty functions

Discussion

- Diverse ways to define penalty factors (static, dynamic, adaptive, co-evolved, fuzzy-adapted, etc.).
- Not clear which approach was more competitive.
- Most of the time, additional parameters were required.

Universidad Veracruzana

Penalty functions

Discussion

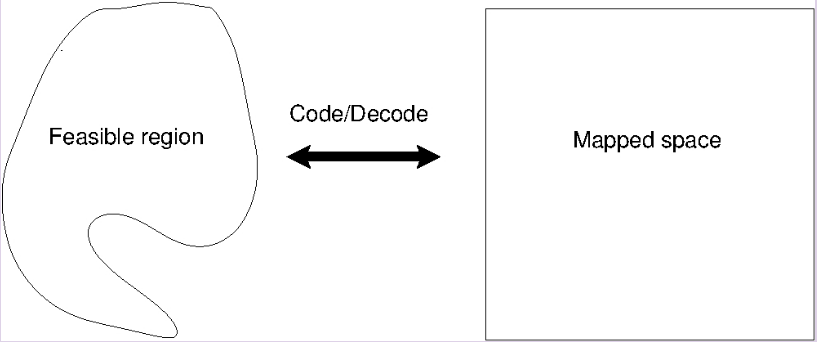
- Diverse ways to define penalty factors (static, dynamic, adaptive, co-evolved, fuzzy-adapted, etc.).
- Not clear which approach was more competitive.
- Most of the time, additional parameters were required.

Outline

Universidad Veracruzana

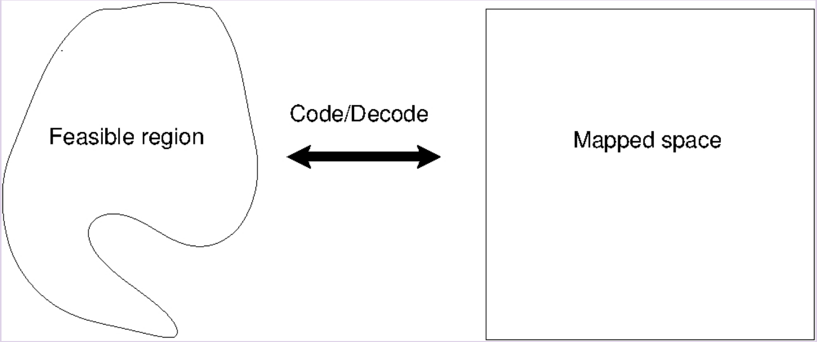
Decoders

- One of the most competitive constraint-handling techniques in the early years.
- They are based on the idea of mapping the feasible region $\mathcal{F}$ of the search space $\mathcal{S}$ onto an easier-to-sample space where a NIA can provide a better performance [56].



Decoders

- One of the most competitive constraint-handling techniques in the early years.
- They are based on the idea of mapping the feasible region $\mathcal{F}$ of the search space $\mathcal{S}$ onto an easier-to-sample space where a NIA can provide a better performance [56].



Decoders

- The mapping process must guarantee that each feasible solution in the search space is included in the decoded space and that a decoded solution corresponds to a feasible solution in the search space.
- The transformation process must be fast and it is highly desirable that small changes in the search space of the original problem cause small changes in the decoded space as well.
  - Homomorphous maps: the feasible region is mapped into an n-dimensional cube, by Koziel and Michalewicz [56, 57].
  - Riemann mappings by Kim and Husbands [52, 53, 54].

Universidad Veracruzana

Decoders

- The mapping process must guarantee that each feasible solution in the search space is included in the decoded space and that a decoded solution corresponds to a feasible solution in the search space.
- The transformation process must be fast and it is highly desirable that small changes in the search space of the original problem cause small changes in the decoded space as well.
  - Homomorphous maps: the feasible region is mapped into an *n*-dimensional cube, by Koziel and Michalewicz [56, 57].
  - Riemann mappings by Kim and Husbands [52, 53, 54].

Decoders

Discussion

- Their actual implementation is far from trivial.
- They may involve a computational cost.
- Decoders are rarely used nowadays.

Decoders

Discussion

- Their actual implementation is far from trivial.
- They may involve a computational cost.
- Decoders are rarely used nowadays.

Decoders

Discussion

- Their actual implementation is far from trivial.
- They may involve a computational cost.
- Decoders are rarely used nowadays.

Outline

Universidad Veracruzana

Special operators

- A special operator is conceived as a way of either preserving the feasibility of a solution or moving within a specific region of interest within the search space.
- A variation operator which constructs linear combinations of feasible solutions to preserve their feasibility (GENOCOP) by Michalewicz [93].
- Special operators designed to convert solutions which only satisfy linear constraints into fully feasible solutions (GENOCOP III) by Michalewicz and Nazhiyath [95].
- Special operators to assign values to the decision variables aiming to keep the feasibility of the solution by Kowalczyk [55].
- Special operators for two specific problems to sample the boundaries of their feasible regions by Schoenauer and Michalewicz [116, 117].

Special operators

- A special operator is conceived as a way of either preserving the feasibility of a solution or moving within a specific region of interest within the search space.
- A variation operator which constructs linear combinations of feasible solutions to preserve their feasibility (GENOCOP) by Michalewicz [93].
- Special operators designed to convert solutions which only satisfy linear constraints into fully feasible solutions (GENOCOP III) by Michalewicz and Nazhiyath [95].
- Special operators to assign values to the decision variables aiming to keep the feasibility of the solution by Kowalczyk [55].
- Special operators for two specific problems to sample the boundaries of their feasible regions by Schoenauer and Michalewicz [116, 117].

Special operators

- A special operator is conceived as a way of either preserving the feasibility of a solution or moving within a specific region of interest within the search space.
- A variation operator which constructs linear combinations of feasible solutions to preserve their feasibility (GENOCOP) by Michalewicz [93].
- Special operators designed to convert solutions which only satisfy linear constraints into fully feasible solutions (GENOCOP III) by Michalewicz and Nazhiyath [95].
- Special operators to assign values to the decision variables aiming to keep the feasibility of the solution by Kowalczyk [55].
- Special operators for two specific problems to sample the boundaries of their feasible regions by Schoenauer and Michalewicz [116, 117].

Special operators

- A special operator is conceived as a way of either preserving the feasibility of a solution or moving within a specific region of interest within the search space.
- A variation operator which constructs linear combinations of feasible solutions to preserve their feasibility (GENOCOP) by Michalewicz [93].
- Special operators designed to convert solutions which only satisfy linear constraints into fully feasible solutions (GENOCOP III) by Michalewicz and Nazhiyath [95].
- Special operators to assign values to the decision variables aiming to keep the feasibility of the solution by Kowalczyk [55].
- Special operators for two specific problems to sample the boundaries of their feasible regions by Schoenauer and Michalewicz [116, 117].

Special operators

- A special operator is conceived as a way of either preserving the feasibility of a solution or moving within a specific region of interest within the search space.
- A variation operator which constructs linear combinations of feasible solutions to preserve their feasibility (GENOCOP) by Michalewicz [93].
- Special operators designed to convert solutions which only satisfy linear constraints into fully feasible solutions (GENOCOP III) by Michalewicz and Nazhiyath [95].
- Special operators to assign values to the decision variables aiming to keep the feasibility of the solution by Kowalczyk [55].
- Special operators for two specific problems to sample the boundaries of their feasible regions by Schoenauer and Michalewicz [116, 117].

Special operators

Discussion

- Highly competitive results can be found when adopting special operators.
- Their main drawback is their limited applicability.
- Most of them require an ad-hoc initialization process or at least one feasible or partially-feasible solution in the initial population.

Universidad Veracruzana

Special operators

Discussion

- Highly competitive results can be found when adopting special operators.
- Their main drawback is their limited applicability.
- Most of them require an ad-hoc initialization process or at least one feasible or partially-feasible solution in the initial population.

Special operators

Discussion

- Highly competitive results can be found when adopting special operators.
- Their main drawback is their limited applicability.
- Most of them require an ad-hoc initialization process or at least one feasible or partially-feasible solution in the initial population.

Outline

Universidad Veracruzana

Separation of objective function and constraints

- Unlike combining the objective function and the values of the constraints into a single value (i.e. penalty function), there are constraint-handling techniques which work with the opposite idea.

Separation of objective function and constraints

- Powell and Skolnick in [103] proposed an approach based on the following Equation.

$$\text{fitness}(\vec{x}) = \begin{cases} f(\vec{x}) & \text{if feasible} \\ 1 + r\left(\sum_{i=1}^{m} g_i(\vec{x}) + \sum_{j=1}^{p} h_j(\vec{x})\right) & \text{otherwise} \end{cases}$$

where a feasible solution has always a better fitness value with respect to that of an infeasible solution, whose fitness is based only on their accumulated constraint violation.

Separation of objective function and constraints

- Hinterding and Michalewicz in [41] proposed the idea of dividing the search in two phases: (1) finding feasible solutions, regardless of the objective function value, and (2) after a suitable number of feasible solutions has been found, optimizing the objective function.
- Such idea was revisited by Venkatraman and Yen [138].
- Schoenauer and Xanthakis in [118] proposed a lexicographic ordering (behavioral memory) to satisfy constraints, i.e., when a certain number of solutions in the population satisfy the first constraint, an attempt to satisfy the second one is made (but the first constraint must continue to be satisfied), and so on.

Universidad Veracruzana

Separation of objective function and constraints

- Hinterding and Michalewicz in [41] proposed the idea of dividing the search in two phases: (1) finding feasible solutions, regardless of the objective function value, and (2) after a suitable number of feasible solutions has been found, optimizing the objective function.
- Such idea was revisited by Venkatraman and Yen [138].
- Schoenauer and Xanthakis in [118] proposed a lexicographic ordering (behavioral memory) to satisfy constraints, i.e., when a certain number of solutions in the population satisfy the first constraint, an attempt to satisfy the second one is made (but the first constraint must continue to be satisfied), and so on.

Separation of objective function and constraints

- Hinterding and Michalewicz in [41] proposed the idea of dividing the search in two phases: (1) finding feasible solutions, regardless of the objective function value, and (2) after a suitable number of feasible solutions has been found, optimizing the objective function.
- Such idea was revisited by Venkatraman and Yen [138].
- Schoenauer and Xanthakis in [118] proposed a lexicographic ordering (behavioral memory) to satisfy constraints, i.e., when a certain number of solutions in the population satisfy the first constraint, an attempt to satisfy the second one is made (but the first constraint must continue to be satisfied), and so on.

Separation of objective function and constraints

- Deb [25] proposed a set of three feasibility criteria as follows:
  1. When comparing two feasible solutions, the one with the best objective function is chosen.
  2. When comparing a feasible and an infeasible solution, the feasible one is chosen.
  3. When comparing two infeasible solutions, the one with the lowest sum of constraint violation is chosen.

  The sum of constraint violation can be calculated as follows:

$$\phi(\vec{x}) = \sum_{i=1}^{m} \max(0, g_i(\vec{x}))^2 + \sum_{j=1}^{p} |h_j(\vec{x})|$$

Universidad Veracruzana

Separation of objective function and constraints

- Different multi-population schemes have been proposed.
- Coello [19] divided a GA-population into sub-populations and each sub-population tried to satisfy one constraint of a CNOP and another one optimized the objective function.
- Liang and Suganthan proposed a dynamic assignment of sub-swarms to constraints in PSO [67].
- The approach was further improved in [68], where only two sub-swarms, one of them with a tolerance for inequality constraints, were used. Each particle, and not a sub-swarm, was dynamically assigned the objective function or the constraint, in such a way that more difficult objectives to optimize (satisfy) were assigned more frequently.
- Li et al. [66] adopted a similar approach but using DE as a search algorithm.

Separation of objective function and constraints

- Different multi-population schemes have been proposed.
- Coello [19] divided a GA-population into sub-populations and each sub-population tried to satisfy one constraint of a CNOP and another one optimized the objective function.
- Liang and Suganthan proposed a dynamic assignment of sub-swarms to constraints in PSO [67].
- The approach was further improved in [68], where only two sub-swarms, one of them with a tolerance for inequality constraints, were used. Each particle, and not a sub-swarm, was dynamically assigned the objective function or the constraint, in such a way that more difficult objectives to optimize (satisfy) were assigned more frequently.
- Li et al. [66] adopted a similar approach but using DE as a search algorithm.

Separation of objective function and constraints

- Different multi-population schemes have been proposed.
- Coello [19] divided a GA-population into sub-populations and each sub-population tried to satisfy one constraint of a CNOP and another one optimized the objective function.
- Liang and Suganthan proposed a dynamic assignment of sub-swarms to constraints in PSO [67].
- The approach was further improved in [68], where only two sub-swarms, one of them with a tolerance for inequality constraints, were used. Each particle, and not a sub-swarm, was dynamically assigned the objective function or the constraint, in such a way that more difficult objectives to optimize (satisfy) were assigned more frequently.
- Li et al. [66] adopted a similar approach but using DE as a search algorithm.

Separation of objective function and constraints

- Different multi-population schemes have been proposed.
- Coello [19] divided a GA-population into sub-populations and each sub-population tried to satisfy one constraint of a CNOP and another one optimized the objective function.
- Liang and Suganthan proposed a dynamic assignment of sub-swarms to constraints in PSO [67].
- The approach was further improved in [68], where only two sub-swarms, one of them with a tolerance for inequality constraints, were used. Each particle, and not a sub-swarm, was dynamically assigned the objective function or the constraint, in such a way that more difficult objectives to optimize (satisfy) were assigned more frequently.
- Li et al. [66] adopted a similar approach but using DE as a search algorithm.

Separation of objective function and constraints

- Different multi-population schemes have been proposed.
- Coello [19] divided a GA-population into sub-populations and each sub-population tried to satisfy one constraint of a CNOP and another one optimized the objective function.
- Liang and Suganthan proposed a dynamic assignment of sub-swarms to constraints in PSO [67].
- The approach was further improved in [68], where only two sub-swarms, one of them with a tolerance for inequality constraints, were used. Each particle, and not a sub-swarm, was dynamically assigned the objective function or the constraint, in such a way that more difficult objectives to optimize (satisfy) were assigned more frequently.
- Li et al. [66] adopted a similar approach but using DE as a search algorithm.

Separation of objective function and constraints

- Liu et al. [69] proposed a separation scheme based on a co-evolutionary approach in which two populations are adopted. The first one optimized the objective function without considering the constraints, while the second population aimed to satisfy the constraints of the problem. Each population could migrate solutions to the other.

Separation of objective function and constraints

- Multi-objective optimization concepts (Pareto dominance and Pareto ranking) have been quite popular to solve constrained optimization problems [84]. Two groups can be identified:
  1. CNOP as a bi-objective problem (the original objective function and the sum of constraint violation).
  2. CNOP as a multi-objective optimization problem (the original objective function and each constraint are handled as objectives).

Universidad Veracruzana

Separation of objective function and constraints

- The main shortcomings are related to the lack of bias provided by Pareto ranking when used in a straightforward manner [115], and the difficulties of these approaches to preserve diversity in the population [84].
- Additional mechanisms have been adopted such as Pareto ranking in different search spaces [106, 107, 1, 4], the shrinking of the search space [40] and the use of non-dominated sorting and clustering techniques to generate collaboration among sub-populations [108].

Universidad Veracruzana

Separation of objective function and constraints

- The main shortcomings are related to the lack of bias provided by Pareto ranking when used in a straightforward manner [115], and the difficulties of these approaches to preserve diversity in the population [84].
- Additional mechanisms have been adopted such as Pareto ranking in different search spaces [106, 107, 1, 4], the shrinking of the search space [40] and the use of non-dominated sorting and clustering techniques to generate collaboration among sub-populations [108].

Separation of objective function and constraints

Discussion

- This type of constraint-handling technique has been found to generate an important diversity loss.
- It is important to design appropriate diversity maintenance mechanisms.
- However, they are quite popular (usually no additional parameters required and easy to generalize).

Separation of objective function and constraints

Discussion

- This type of constraint-handling technique has been found to generate an important diversity loss.
- It is important to design appropriate diversity maintenance mechanisms.
- However, they are quite popular (usually no additional parameters required and easy to generalize).

Universidad Veracruzana

Separation of objective function and constraints

Discussion

- This type of constraint-handling technique has been found to generate an important diversity loss.
- It is important to design appropriate diversity maintenance mechanisms.
- However, they are quite popular (usually no additional parameters required and easy to generalize).

Outline

General comments

- The first attempts to generate constraint-handling techniques were similar to an exploration phase, in which a variety of approaches were proposed.
- Main shortcomings:
  - Unsuitable bias.
  - Need of a careful fine-tuning of parameters.
  - Difficult to generalize.
  - High computational cost and difficult implementations.
- The exploitation phase was about to begin.

Universidad Veracruzana

General comments

- The first attempts to generate constraint-handling techniques were similar to an exploration phase, in which a variety of approaches were proposed.
- Main shortcomings:
  - Unsuitable bias.
  - Need of a careful fine-tuning of parameters.
  - Difficult to generalize.
  - High computational cost and difficult implementations.
- The exploitation phase was about to begin.

Universidad Veracruzana

General comments

- The first attempts to generate constraint-handling techniques were similar to an exploration phase, in which a variety of approaches were proposed.
- Main shortcomings:
  - Unsuitable bias.
  - Need of a careful fine-tuning of parameters.
  - Difficult to generalize.
  - High computational cost and difficult implementations.
- The exploitation phase was about to begin.

Outline

Universidad Veracruzana

Feasibility rules

- The feasibility rules proposed by Deb [25, 100] constitute an example of a constraint-handling technique that was proposed several years ago, but whose impact is still present in the literature.
- Its popularity lies on its ability to be coupled to a variety of algorithms, without introducing new parameters.

Universidad Veracruzana

Feasibility rules

- The feasibility rules proposed by Deb [25, 100] constitute an example of a constraint-handling technique that was proposed several years ago, but whose impact is still present in the literature.
- Its popularity lies on its ability to be coupled to a variety of algorithms, without introducing new parameters.

Feasibility rules

Studies on other topics

- Parameter control mechanisms in DE-based constrained numerical optimization by Palomeque and Mezura-Montes (DE self-adaptive parameters, including diversity parameters) [89] and by Zielinski et al. (DE adaptive parameters) [162].
- Zielinski and Laur [160] explored different termination conditions (e.g., improvement-based criteria, movement-based criteria, distribution-based criteria) for DE in constrained optimization.
- Zielinski and Laur [161] studied the effect of the tolerance utilized in the equality constraints, where values between $\epsilon = 1 \times 10^{-7}$ and $\epsilon = 1 \times 10^{-15}$ allowed the algorithm, coupled with the feasibility rules, to reach competitive results.
- Mezura-Montes and Coello Coello [83] explored diversity mechanisms to improve the performance of evolution strategies when solving CNOPs.

Feasibility rules

Studies on other topics

- Parameter control mechanisms in DE-based constrained numerical optimization by Palomeque and Mezura-Montes (DE self-adaptive parameters, including diversity parameters) [89] and by Zielinski et al. (DE adaptive parameters) [162].
- Zielinski and Laur [160] explored different termination conditions (e.g., improvement-based criteria, movement-based criteria, distribution-based criteria) for DE in constrained optimization.
- Zielinski and Laur [161] studied the effect of the tolerance utilized in the equality constraints, where values between $\epsilon = 1 \times 10^{-7}$ and $\epsilon = 1 \times 10^{-15}$ allowed the algorithm, coupled with the feasibility rules, to reach competitive results.
- Mezura-Montes and Coello Coello [83] explored diversity mechanisms to improve the performance of evolution strategies when solving CNOPs.

Feasibility rules

Studies on other topics

- Parameter control mechanisms in DE-based constrained numerical optimization by Palomeque and Mezura-Montes (DE self-adaptive parameters, including diversity parameters) [89] and by Zielinski et al. (DE adaptive parameters) [162].
- Zielinski and Laur [160] explored different termination conditions (e.g., improvement-based criteria, movement-based criteria, distribution-based criteria) for DE in constrained optimization.
- Zielinski and Laur [161] studied the effect of the tolerance utilized in the equality constraints, where values between $\epsilon = 1 \times 10^{-7}$ and $\epsilon = 1 \times 10^{-15}$ allowed the algorithm, coupled with the feasibility rules, to reach competitive results.
- Mezura-Montes and Coello Coello [83] explored diversity mechanisms to improve the performance of evolution strategies when solving CNOPs.

Feasibility rules

Studies on other topics

- Parameter control mechanisms in DE-based constrained numerical optimization by Palomeque and Mezura-Montes (DE self-adaptive parameters, including diversity parameters) [89] and by Zielinski et al. (DE adaptive parameters) [162].
- Zielinski and Laur [160] explored different termination conditions (e.g., improvement-based criteria, movement-based criteria, distribution-based criteria) for DE in constrained optimization.
- Zielinski and Laur [161] studied the effect of the tolerance utilized in the equality constraints, where values between $\epsilon = 1 \times 10^{-7}$ and $\epsilon = 1 \times 10^{-15}$ allowed the algorithm, coupled with the feasibility rules, to reach competitive results.
- Mezura-Montes and Coello Coello [83] explored diversity mechanisms to improve the performance of evolution strategies when solving CNOPs.

Feasibility rules

Multi-operator mechanisms

- The use of feasibility rules has favored the development of approaches with self-adaptive variation operator selection mechanisms on DE:
- jDE-2 by Brest [14], where different variants are combined with an injection of solutions generated at random
- SaDE by Huang et al. [46], where, besides the combination of DE variants, SQP is adopted as a local search operator.
- Four DE variants with four sub-populations with fixed-dynamic size and migration by Elsayed et al. [80].

Universidad Veracruzana

Feasibility rules

Multi-operator mechanisms

- The use of feasibility rules has favored the development of approaches with self-adaptive variation operator selection mechanisms on DE:
- jDE-2 by Brest [14], where different variants are combined with an injection of solutions generated at random
- SaDE by Huang et al. [46], where, besides the combination of DE variants, SQP is adopted as a local search operator.
- Four DE variants with four sub-populations with fixed-dynamic size and migration by Elsayed et al. [80].

Universidad Veracruzana

Feasibility rules

Multi-operator mechanisms

- The use of feasibility rules has favored the development of approaches with self-adaptive variation operator selection mechanisms on DE:
- jDE-2 by Brest [14], where different variants are combined with an injection of solutions generated at random
- SaDE by Huang et al. [46], where, besides the combination of DE variants, SQP is adopted as a local search operator.
- Four DE variants with four sub-populations with fixed-dynamic size and migration by Elsayed et al. [80].

Feasibility rules

Multi-operator mechanisms

- The use of feasibility rules has favored the development of approaches with self-adaptive variation operator selection mechanisms on DE:
- jDE-2 by Brest [14], where different variants are combined with an injection of solutions generated at random
- SaDE by Huang et al. [46], where, besides the combination of DE variants, SQP is adopted as a local search operator.
- Four DE variants with four sub-populations with fixed-dynamic size and migration by Elsayed et al. [80].

Feasibility rules

Multi-operator mechanisms

- ... and studies on PSO:
- A combination of global-local best PSO with dynamic mutation operator by Cagnina et al.[17]. Due to stagnation in some test problems, a further version of this approach was proposed by the same authors in [16], where a bi-population scheme and a "shake" operator were added [16].

Multi-operator mechanisms

- ... and studies on PSO:
- A combination of global-local best PSO with dynamic mutation operator by Cagnina et al.[17]. Due to stagnation in some test problems, a further version of this approach was proposed by the same authors in [16], where a bi-population scheme and a “shake” operator were added [16].

Multi-operator mechanisms

- and also on GAs.
- Elsayed et al. implemented their four sub-population scheme with GAs in [80].
- Elsayed et al. [28] proposed a modified GA where a novel crossover operator called multi-parent crossover and also a randomized operator were added to a real-coded GA.
- Elsayed et al. [27] compared ten different GA variants. The crossover operators employed were triangular crossover, Simulated binary crossover, parent-centric crossover, simplex crossover, and blend crossover. The mutation operators adopted were non-uniform mutation and polynomial crossover.

Universidad Veracruzana

Multi-operator mechanisms

- and also on GAs.
- Elsayed et al. implemented their four sub-population scheme with GAs in [80].
- Elsayed et al. [28] proposed a modified GA where a novel crossover operator called multi-parent crossover and also a randomized operator were added to a real-coded GA.
- Elsayed et al. [27] compared ten different GA variants. The crossover operators employed were triangular crossover, Simulated binary crossover, parent-centric crossover, simplex crossover, and blend crossover. The mutation operators adopted were non-uniform mutation and polynomial crossover.

Universidad Veracruzana

Feasibility rules

Multi-operator mechanisms

- and also on GAs.
- Elsayed et al. implemented their four sub-population scheme with GAs in [80].
- Elsayed et al. [28] proposed a modified GA where a novel crossover operator called multi-parent crossover and also a randomized operator were added to a real-coded GA.
- Elsayed et al. [27] compared ten different GA variants. The crossover operators employed were triangular crossover, Simulated binary crossover, parent-centric crossover, simplex crossover, and blend crossover. The mutation operators adopted were non-uniform mutation and polynomial crossover.

Universidad Veracruzana

Feasibility rules

Multi-operator mechanisms

- and also on GAs.
- Elsayed et al. implemented their four sub-population scheme with GAs in [80].
- Elsayed et al. [28] proposed a modified GA where a novel crossover operator called multi-parent crossover and also a randomized operator were added to a real-coded GA.
- Elsayed et al. [27] compared ten different GA variants. The crossover operators employed were triangular crossover, Simulated binary crossover, parent-centric crossover, simplex crossover, and blend crossover. The mutation operators adopted were non-uniform mutation and polynomial crossover.

Feasibility rules

Combination with special operators

- Barkat Ullah [135] designed a mechanism to force infeasible individuals to move to the feasible region through the application of search space reduction and diversity checking mechanisms designed to avoid premature convergence.
- Mezura-Montes and Cetina-Domíngez [82] proposed a special operator designed to locate infeasible solutions close to the best feasible solution.
- A more recent version was proposed by the authors in [92], where two operators were improved and a direct-search local operator was added to the algorithm.

Feasibility rules

Combination with special operators

- Barkat Ullah [135] designed a mechanism to force infeasible individuals to move to the feasible region through the application of search space reduction and diversity checking mechanisms designed to avoid premature convergence.
- Mezura-Montes and Cetina-Domíngez [82] proposed a special operator designed to locate infeasible solutions close to the best feasible solution.
- A more recent version was proposed by the authors in [92], where two operators were improved and a direct-search local operator was added to the algorithm.

Feasibility rules

Combination with special operators

- Barkat Ullah [135] designed a mechanism to force infeasible individuals to move to the feasible region through the application of search space reduction and diversity checking mechanisms designed to avoid premature convergence.
- Mezura-Montes and Cetina-Domíngez [82] proposed a special operator designed to locate infeasible solutions close to the best feasible solution.
- A more recent version was proposed by the authors in [92], where two operators were improved and a direct-search local operator was added to the algorithm.

Feasibility rules

Adapted to DE

- Zielinski and Laur [159] coupled DE with the feasibility rules in a greedy selection scheme between target and trial vectors.
- Lampinen used a similar DE-based approach in [60]. However, the third criterion (originally based on the sum of constraint violation) was based on Pareto dominance in constraints space. Kukkonen and Lampinen proposed their Generalized Differential Evolution (GDE) [58] based on the aforementioned idea.
- Mezura-Montes et al. [85, 90, 91] proposed a new DE variant coupled with the ability to generate more than one offspring per parent.

Feasibility rules

Adapted to DE

- Zielinski and Laur [159] coupled DE with the feasibility rules in a greedy selection scheme between target and trial vectors.
- Lampinen used a similar DE-based approach in [60]. However, the third criterion (originally based on the sum of constraint violation) was based on Pareto dominance in constraints space. Kukkonen and Lampinen proposed their Generalized Differential Evolution (GDE) [58] based on the aforementioned idea.
- Mezura-Montes et al. [85, 90, 91] proposed a new DE variant coupled with the ability to generate more than one offspring per parent.

Feasibility rules

Adapted to DE

- Zielinski and Laur [159] coupled DE with the feasibility rules in a greedy selection scheme between target and trial vectors.
- Lampinen used a similar DE-based approach in [60]. However, the third criterion (originally based on the sum of constraint violation) was based on Pareto dominance in constraints space. Kukkonen and Lampinen proposed their Generalized Differential Evolution (GDE) [58] based on the aforementioned idea.
- Mezura-Montes et al. [85, 90, 91] proposed a new DE variant coupled with the ability to generate more than one offspring per parent.

Adapted to artificial immune systems

- Cruz et al. [22] and Aragón et al. [6], based on the clonal selection principle used the feasibility rules to rank antibodies based on affinity.
- Aragón et al. [7], based on a T-cell model in which three types of cells (solutions) are adopted, used the feasibility rules as the criteria in the replacement process.

Universidad Veracruzana

Feasibility rules

Adapted to artificial immune systems

- Cruz et al. [22] and Aragón et al. [6], based on the clonal selection principle used the feasibility rules to rank antibodies based on affinity.
- Aragón et al. [7], based on a T-cell model in which three types of cells (solutions) are adopted, used the feasibility rules as the criteria in the replacement process.

Feasibility rules

Adapted to artificial bee colony

- Karaboga and Basturk [50] and Karaboga and Akay [49] changed a greedy selection based only on the objective function values by the use of the feasibility rules with the aim of adapting an artificial bee colony algorithm (ABC) to solve CNOPs.
- Mezura-Montes and Cetina-Domínguez [82] and Mezura-Montes and Velez-Koeppel [92] combined ABC with a smart-flight and a local-search operator, respectively, to improve its performance in constrained search spaces.

Universidad Veracruzana

Feasibility rules

Adapted to artificial bee colony

- Karaboga and Basturk [50] and Karaboga and Akay [49] changed a greedy selection based only on the objective function values by the use of the feasibility rules with the aim of adapting an artificial bee colony algorithm (ABC) to solve CNOPs.
- Mezura-Montes and Cetina-Domínguez [82] and Mezura-Montes and Velez-Koeppel [92] combined ABC with a smart-flight and a local-search operator, respectively, to improve its performance in constrained search spaces.

Feasibility rules

Adapted to other NIAs

- Ma and Simon [76] proposed an improved version of the biogeography-based optimization (BBO) algorithm (inspired on the study of distributions of species over time and space) with the feasibility rules as criteria to choose solutions with the so-called “habitat suitability index”.
- Liu et al. [72] proposed the organizational evolutionary algorithm (OEA). A static penalty function and the feasibility rules were compared as constraint-handling techniques.
- Mezura-Montes and Hernández-Ocaña [88] used the feasibility rules with the Bacterial Foraging Optimization Algorithm (BFOA) in the greedy selection mechanism within the chemotactic loop.
- Landa and Coello [61] adopted the rules in an approach where a cultural DE-based mechanism was developed.

Feasibility rules

Adapted to other NIAs

- Ma and Simon [76] proposed an improved version of the biogeography-based optimization (BBO) algorithm (inspired on the study of distributions of species over time and space) with the feasibility rules as criteria to choose solutions with the so-called “habitat suitability index”.
- Liu et al. [72] proposed the organizational evolutionary algorithm (OEA). A static penalty function and the feasibility rules were compared as constraint-handling techniques.
- Mezura-Montes and Hernández-Ocaña [88] used the feasibility rules with the Bacterial Foraging Optimization Algorithm (BFOA) in the greedy selection mechanism within the chemotactic loop.
- Landa and Coello [61] adopted the rules in an approach where a cultural DE-based mechanism was developed.

Feasibility rules

Adapted to other NIAs

- Ma and Simon [76] proposed an improved version of the biogeography-based optimization (BBO) algorithm (inspired on the study of distributions of species over time and space) with the feasibility rules as criteria to choose solutions with the so-called “habitat suitability index”.
- Liu et al. [72] proposed the organizational evolutionary algorithm (OEA). A static penalty function and the feasibility rules were compared as constraint-handling techniques.
- Mezura-Montes and Hernández-Ocaña [88] used the feasibility rules with the Bacterial Foraging Optimization Algorithm (BFOA) in the greedy selection mechanism within the chemotactic loop.
- Landa and Coello [61] adopted the rules in an approach where a cultural DE-based mechanism was developed.

Feasibility rules

Adapted to other NIAs

- Ma and Simon [76] proposed an improved version of the biogeography-based optimization (BBO) algorithm (inspired on the study of distributions of species over time and space) with the feasibility rules as criteria to choose solutions with the so-called "habitat suitability index".
- Liu et al. [72] proposed the organizational evolutionary algorithm (OEA). A static penalty function and the feasibility rules were compared as constraint-handling techniques.
- Mezura-Montes and Hernández-Ocaña [88] used the feasibility rules with the Bacterial Foraging Optimization Algorithm (BFOA) in the greedy selection mechanism within the chemotactic loop.
- Landa and Coello [61] adopted the rules in an approach where a cultural DE-based mechanism was developed.

Feasibility rules

Used in hybrid approaches

- Muñoz-Zavala et al. [98] used a DE mutation operator to update the local-best particle in PSO.
- Wang et al. [142] implicitly used feasibility rules to rank the particles in a hybrid multi-swarm PSO (HMPSO) where the DE mutation operator was also adopted.
- HMPSO was improved by Lui et al. in [70], where two additional mutation operators were used. The number of evaluations required by the improved approach, decreased in almost 50%.
- He and Wang [38] used simulated annealing (SA) as a local search operator and applied it to the gbest particle at each generation in PSO.

Universidad Veracruzana

Feasibility rules

Used in hybrid approaches

- Muñoz-Zavala et al. [98] used a DE mutation operator to update the local-best particle in PSO.
- Wang et al. [142] implicitly used feasibility rules to rank the particles in a hybrid multi-swarm PSO (HMPSO) where the DE mutation operator was also adopted.
- HMPSO was improved by Lui et al. in [70], where two additional mutation operators were used. The number of evaluations required by the improved approach, decreased in almost 50%.
- He and Wang [38] used simulated annealing (SA) as a local search operator and applied it to the gbest particle at each generation in PSO.

Universidad Veracruzana

Feasibility rules

Used in hybrid approaches

- Muñoz-Zavala et al. [98] used a DE mutation operator to update the local-best particle in PSO.
- Wang et al. [142] implicitly used feasibility rules to rank the particles in a hybrid multi-swarm PSO (HMPSO) where the DE mutation operator was also adopted.
- HMPSO was improved by Lui et al. in [70], where two additional mutation operators were used. The number of evaluations required by the improved approach, decreased in almost 50%.
- He and Wang [38] used simulated annealing (SA) as a local search operator and applied it to the gbest particle at each generation in PSO.

Feasibility rules

Used in hybrid approaches

- Muñoz-Zavala et al. [98] used a DE mutation operator to update the local-best particle in PSO.
- Wang et al. [142] implicitly used feasibility rules to rank the particles in a hybrid multi-swarm PSO (HMPSO) where the DE mutation operator was also adopted.
- HMPSO was improved by Lui et al. in [70], where two additional mutation operators were used. The number of evaluations required by the improved approach, decreased in almost 50%.
- He and Wang [38] used simulated annealing (SA) as a local search operator and applied it to the gbest particle at each generation in PSO.

Universidad Veracruzana

Feasibility rules

Used in memetic algorithms

- Menchaca-Mendez and Coello Coello [81] proposed a DE-based algorithm with a variation of the Nelder-Mead algorithm as a local search operator.
- Sun and Garibaldi [122] proposed an estimation of distribution algorithm (EDA) with SQP as a local search operator.
- Ullah et al. [136, 137] presented an agent-based memetic algorithm where a learning process (a mutation operator chosen by the solution) is used to improve solutions.
- Ali and Kajee-Bagdadi [2] presented a DE-based approach with a modified version of the pattern search method as a local search operator.
- Hamza et al. [37] proposed a DE-based algorithm with a constraint-consensus operator applied to infeasible vectors so as to become them feasible.

Feasibility rules

Used in memetic algorithms

- Menchaca-Mendez and Coello Coello [81] proposed a DE-based algorithm with a variation of the Nelder-Mead algorithm as a local search operator.
- Sun and Garibaldi [122] proposed an estimation of distribution algorithm (EDA) with SQP as a local search operator.
- Ullah et al. [136, 137] presented an agent-based memetic algorithm where a learning process (a mutation operator chosen by the solution) is used to improve solutions.
- Ali and Kajee-Bagdadi [2] presented a DE-based approach with a modified version of the pattern search method as a local search operator.
- Hamza et al. [37] proposed a DE-based algorithm with a constraint-consensus operator applied to infeasible vectors so as to become them feasible.

Feasibility rules

Used in memetic algorithms

- Menchaca-Mendez and Coello Coello [81] proposed a DE-based algorithm with a variation of the Nelder-Mead algorithm as a local search operator.
- Sun and Garibaldi [122] proposed an estimation of distribution algorithm (EDA) with SQP as a local search operator.
- Ullah et al. [136, 137] presented an agent-based memetic algorithm where a learning process (a mutation operator chosen by the solution) is used to improve solutions.
- Ali and Kajee-Bagdadi [2] presented a DE-based approach with a modified version of the pattern search method as a local search operator.
- Hamza et al. [37] proposed a DE-based algorithm with a constraint-consensus operator applied to infeasible vectors so as to become them feasible.

Feasibility rules

Used in memetic algorithms

- Menchaca-Mendez and Coello Coello [81] proposed a DE-based algorithm with a variation of the Nelder-Mead algorithm as a local search operator.
- Sun and Garibaldi [122] proposed an estimation of distribution algorithm (EDA) with SQP as a local search operator.
- Ullah et al. [136, 137] presented an agent-based memetic algorithm where a learning process (a mutation operator chosen by the solution) is used to improve solutions.
- Ali and Kajee-Bagdadi [2] presented a DE-based approach with a modified version of the pattern search method as a local search operator.
- Hamza et al. [37] proposed a DE-based algorithm with a constraint-consensus operator applied to infeasible vectors so as to become them feasible.

Feasibility rules

Used in memetic algorithms

- Menchaca-Mendez and Coello Coello [81] proposed a DE-based algorithm with a variation of the Nelder-Mead algorithm as a local search operator.
- Sun and Garibaldi [122] proposed an estimation of distribution algorithm (EDA) with SQP as a local search operator.
- Ullah et al. [136, 137] presented an agent-based memetic algorithm where a learning process (a mutation operator chosen by the solution) is used to improve solutions.
- Ali and Kajee-Bagdadi [2] presented a DE-based approach with a modified version of the pattern search method as a local search operator.
- Hamza et al. [37] proposed a DE-based algorithm with a constraint-consensus operator applied to infeasible vectors so as to become them feasible.

Feasibility rules

Empirical studies

- Mezura-Montes and Flores-Mendoza compared PSO variants [87].
- Mezura-Montes et al. compared DE variants [86].

Feasibility rules

Empirical studies

- Mezura-Montes and Flores-Mendoza compared PSO variants [87].
- Mezura-Montes et al. compared DE variants [86].

Outline

Universidad Veracruzana

Stochastic ranking (SR)

- Proposed by Runarsson and Yao [114] to deal with the shortcomings of a penalty function (over and under penalization).
- A user-defined parameter called P_f controls the criterion used for comparison of infeasible solutions:
  - Based on their sum of constraint violation
  - Based only on their objective function value.
- SR uses a bubble-sort-like process to rank the solutions in the population.

Universidad Veracruzana

Stochastic ranking (SR)

- Proposed by Runarsson and Yao [114] to deal with the shortcomings of a penalty function (over and under penalization).
- A user-defined parameter called P_f controls the criterion used for comparison of infeasible solutions:
  - Based on their sum of constraint violation
  - Based only on their objective function value.
- SR uses a bubble-sort-like process to rank the solutions in the population.

Universidad Veracruzana

Stochastic ranking (SR)

- Proposed by Runarsson and Yao [114] to deal with the shortcomings of a penalty function (over and under penalization).
- A user-defined parameter called P_f controls the criterion used for comparison of infeasible solutions:
  - Based on their sum of constraint violation
  - Based only on their objective function value.
- SR uses a bubble-sort-like process to rank the solutions in the population.

Stochastic ranking (SR)

```
Begin
    For i=1 to N
        For j=1 to P-1
            u=random(0,1)
            If (φ(I_j) = φ(I_{j+1}) = 0) or (u < P_f)
                If (f(I_j) > f(I_{j+1}))
                    swap(I_j,I_{j+1})
            Else
                If (φ(I_j) > φ(I_{j+1}))
                    swap(I_j,I_{j+1})
        End For
        If (not swap performed)
            break
    End For
End
```

Universidad Veracruzana

Stochastic ranking (SR)

Adapted to DE

- Despite being a ranking process, SR has been adopted by NIAs which do not rank solutions, such as DE.
- Zhang et al. [156] used SR in a DE variant based on [90]. P_f was defined by a dynamic parameter control mechanism (high value at the beginning, low value at the end).
- Liu et al. [73, 71] also used SR in DE and proposed the concept of directional information related to the choice of the most convenient search direction based on the DE mutation operator.
- Fan et al. [30] ranked vectors with SR before the DE operators are applied. The population is split into two sets: (1) the vectors with the highest ranks, and (2) the remaining vectors. The base vector and the vector which determines the search direction are chosen at random from the first set. The other vector is chosen at random from the second set.

Stochastic ranking (SR)

Adapted to DE

- Despite being a ranking process, SR has been adopted by NIAs which do not rank solutions, such as DE.
- Zhang et al. [156] used SR in a DE variant based on [90]. P_f was defined by a dynamic parameter control mechanism (high value at the beginning, low value at the end).
- Liu et al. [73, 71] also used SR in DE and proposed the concept of directional information related to the choice of the most convenient search direction based on the DE mutation operator.
- Fan et al. [30] ranked vectors with SR before the DE operators are applied. The population is split into two sets: (1) the vectors with the highest ranks, and (2) the remaining vectors. The base vector and the vector which determines the search direction are chosen at random from the first set. The other vector is chosen at random from the second set.

Stochastic ranking (SR)

Adapted to DE

- Despite being a ranking process, SR has been adopted by NIAs which do not rank solutions, such as DE.
- Zhang et al. [156] used SR in a DE variant based on [90]. P_f was defined by a dynamic parameter control mechanism (high value at the beginning, low value at the end).
- Liu et al. [73, 71] also used SR in DE and proposed the concept of directional information related to the choice of the most convenient search direction based on the DE mutation operator.
- Fan et al. [30] ranked vectors with SR before the DE operators are applied. The population is split into two sets: (1) the vectors with the highest ranks, and (2) the remaining vectors. The base vector and the vector which determines the search direction are chosen at random from the first set. The other vector is chosen at random from the second set.

Stochastic ranking (SR)

Adapted to DE

- Despite being a ranking process, SR has been adopted by NIAs which do not rank solutions, such as DE.
- Zhang et al. [156] used SR in a DE variant based on [90]. P_f was defined by a dynamic parameter control mechanism (high value at the beginning, low value at the end).
- Liu et al. [73, 71] also used SR in DE and proposed the concept of directional information related to the choice of the most convenient search direction based on the DE mutation operator.
- Fan et al. [30] ranked vectors with SR before the DE operators are applied. The population is split into two sets: (1) the vectors with the highest ranks, and (2) the remaining vectors. The base vector and the vector which determines the search direction are chosen at random from the first set. The other vector is chosen at random from the second set.

Stochastic ranking (SR)

Adapted to other NIAs

- Leguizamón and Coello Coello [63] added SR to an ACO version for dealing with CNOPs. A comparison against traditional penalty functions showed that SR provided better and more robust results.
- Fonseca et al. [32] used ACO with SR to solve discrete structural optimization problems.
- Mallipeddi et al. [77] proposed a two-population evolutionary programming (EP) approach with an external memory to store solutions based on an Euclidean distance measure that aimed to promote diversity. SR was compared against the feasibility rules.

Stochastic ranking (SR)

Adapted to other NIAs

- Leguizamón and Coello Coello [63] added SR to an ACO version for dealing with CNOPs. A comparison against traditional penalty functions showed that SR provided better and more robust results.
- Fonseca et al. [32] used ACO with SR to solve discrete structural optimization problems.
- Mallipeddi et al. [77] proposed a two-population evolutionary programming (EP) approach with an external memory to store solutions based on an Euclidean distance measure that aimed to promote diversity. SR was compared against the feasibility rules.

Universidad Veracruzana

Stochastic ranking (SR)

Adapted to other NIAs

- Leguizamón and Coello Coello [63] added SR to an ACO version for dealing with CNOPs. A comparison against traditional penalty functions showed that SR provided better and more robust results.
- Fonseca et al. [32] used ACO with SR to solve discrete structural optimization problems.
- Mallipeddi et al. [77] proposed a two-population evolutionary programming (EP) approach with an external memory to store solutions based on an Euclidean distance measure that aimed to promote diversity. SR was compared against the feasibility rules.

Universidad Veracruzana

Stochastic ranking (SR)

Applications and other studies

- Huan-Tong et al. [44] used SR with its original search algorithm, an evolution strategy, (ES) for solving reactive power optimization problems.
- Runarsson and Yao [115] improved their ES by adding a differential mutation similar to that used in DE. The authors concluded that a good constraint-handling mechanism needs to be coupled to an appropriate search engine.
- SR has been further developed by Runarsson and Yao [113] in one of the earliest approaches focused on using fitness approximation for constrained numerical optimization (k-nearest-neighbors was adopted for such purpose).

Universidad Veracruzana

Stochastic ranking (SR)

Applications and other studies

- Huan-Tong et al. [44] used SR with its original search algorithm, an evolution strategy, (ES) for solving reactive power optimization problems.
- Runarsson and Yao [115] improved their ES by adding a differential mutation similar to that used in DE. The authors concluded that a good constraint-handling mechanism needs to be coupled to an appropriate search engine.
- SR has been further developed by Runarsson and Yao [113] in one of the earliest approaches focused on using fitness approximation for constrained numerical optimization (*k*-nearest-neighbors was adopted for such purpose).

Universidad Veracruzana

Stochastic ranking (SR)

Applications and other studies

- Huan-Tong et al. [44] used SR with its original search algorithm, an evolution strategy, (ES) for solving reactive power optimization problems.
- Runarsson and Yao [115] improved their ES by adding a differential mutation similar to that used in DE. The authors concluded that a good constraint-handling mechanism needs to be coupled to an appropriate search engine.
- SR has been further developed by Runarsson and Yao [113] in one of the earliest approaches focused on using fitness approximation for constrained numerical optimization (k-nearest-neighbors was adopted for such purpose).

Outline

Universidad Veracruzana

ε-constrained method

- Proposed by Takahama and Sakai [130]. It transforms a CNOP into an unconstrained numerical optimization problem.
- Two main components:
  - A relaxation of the limit to consider a solution as feasible.
  - A lexicographical ordering mechanism in which the minimization of the sum of constraint violation precedes the objective function.
- The value $\varepsilon > 0$, determines the so-called ε-level comparisons between a pair of solutions $\vec{x}_1$ and $\vec{x}_2$ with objective function values $f(\vec{x}_1)$ and $f(\vec{x}_2)$ and sums of constraint violation $\phi(\vec{x}_1)$ and $\phi(\vec{x}_2)$.

Universidad Veracruzana

ε-constrained method

- Proposed by Takahama and Sakai [130]. It transforms a CNOP into an unconstrained numerical optimization problem.
- Two main components:
  - A relaxation of the limit to consider a solution as feasible.
  - A lexicographical ordering mechanism in which the minimization of the sum of constraint violation precedes the objective function.
- The value $\varepsilon > 0$, determines the so-called ε-level comparisons between a pair of solutions $\vec{x}_1$ and $\vec{x}_2$ with objective function values $f(\vec{x}_1)$ and $f(\vec{x}_2)$ and sums of constraint violation $\phi(\vec{x}_1)$ and $\phi(\vec{x}_2)$.

ε-constrained method

- Proposed by Takahama and Sakai [130]. It transforms a CNOP into an unconstrained numerical optimization problem.
- Two main components:
  - A relaxation of the limit to consider a solution as feasible.
  - A lexicographical ordering mechanism in which the minimization of the sum of constraint violation precedes the objective function.
- The value $\varepsilon > 0$, determines the so-called ε-level comparisons between a pair of solutions $\vec{x}_1$ and $\vec{x}_2$ with objective function values $f(\vec{x}_1)$ and $f(\vec{x}_2)$ and sums of constraint violation $\phi(\vec{x}_1)$ and $\phi(\vec{x}_2)$.

ε-constrained method

$$(f(\vec{x}_1), \phi(\vec{x}_1)) <_\varepsilon (f(\vec{x}_2), \phi(\vec{x}_2)) \Leftrightarrow \begin{cases} f(\vec{x}_1) < f(\vec{x}_2) & \text{, if } \phi(\vec{x}_1), \phi(\vec{x}_2) \leq \varepsilon \\ f(\vec{x}_1) < f(\vec{x}_2) & \text{, if } \phi(\vec{x}_1) = \phi(\vec{x}_2) \\ \phi(\vec{x}_1) < \phi(\vec{x}_2) & \text{, otherwise} \end{cases}$$

$$(f(\vec{x}_1), \phi(\vec{x}_1)) \leq_\varepsilon (f(\vec{x}_2), \phi(\vec{x}_2)) \Leftrightarrow \begin{cases} f(\vec{x}_1) \leq f(\vec{x}_2) & \text{, if } \phi(\vec{x}_1), \phi(\vec{x}_2) \leq \varepsilon \\ f(\vec{x}_1) \leq f(\vec{x}_2) & \text{, if } \phi(\vec{x}_1) = \phi(\vec{x}_2) \\ \phi(\vec{x}_1) < \phi(\vec{x}_2) & \text{, otherwise} \end{cases}$$

Universidad Veracruzana

ε-constrained method

- If both solutions in the pairwise comparison are feasible, slightly infeasible (as determined by the ε value) or even if they have the same sum of constraint violation, they are compared using their objective function values.
- If both solutions are infeasible, they are compared based on their sum of constraint violation.
- If $\varepsilon = \infty$, the ε-level comparison works by using only the objective function values as the comparison criteria.
- If $\varepsilon = 0$, then the ε-level comparisons $<_0$ and $\leq_0$ are equivalent to a lexicographical ordering (i.e., $\phi(\vec{x})$ precedes $f(\vec{x})$).

Universidad Veracruzana

ε-constrained method

- If both solutions in the pairwise comparison are feasible, slightly infeasible (as determined by the ε value) or even if they have the same sum of constraint violation, they are compared using their objective function values.
- If both solutions are infeasible, they are compared based on their sum of constraint violation.
- If $\varepsilon = \infty$, the ε-level comparison works by using only the objective function values as the comparison criteria.
- If $\varepsilon = 0$, then the ε-level comparisons $<_0$ and $\leq_0$ are equivalent to a lexicographical ordering (i.e., $\phi(\vec{x})$ precedes $f(\vec{x})$).

Universidad Veracruzana

ε-constrained method

- If both solutions in the pairwise comparison are feasible, slightly infeasible (as determined by the ε value) or even if they have the same sum of constraint violation, they are compared using their objective function values.
- If both solutions are infeasible, they are compared based on their sum of constraint violation.
- If $\varepsilon = \infty$, the ε-level comparison works by using only the objective function values as the comparison criteria.
- If $\varepsilon = 0$, then the ε-level comparisons $<_0$ and $\leq_0$ are equivalent to a lexicographical ordering (i.e., $\phi(\vec{x})$ precedes $f(\vec{x})$).

Universidad Veracruzana

ε-constrained method

- If both solutions in the pairwise comparison are feasible, slightly infeasible (as determined by the ε value) or even if they have the same sum of constraint violation, they are compared using their objective function values.
- If both solutions are infeasible, they are compared based on their sum of constraint violation.
- If $\varepsilon = \infty$, the ε-level comparison works by using only the objective function values as the comparison criteria.
- If $\varepsilon = 0$, then the ε-level comparisons $<_0$ and $\leq_0$ are equivalent to a lexicographical ordering (i.e., $\phi(\vec{x})$ precedes $f(\vec{x})$).

ε-constrained method

Adapted to other NIAs

- Takahama and Sakai used a similar approach called α-constrained method into a GA [123]. Even mathematical programming methods have been used with this approach (Nealder-Mead) [124].
- Wang and Li also adopted the α-constrained method in [140], using DE as their search engine
- Takahama and Sakai adopted the ε-constrained method in PSO [125], and mainly in DE [126].
- Takahama et al. used their constraint-handling technique in a hybrid PSO-GA [130].

Universidad Veracruzana

ε-constrained method

Adapted to other NIAs

- Takahama and Sakai used a similar approach called α-constrained method into a GA [123]. Even mathematical programming methods have been used with this approach (Nealder-Mead) [124].
- Wang and Li also adopted the α-constrained method in [140], using DE as their search engine
- Takahama and Sakai adopted the ε-constrained method in PSO [125], and mainly in DE [126].
- Takahama et al. used their constraint-handling technique in a hybrid PSO-GA [130].

Adapted to other NIAs

- Takahama and Sakai used a similar approach called α-constrained method into a GA [123]. Even mathematical programming methods have been used with this approach (Nealder-Mead) [124].
- Wang and Li also adopted the α-constrained method in [140], using DE as their search engine
- Takahama and Sakai adopted the ε-constrained method in PSO [125], and mainly in DE [126].
- Takahama et al. used their constraint-handling technique in a hybrid PSO-GA [130].

Universidad Veracruzana

ε-constrained method

Adapted to other NIAs

- Takahama and Sakai used a similar approach called α-constrained method into a GA [123]. Even mathematical programming methods have been used with this approach (Nealder-Mead) [124].
- Wang and Li also adopted the α-constrained method in [140], using DE as their search engine
- Takahama and Sakai adopted the ε-constrained method in PSO [125], and mainly in DE [126].
- Takahama et al. used their constraint-handling technique in a hybrid PSO-GA [130].

Universidad Veracruzana

ε-constrained method

Improvements

- The ε value is fine-tuned by Takahama and Sakai with dynamic [124], and adaptive parameter control mechanisms [127].
- Zeng et al. [155] also proposed a dynamic decreasing mechanism inspired in [36].
- A gradient-based mutation was added to the DE-based approach by the same authors in [126] and by Zhang et al. in an EA in [157].
- In [128], Takahama and Sakai improved their approach by adding a decreasing probability on the use of the gradient-based mutation. They also introduced two new mechanisms to deal with boundary constraints (reflecting back and assigning the limit value).
- The authors in [129] added an archive to store solutions and the ability of a vector to generate more than one trial vector.

ε-constrained method

Improvements

- The ε value is fine-tuned by Takahama and Sakai with dynamic [124], and adaptive parameter control mechanisms [127].
- Zeng et al. [155] also proposed a dynamic decreasing mechanism inspired in [36].
- A gradient-based mutation was added to the DE-based approach by the same authors in [126] and by Zhang et al. in an EA in [157].
- In [128], Takahama and Sakai improved their approach by adding a decreasing probability on the use of the gradient-based mutation. They also introduced two new mechanisms to deal with boundary constraints (reflecting back and assigning the limit value).
- The authors in [129] added an archive to store solutions and the ability of a vector to generate more than one trial vector.

ε-constrained method

Improvements

- The ε value is fine-tuned by Takahama and Sakai with dynamic [124], and adaptive parameter control mechanisms [127].
- Zeng et al. [155] also proposed a dynamic decreasing mechanism inspired in [36].
- A gradient-based mutation was added to the DE-based approach by the same authors in [126] and by Zhang et al. in an EA in [157].
- In [128], Takahama and Sakai improved their approach by adding a decreasing probability on the use of the gradient-based mutation. They also introduced two new mechanisms to deal with boundary constraints (reflecting back and assigning the limit value).
- The authors in [129] added an archive to store solutions and the ability of a vector to generate more than one trial vector.

ε-constrained method

Improvements

- The ε value is fine-tuned by Takahama and Sakai with dynamic [124], and adaptive parameter control mechanisms [127].
- Zeng et al. [155] also proposed a dynamic decreasing mechanism inspired in [36].
- A gradient-based mutation was added to the DE-based approach by the same authors in [126] and by Zhang et al. in an EA in [157].
- In [128], Takahama and Sakai improved their approach by adding a decreasing probability on the use of the gradient-based mutation. They also introduced two new mechanisms to deal with boundary constraints (reflecting back and assigning the limit value).
- The authors in [129] added an archive to store solutions and the ability of a vector to generate more than one trial vector.

ε-constrained method

Improvements

- The ε value is fine-tuned by Takahama and Sakai with dynamic [124], and adaptive parameter control mechanisms [127].
- Zeng et al. [155] also proposed a dynamic decreasing mechanism inspired in [36].
- A gradient-based mutation was added to the DE-based approach by the same authors in [126] and by Zhang et al. in an EA in [157].
- In [128], Takahama and Sakai improved their approach by adding a decreasing probability on the use of the gradient-based mutation. They also introduced two new mechanisms to deal with boundary constraints (reflecting back and assigning the limit value).
- The authors in [129] added an archive to store solutions and the ability of a vector to generate more than one trial vector.

ε-constrained method

NIAs turning to the ε-constrained method

- ε-jDE by Brest et al. [12], where different DE variants, parameter self-adaptation (including ε), and population reduction were employed.
- An improved version called jDEsoco was proposed by Brest et al. in [13], where an ageing mechanism to replace those solutions stagnated in a local optimum was added. Moreover, only the 60% of the population was compared by the ε-constrained method and the remaining 40% was compared by only using the objective function value.
- Mezura-Montes et al. used the ε-constrained method in ABC [86]. A dynamic mechanism for the equality constraints tolerance was considered. The results obtained outperformed those reported by a previous ABC version with the feasibility rules [82].

NIAs turning to the ε-constrained method

- ε-jDE by Brest et al. [12], where different DE variants, parameter self-adaptation (including ε), and population reduction were employed.
- An improved version called jDEsoco was proposed by Brest et al. in [13], where an ageing mechanism to replace those solutions stagnated in a local optimum was added. Moreover, only the 60% of the population was compared by the ε-constrained method and the remaining 40% was compared by only using the objective function value.
- Mezura-Montes et al. used the ε-constrained method in ABC [86]. A dynamic mechanism for the equality constraints tolerance was considered. The results obtained outperformed those reported by a previous ABC version with the feasibility rules [82].

ε-constrained method

NIAs turning to the ε-constrained method

- ε-jDE by Brest et al. [12], where different DE variants, parameter self-adaptation (including ε), and population reduction were employed.
- An improved version called jDEsoco was proposed by Brest et al. in [13], where an ageing mechanism to replace those solutions stagnated in a local optimum was added. Moreover, only the 60% of the population was compared by the ε-constrained method and the remaining 40% was compared by only using the objective function value.
- Mezura-Montes et al. used the ε-constrained method in ABC [86]. A dynamic mechanism for the equality constraints tolerance was considered. The results obtained outperformed those reported by a previous ABC version with the feasibility rules [82].

Outline

Universidad Veracruzana

- The most popular approach recently found regarding penalization-based approaches is the adaptive penalty function.
- Dynamic penalty functions, which adopt the current generation number to control the decrement of the penalty factor, are still popular.

Universidad Veracruzana

Novel penalty functions

- The most popular approach recently found regarding penalization-based approaches is the adaptive penalty function.
- Dynamic penalty functions, which adopt the current generation number to control the decrement of the penalty factor, are still popular.

Universidad Veracruzana

Novel penalty functions

Adaptive penalty functions

- Farmani and Wright [31] proposed a two-part adaptive penalty function. The first part increases the fitness of the infeasible solutions with a better value of the objective function with respect to the best solution in the current population. The second part modifies the fitness values of the worst infeasible solutions.
- Tessema and Yen [133] used the number of feasible solutions in the current population to determine penalization values so as to favor slightly infeasible solutions having a good objective function value.

Novel penalty functions

Adaptive penalty functions

- Farmani and Wright [31] proposed a two-part adaptive penalty function. The first part increases the fitness of the infeasible solutions with a better value of the objective function with respect to the best solution in the current population. The second part modifies the fitness values of the worst infeasible solutions.
- Tessema and Yen [133] used the number of feasible solutions in the current population to determine penalization values so as to favor slightly infeasible solutions having a good objective function value.

Novel penalty functions

Adaptive penalty functions

- Mani and Patvardhan [79] proposed a two-population GA-like-based approach. One population evolves by using an adaptive penalty function. The other population evolves based on feasibility rules. Both populations exchange solutions.
- He et al. [39] used two PSO algorithms, one to co-evolve penalty factors and the other one to evolve solutions to the optimization problem.
- Wu [151] proposed an artificial immune system (AIS) where an adaptive penalty function was defined to assign its affinity to each antibody.

Universidad Veracruzana

Novel penalty functions

Adaptive penalty functions

- Mani and Patvardhan [79] proposed a two-population GA-like-based approach. One population evolves by using an adaptive penalty function. The other population evolves based on feasibility rules. Both populations exchange solutions.
- He et al. [39] used two PSO algorithms, one to co-evolve penalty factors and the other one to evolve solutions to the optimization problem.
- Wu [151] proposed an artificial immune system (AIS) where an adaptive penalty function was defined to assign its affinity to each antibody.

Novel penalty functions

Adaptive penalty functions

- Mani and Patvardhan [79] proposed a two-population GA-like-based approach. One population evolves by using an adaptive penalty function. The other population evolves based on feasibility rules. Both populations exchange solutions.
- He et al. [39] used two PSO algorithms, one to co-evolve penalty factors and the other one to evolve solutions to the optimization problem.
- Wu [151] proposed an artificial immune system (AIS) where an adaptive penalty function was defined to assign its affinity to each antibody.

Dynamic penalty functions

- Tasgetiren and Suganthan [132] used of a dynamic penalty function coupled with a multi-population DE algorithm where each populations evolved independently.
- Puzzi and Carpinteri [104] explored a dynamic penalty function based on multiplications instead of summations in a GA-based approach.

Novel penalty functions

Dynamic penalty functions

- Tasgetiren and Suganthan [132] used of a dynamic penalty function coupled with a multi-population DE algorithm where each populations evolved independently.
- Puzzi and Carpinteri [104] explored a dynamic penalty function based on multiplications instead of summations in a GA-based approach.

Novel penalty functions

Static penalty functions

- Deb and Datta [26] obtained suitable penalty factors as follows:
- A bi-objective problem (original objective function and sum of constraint violation ϕ, restricted by a tolerance value) was solved by a MOEA
- A cubic curve to approximate the current obtained Pareto front was generated by using four points whose ϕ values were below a small tolerance.
- The penalty factor was then defined by calculating the corresponding slope at $\phi = 0$.
- After that, a traditional static penalty function was used to solve the original CNOP by using a local search algorithm (Matlab's **fmincon()** procedure was used by the authors) using the solution with the lowest ϕ value from the population of the MOEA as the starting point for the search.

Novel penalty functions

Static penalty functions

- Deb and Datta [26] obtained suitable penalty factors as follows:
- A bi-objective problem (original objective function and sum of constraint violation ϕ, restricted by a tolerance value) was solved by a MOEA
- A cubic curve to approximate the current obtained Pareto front was generated by using four points whose ϕ values were below a small tolerance.
- The penalty factor was then defined by calculating the corresponding slope at $\phi = 0$.
- After that, a traditional static penalty function was used to solve the original CNOP by using a local search algorithm (Matlab's **fmincon()** procedure was used by the authors) using the solution with the lowest ϕ value from the population of the MOEA as the starting point for the search.

Novel penalty functions

Static penalty functions

- Deb and Datta [26] obtained suitable penalty factors as follows:
- A bi-objective problem (original objective function and sum of constraint violation ϕ, restricted by a tolerance value) was solved by a MOEA
- A cubic curve to approximate the current obtained Pareto front was generated by using four points whose ϕ values were below a small tolerance.
- The penalty factor was then defined by calculating the corresponding slope at $\phi = 0$.
- After that, a traditional static penalty function was used to solve the original CNOP by using a local search algorithm (Matlab's **fmincon()** procedure was used by the authors) using the solution with the lowest ϕ value from the population of the MOEA as the starting point for the search.

Novel penalty functions

Static penalty functions

- Deb and Datta [26] obtained suitable penalty factors as follows:
- A bi-objective problem (original objective function and sum of constraint violation ϕ, restricted by a tolerance value) was solved by a MOEA
- A cubic curve to approximate the current obtained Pareto front was generated by using four points whose ϕ values were below a small tolerance.
- The penalty factor was then defined by calculating the corresponding slope at $\phi = 0$.
- After that, a traditional static penalty function was used to solve the original CNOP by using a local search algorithm (Matlab's **fmincon()** procedure was used by the authors) using the solution with the lowest ϕ value from the population of the MOEA as the starting point for the search.

Novel penalty functions

Static penalty functions

- Deb and Datta [26] obtained suitable penalty factors as follows:
- A bi-objective problem (original objective function and sum of constraint violation ϕ, restricted by a tolerance value) was solved by a MOEA
- A cubic curve to approximate the current obtained Pareto front was generated by using four points whose ϕ values were below a small tolerance.
- The penalty factor was then defined by calculating the corresponding slope at $\phi = 0$.
- After that, a traditional static penalty function was used to solve the original CNOP by using a local search algorithm (Matlab's **fmincon()** procedure was used by the authors) using the solution with the lowest ϕ value from the population of the MOEA as the starting point for the search.

Novel penalty functions

Static penalty functions

- In [23], Datta and Deb extended their approach to deal with equality constraints.
- Two main changes:
  - The punishment provided by the penalty value obtained by the bi-objective problem was increased if the local search failed to generate a feasible solution.
  - The small tolerance used for choosing the four points employed to approximate the cubic curve was relaxed.

Static penalty functions

- In [23], Datta and Deb extended their approach to deal with equality constraints.
- Two main changes:
  - The punishment provided by the penalty value obtained by the bi-objective problem was increased if the local search failed to generate a feasible solution.
  - The small tolerance used for choosing the four points employed to approximate the cubic curve was relaxed.

Outline

Universidad Veracruzana

Novel special operators

- The recent proposals based on the use of special operators that have been revised here emphasize the current focus on generating proposals which are easier to generalize.
- Leguizamón and Coello Coello [62] proposed a boundary operator based on conducting a binary search between a feasible and an infeasible solution. Furthermore, the authors designed a strategy to select which constraint (if more than one is present in a CNOP) is analyzed.
- The search algorithm was an ACO variant for continuous search spaces.
- The approach needed an additional constraint-handling technique (a penalty function was used in this case)

Novel special operators

- The recent proposals based on the use of special operators that have been revised here emphasize the current focus on generating proposals which are easier to generalize.
- Leguizamón and Coello Coello [62] proposed a boundary operator based on conducting a binary search between a feasible and an infeasible solution. Furthermore, the authors designed a strategy to select which constraint (if more than one is present in a CNOP) is analyzed.
- The search algorithm was an ACO variant for continuous search spaces.
- The approach needed an additional constraint-handling technique (a penalty function was used in this case)

Novel special operators

- The recent proposals based on the use of special operators that have been revised here emphasize the current focus on generating proposals which are easier to generalize.
- Leguizamón and Coello Coello [62] proposed a boundary operator based on conducting a binary search between a feasible and an infeasible solution. Furthermore, the authors designed a strategy to select which constraint (if more than one is present in a CNOP) is analyzed.
- The search algorithm was an ACO variant for continuous search spaces.
- The approach needed an additional constraint-handling technique (a penalty function was used in this case)

Universidad Veracruzana

Novel special operators

- The recent proposals based on the use of special operators that have been revised here emphasize the current focus on generating proposals which are easier to generalize.
- Leguizamón and Coello Coello [62] proposed a boundary operator based on conducting a binary search between a feasible and an infeasible solution. Furthermore, the authors designed a strategy to select which constraint (if more than one is present in a CNOP) is analyzed.
- The search algorithm was an ACO variant for continuous search spaces.
- The approach needed an additional constraint-handling technique (a penalty function was used in this case)

Novel special operators

- Huang et al. [45] proposed a boundary operator in a two-population approach.
- The first population evolves by using DE as the search engine, based only on the objective function value (regardless of feasibility).
- The second population stores only feasible solutions and the boundary operator uses solutions from both populations to generate new solutions, through the application of the bisection method in the boundaries of the feasible region.
- The Nelder-Mead simplex method was used as a local search operator
- It does not require an additional constraint-handling technique, but a feasible solutions is needed at the beginning of the process.

Novel special operators

- Huang et al. [45] proposed a boundary operator in a two-population approach.
- The first population evolves by using DE as the search engine, based only on the objective function value (regardless of feasibility).
- The second population stores only feasible solutions and the boundary operator uses solutions from both populations to generate new solutions, through the application of the bisection method in the boundaries of the feasible region.
- The Nelder-Mead simplex method was used as a local search operator
- It does not require an additional constraint-handling technique, but a feasible solutions is needed at the beginning of the process.

Novel special operators

- Huang et al. [45] proposed a boundary operator in a two-population approach.
- The first population evolves by using DE as the search engine, based only on the objective function value (regardless of feasibility).
- The second population stores only feasible solutions and the boundary operator uses solutions from both populations to generate new solutions, through the application of the bisection method in the boundaries of the feasible region.
- The Nelder-Mead simplex method was used as a local search operator
- It does not require an additional constraint-handling technique, but a feasible solutions is needed at the beginning of the process.

Universidad Veracruzana

Novel special operators

- Huang et al. [45] proposed a boundary operator in a two-population approach.
- The first population evolves by using DE as the search engine, based only on the objective function value (regardless of feasibility).
- The second population stores only feasible solutions and the boundary operator uses solutions from both populations to generate new solutions, through the application of the bisection method in the boundaries of the feasible region.
- The Nelder-Mead simplex method was used as a local search operator
- It does not require an additional constraint-handling technique, but a feasible solutions is needed at the beginning of the process.

Novel special operators

- Huang et al. [45] proposed a boundary operator in a two-population approach.
- The first population evolves by using DE as the search engine, based only on the objective function value (regardless of feasibility).
- The second population stores only feasible solutions and the boundary operator uses solutions from both populations to generate new solutions, through the application of the bisection method in the boundaries of the feasible region.
- The Nelder-Mead simplex method was used as a local search operator
- It does not require an additional constraint-handling technique, but a feasible solutions is needed at the beginning of the process.

Novel special operators

Constraint satisfaction

- Wanner et al. [149] proposed the Constraint Quadratic Approximation (CQA), which is a special operator designed to restrict an evolutionary algorithm (a GA in this case) to sample solutions inside an object with the same dimensions of the feasible region of the search space.
- This is achieved by a second-order approximation of the objective function and one equality constraint.
- A static penalty function was used to guide the GA search and the equality constraint was transformed into two inequality constraints by using a small ϵ tolerance.

Universidad Veracruzana

Novel special operators

Constraint satisfaction

- Wanner et al. [149] proposed the Constraint Quadratic Approximation (CQA), which is a special operator designed to restrict an evolutionary algorithm (a GA in this case) to sample solutions inside an object with the same dimensions of the feasible region of the search space.
- This is achieved by a second-order approximation of the objective function and one equality constraint.
- A static penalty function was used to guide the GA search and the equality constraint was transformed into two inequality constraints by using a small ϵ tolerance.

Universidad Veracruzana

Novel special operators

Constraint satisfaction

- Wanner et al. [149] proposed the Constraint Quadratic Approximation (CQA), which is a special operator designed to restrict an evolutionary algorithm (a GA in this case) to sample solutions inside an object with the same dimensions of the feasible region of the search space.
- This is achieved by a second-order approximation of the objective function and one equality constraint.
- A static penalty function was used to guide the GA search and the equality constraint was transformed into two inequality constraints by using a small ϵ tolerance.

Universidad Veracruzana

Novel special operators

Constraint satisfaction

- Peconick et al. [102] proposed the Constraint Quadratic Approximation for Multiple Equality Constraints (CQA-MEC).
- An iterative projection algorithm was able to find points satisfying the approximated quadratic constraints with a low computational overhead.
- It still requires the static penalty function to work.
- Araujo et al. [8] extended the previous approaches to deal with multiple inequality constraints by using a special operator in which the locally convex inequality constraints are approximated by quadratic functions, while the locally non-convex inequality constraints are approximated by linear functions.
- The dependence of the static penalty function remains in this last approach.

Novel special operators

Constraint satisfaction

- Peconick et al. [102] proposed the Constraint Quadratic Approximation for Multiple Equality Constraints (CQA-MEC).
- An iterative projection algorithm was able to find points satisfying the approximated quadratic constraints with a low computational overhead.
- It still requires the static penalty function to work.
- Araujo et al. [8] extended the previous approaches to deal with multiple inequality constraints by using a special operator in which the locally convex inequality constraints are approximated by quadratic functions, while the locally non-convex inequality constraints are approximated by linear functions.
- The dependence of the static penalty function remains in this last approach.

Novel special operators

Constraint satisfaction

- Peconick et al. [102] proposed the Constraint Quadratic Approximation for Multiple Equality Constraints (CQA-MEC).
- An iterative projection algorithm was able to find points satisfying the approximated quadratic constraints with a low computational overhead.
- It still requires the static penalty function to work.
- Araujo et al. [8] extended the previous approaches to deal with multiple inequality constraints by using a special operator in which the locally convex inequality constraints are approximated by quadratic functions, while the locally non-convex inequality constraints are approximated by linear functions.
- The dependence of the static penalty function remains in this last approach.

Novel special operators

Constraint satisfaction

- Peconick et al. [102] proposed the Constraint Quadratic Approximation for Multiple Equality Constraints (CQA-MEC).
- An iterative projection algorithm was able to find points satisfying the approximated quadratic constraints with a low computational overhead.
- It still requires the static penalty function to work.
- Araujo et al. [8] extended the previous approaches to deal with multiple inequality constraints by using a special operator in which the locally convex inequality constraints are approximated by quadratic functions, while the locally non-convex inequality constraints are approximated by linear functions.
- The dependence of the static penalty function remains in this last approach.

Novel special operators

Constraint satisfaction

- Peconick et al. [102] proposed the Constraint Quadratic Approximation for Multiple Equality Constraints (CQA-MEC).
- An iterative projection algorithm was able to find points satisfying the approximated quadratic constraints with a low computational overhead.
- It still requires the static penalty function to work.
- Araujo et al. [8] extended the previous approaches to deal with multiple inequality constraints by using a special operator in which the locally convex inequality constraints are approximated by quadratic functions, while the locally non-convex inequality constraints are approximated by linear functions.
- The dependence of the static penalty function remains in this last approach.

Novel special operators

Constraint satisfaction

- Ullah et al.[134] proposed an agent-based memetic algorithm in which the authors adopt a special local operator for equality constraints.
- It is applied to some individuals in the population as follows: the satisfaction of a randomly chosen equality constraint is verified for a given solution. If it is not satisfied, a decision variable, also chosen at random, is updated with the aim to satisfy it. If the constraint is indeed satisfied, two other variables are modified in such a way that the constraint is still satisfied (i.e., the constraint is sampled).
- This special operator is only applied during the early stages of the search because it reduces the diversity in the population.

Universidad Veracruzana

Novel special operators

Constraint satisfaction

- Ullah et al.[134] proposed an agent-based memetic algorithm in which the authors adopt a special local operator for equality constraints.
- It is applied to some individuals in the population as follows: the satisfaction of a randomly chosen equality constraint is verified for a given solution. If it is not satisfied, a decision variable, also chosen at random, is updated with the aim to satisfy it. If the constraint is indeed satisfied, two other variables are modified in such a way that the constraint is still satisfied (i.e., the constraint is sampled).
- This special operator is only applied during the early stages of the search because it reduces the diversity in the population.

Universidad Veracruzana

Novel special operators

Constraint satisfaction

- Ullah et al.[134] proposed an agent-based memetic algorithm in which the authors adopt a special local operator for equality constraints.
- It is applied to some individuals in the population as follows: the satisfaction of a randomly chosen equality constraint is verified for a given solution. If it is not satisfied, a decision variable, also chosen at random, is updated with the aim to satisfy it. If the constraint is indeed satisfied, two other variables are modified in such a way that the constraint is still satisfied (i.e., the constraint is sampled).
- This special operator is only applied during the early stages of the search because it reduces the diversity in the population.

Novel special operators

Feasible directions

- Spadoni and Stefanini [121] transformed a CNOP into an unconstrained search problem by sampling feasible directions instead of solutions of a CNOP.
- Three special operators, related to feasible directions for box constraints, linear inequality constraints, and quadratic inequality constraints, are utilized to generate new solutions by using DE as the search algorithm.
- The main contribution of the approach is that it transforms a CNOP into an unconstrained search problem without using a penalty function. However, it cannot deal with nonlinear (either equality or inequality) constraints.

Universidad Veracruzana

Novel special operators

Feasible directions

- Spadoni and Stefanini [121] transformed a CNOP into an unconstrained search problem by sampling feasible directions instead of solutions of a CNOP.
- Three special operators, related to feasible directions for box constraints, linear inequality constraints, and quadratic inequality constraints, are utilized to generate new solutions by using DE as the search algorithm.
- The main contribution of the approach is that it transforms a CNOP into an unconstrained search problem without using a penalty function. However, it cannot deal with nonlinear (either equality or inequality) constraints.

Novel special operators

Feasible directions

- Spadoni and Stefanini [121] transformed a CNOP into an unconstrained search problem by sampling feasible directions instead of solutions of a CNOP.
- Three special operators, related to feasible directions for box constraints, linear inequality constraints, and quadratic inequality constraints, are utilized to generate new solutions by using DE as the search algorithm.
- The main contribution of the approach is that it transforms a CNOP into an unconstrained search problem without using a penalty function. However, it cannot deal with nonlinear (either equality or inequality) constraints.

Universidad Veracruzana

Novel special operators

General operators made special

- Lu and Chen [75] proposed an approach called self-adaptive velocity particle swarm optimization (SAVPSO).
- Three elements:
  - The position of the feasible region with respect to the whole search space.
  - The connectivity and the shape of the feasible region.
  - The ratio of the feasible region with respect to the search space.
- The velocity formula was modified in such a way that each particle has the ability to self-adjust its velocity according to the aforementioned features of the feasible region.

Novel special operators

General operators made special

- Lu and Chen [75] proposed an approach called self-adaptive velocity particle swarm optimization (SAVPSO).
- Three elements:
  - The position of the feasible region with respect to the whole search space.
  - The connectivity and the shape of the feasible region.
  - The ratio of the feasible region with respect to the search space.
- The velocity formula was modified in such a way that each particle has the ability to self-adjust its velocity according to the aforementioned features of the feasible region.

Novel special operators

General operators made special

- Lu and Chen [75] proposed an approach called self-adaptive velocity particle swarm optimization (SAVPSO).
- Three elements:
  - The position of the feasible region with respect to the whole search space.
  - The connectivity and the shape of the feasible region.
  - The ratio of the feasible region with respect to the search space.
- The velocity formula was modified in such a way that each particle has the ability to self-adjust its velocity according to the aforementioned features of the feasible region.

Novel special operators

Constraint satisfaction

- Wu et al. [152] and Li & Li [65] modified variation operators in NIAs in such a way that the recombination of feasible and infeasible solutions led to the generation of more feasible solutions.
- An adaptive mechanism to maintain infeasible solutions was added to the approach.
- This latter version was specifically based on DE's variation operators [65].

Universidad Veracruzana

Novel special operators

Constraint satisfaction

- Wu et al. [152] and Li & Li [65] modified variation operators in NIAs in such a way that the recombination of feasible and infeasible solutions led to the generation of more feasible solutions.
- An adaptive mechanism to maintain infeasible solutions was added to the approach.
- This latter version was specifically based on DE's variation operators [65].

Universidad Veracruzana

Novel special operators

Constraint satisfaction

- Wu et al. [152] and Li & Li [65] modified variation operators in NIAs in such a way that the recombination of feasible and infeasible solutions led to the generation of more feasible solutions.
- An adaptive mechanism to maintain infeasible solutions was added to the approach.
- This latter version was specifically based on DE's variation operators [65].

Outline

Universidad Veracruzana

Multi-objective concepts

- Despite the fact that empirical evidence has suggested that multi-objective concepts are not well-suited to solve CNOPs, there are highly competitive constraint-handling techniques based on such concepts.
- The use of transformation of a CNOP into a bi-objective optimization problem (objective function and sum of constraint violation) has been preferred over considering each constraint as a separate objective.

Multi-objective concepts

- Despite the fact that empirical evidence has suggested that multi-objective concepts are not well-suited to solve CNOPs, there are highly competitive constraint-handling techniques based on such concepts.
- The use of transformation of a CNOP into a bi-objective optimization problem (objective function and sum of constraint violation) has been preferred over considering each constraint as a separate objective.

Multi-objective concepts

Bi-objective problem

- Ray et al. [109] proposed the Infeasibility Driven Evolutionary Algorithm (IDEA).
- The second objective is the constraint violation measure, (zero value for feasible solutions and a sum of ranking values based on the violation per constraint).
- The union of parents and offspring is split in two sets, one with the feasible solutions and the other with the infeasible ones.
- Non-dominated sorting ranks both sets separately and, based on the proportion of desired feasible solutions, they are chosen first from the infeasible set and later on, the best ranked feasible solutions are chosen.
- SQP was added to IDEA in the Infeasibility Empowered Memetic Algorithm (IMEA) [120].

Multi-objective concepts

Bi-objective problem

- Ray et al. [109] proposed the Infeasibility Driven Evolutionary Algorithm (IDEA).
- The second objective is the constraint violation measure, (zero value for feasible solutions and a sum of ranking values based on the violation per constraint).
- The union of parents and offspring is split in two sets, one with the feasible solutions and the other with the infeasible ones.
- Non-dominated sorting ranks both sets separately and, based on the proportion of desired feasible solutions, they are chosen first from the infeasible set and later on, the best ranked feasible solutions are chosen.
- SQP was added to IDEA in the Infeasibility Empowered Memetic Algorithm (IMEA) [120].

Multi-objective concepts

Bi-objective problem

- Ray et al. [109] proposed the Infeasibility Driven Evolutionary Algorithm (IDEA).
- The second objective is the constraint violation measure, (zero value for feasible solutions and a sum of ranking values based on the violation per constraint).
- The union of parents and offspring is split in two sets, one with the feasible solutions and the other with the infeasible ones.
- Non-dominated sorting ranks both sets separately and, based on the proportion of desired feasible solutions, they are chosen first from the infeasible set and later on, the best ranked feasible solutions are chosen.
- SQP was added to IDEA in the Infeasibility Empowered Memetic Algorithm (IMEA) [120].

Multi-objective concepts

Bi-objective problem

- Ray et al. [109] proposed the Infeasibility Driven Evolutionary Algorithm (IDEA).
- The second objective is the constraint violation measure, (zero value for feasible solutions and a sum of ranking values based on the violation per constraint).
- The union of parents and offspring is split in two sets, one with the feasible solutions and the other with the infeasible ones.
- Non-dominated sorting ranks both sets separately and, based on the proportion of desired feasible solutions, they are chosen first from the infeasible set and later on, the best ranked feasible solutions are chosen.
- SQP was added to IDEA in the Infeasibility Empowered Memetic Algorithm (IMEA) [120].

Multi-objective concepts

Bi-objective problem

- Ray et al. [109] proposed the Infeasibility Driven Evolutionary Algorithm (IDEA).
- The second objective is the constraint violation measure, (zero value for feasible solutions and a sum of ranking values based on the violation per constraint).
- The union of parents and offspring is split in two sets, one with the feasible solutions and the other with the infeasible ones.
- Non-dominated sorting ranks both sets separately and, based on the proportion of desired feasible solutions, they are chosen first from the infeasible set and later on, the best ranked feasible solutions are chosen.
- SQP was added to IDEA in the Infeasibility Empowered Memetic Algorithm (IMEA) [120].

Multi-objective concepts

Bi-objective problem

- Wang et al. [147], in their adaptive trade-off model (ATM) divided the search in three phases based on the feasibility of solutions in the population:
  - Only infeasible solutions (Pareto dominance)
  - Feasible and infeasible solutions (fitness value based on feasible solutions ratio).
  - Only feasible solutions (objective function).
- Wang et al. [146] used the ATM with a NIA in which the variation operators were simplex crossover and one of two mutations.

Universidad Veracruzana

Multi-objective concepts

Bi-objective problem

- Wang et al. [147], in their adaptive trade-off model (ATM) divided the search in three phases based on the feasibility of solutions in the population:
  - Only infeasible solutions (Pareto dominance)
  - Feasible and infeasible solutions (fitness value based on feasible solutions ratio).
  - Only feasible solutions (objective function).
- Wang et al. [146] used the ATM with a NIA in which the variation operators were simplex crossover and one of two mutations.

Multi-objective concepts

Bi-objective problem

- Wang et al. [145] added a shrinking mechanism to ATM in the Accelerated ATM (AATM).
- The ATM was coupled with DE in a recent approach [143], showing an improvement in the results.
- Liu et al. [72] used the ATM in an EA but with two main differences:
  - Good point set crossover was used to generate offspring.
  - Feasibility rules were the criteria to select solutions in the second stage of the ATM.

Universidad Veracruzana

Bi-objective problem

- Wang et al. [145] added a shrinking mechanism to ATM in the Accelerated ATM (AATM).
- The ATM was coupled with DE in a recent approach [143], showing an improvement in the results.
- Liu et al. [72] used the ATM in an EA but with two main differences:
  - Good point set crossover was used to generate offspring.
  - Feasibility rules were the criteria to select solutions in the second stage of the ATM.

Universidad Veracruzana

Multi-objective concepts

Bi-objective problem

- Wang et al. [145] added a shrinking mechanism to ATM in the Accelerated ATM (AATM).
- The ATM was coupled with DE in a recent approach [143], showing an improvement in the results.
- Liu et al. [72] used the ATM in an EA but with two main differences:
  - Good point set crossover was used to generate offspring.
  - Feasibility rules were the criteria to select solutions in the second stage of the ATM.

Bi-objective problem

- Li et al. [64] used a PSO algorithm in which Pareto dominance was used as a criterion in the pbest update process and in the selection of the local-best leaders in a neighborhood. The sum of constraint violation worked as a tie-breaker.
- Venter and Haftka [139] also adopted PSO as their search algorithm. However, the leader selection was based most of the time on the sum of constraint violation, while the rest of the time the criterion was one of the three following choices:
  - The original objective function.
  - The crowding distance.
  - Pareto dominance.

Multi-objective concepts

Bi-objective problem

- Li et al. [64] used a PSO algorithm in which Pareto dominance was used as a criterion in the pbest update process and in the selection of the local-best leaders in a neighborhood. The sum of constraint violation worked as a tie-breaker.
- Venter and Haftka [139] also adopted PSO as their search algorithm. However, the leader selection was based most of the time on the sum of constraint violation, while the rest of the time the criterion was one of the three following choices:
  - The original objective function.
  - The crowding distance.
  - Pareto dominance.

Multi-objective concepts

Bi-objective problem

- Wang et al. [141] used a hybrid selection mechanism based on Pareto dominance and tournament selection into a Adaptive Bacterial Foraging Algorithm (ABFA).
- Wang et al. [144] proposed the use of Pareto dominance in a Hybrid Constrained EA (HCOEA). A global search carried out by an EA is coupled to a local search operator based on SPX.
- Wang et al. [148] proposed a steady state EA by applying orthogonal crossover to a randomly chosen set of solutions in the current population. After that, the non-dominated solutions obtained from the set of offspring are chosen. Alternative, solutions can also be chosen if they have a lower sum of constraint violation.

Universidad Veracruzana

Multi-objective concepts

Bi-objective problem

- Wang et al. [141] used a hybrid selection mechanism based on Pareto dominance and tournament selection into a Adaptive Bacterial Foraging Algorithm (ABFA).
- Wang et al. [144] proposed the use of Pareto dominance in a Hybrid Constrained EA (HCOEA). A global search carried out by an EA is coupled to a local search operator based on SPX.
- Wang et al. [148] proposed a steady state EA by applying orthogonal crossover to a randomly chosen set of solutions in the current population. After that, the non-dominated solutions obtained from the set of offspring are chosen. Alternative, solutions can also be chosen if they have a lower sum of constraint violation.

Universidad Veracruzana

Multi-objective concepts

Bi-objective problem

- Wang et al. [141] used a hybrid selection mechanism based on Pareto dominance and tournament selection into a Adaptive Bacterial Foraging Algorithm (ABFA).
- Wang et al. [144] proposed the use of Pareto dominance in a Hybrid Constrained EA (HCOEA). A global search carried out by an EA is coupled to a local search operator based on SPX.
- Wang et al. [148] proposed a steady state EA by applying orthogonal crossover to a randomly chosen set of solutions in the current population. After that, the non-dominated solutions obtained from the set of offspring are chosen. Alternative, solutions can also be chosen if they have a lower sum of constraint violation.

Universidad Veracruzana

Multi-objective concepts

Three-objective problem

- Reynoso-Meza et al. [111] proposed the spherical-pruning multi-objective optimization differential evolution (sp-MODE).
- The second objective was the sum of constraint violation for inequality constraints and the third objective was the sum of constraint violation for equality constraints.
- An external archive was used to store non-dominated solutions.
- The sphere-pruning operator aims to find the best trade-off between feasibility and the optimization of the objective function.

Multi-objective concepts

Three-objective problem

- Reynoso-Meza et al. [111] proposed the spherical-pruning multi-objective optimization differential evolution (sp-MODE).
- The second objective was the sum of constraint violation for inequality constraints and the third objective was the sum of constraint violation for equality constraints.
- An external archive was used to store non-dominated solutions.
- The sphere-pruning operator aims to find the best trade-off between feasibility and the optimization of the objective function.

Universidad Veracruzana

Multi-objective concepts

Three-objective problem

- Reynoso-Meza et al. [111] proposed the spherical-pruning multi-objective optimization differential evolution (sp-MODE).
- The second objective was the sum of constraint violation for inequality constraints and the third objective was the sum of constraint violation for equality constraints.
- An external archive was used to store non-dominated solutions.
- The sphere-pruning operator aims to find the best trade-off between feasibility and the optimization of the objective function.

Universidad Veracruzana

Multi-objective concepts

Three-objective problem

- Reynoso-Meza et al. [111] proposed the spherical-pruning multi-objective optimization differential evolution (sp-MODE).
- The second objective was the sum of constraint violation for inequality constraints and the third objective was the sum of constraint violation for equality constraints.
- An external archive was used to store non-dominated solutions.
- The sphere-pruning operator aims to find the best trade-off between feasibility and the optimization of the objective function.

Universidad Veracruzana

Three-objective problem

- Zeng et al. [154] proposed converting a constrained problem into a dynamic constrained three-objective optimization problem.
  - The original objective.
  - The constraint-violation (decreasing).
  - A niche count (decreasing).
- Three evolutionary multi-objective optimization algorithms were tested in the approach.

Multi-objective concepts

Three-objective problem

- Zeng et al. [154] proposed converting a constrained problem into a dynamic constrained three-objective optimization problem.
  - The original objective.
  - The constraint-violation (decreasing).
  - A niche count (decreasing).
- Three evolutionary multi-objective optimization algorithms were tested in the approach.

Multi-objective concepts

Many-objective problem

- Gong and Cai [33] used Pareto dominance in the space defined by the constraints of a problem as a constraint-handling mechanism in a DE-based approach.
- An orthogonal process was employed for both, generating the initial population and for applying crossover.
- An external archive stored non-dominated solutions.

Universidad Veracruzana

Many-objective problem

- Gong and Cai [33] used Pareto dominance in the space defined by the constraints of a problem as a constraint-handling mechanism in a DE-based approach.
- An orthogonal process was employed for both, generating the initial population and for applying crossover.
- An external archive stored non-dominated solutions.

Universidad Veracruzana

Multi-objective concepts

Many-objective problem

- Gong and Cai [33] used Pareto dominance in the space defined by the constraints of a problem as a constraint-handling mechanism in a DE-based approach.
- An orthogonal process was employed for both, generating the initial population and for applying crossover.
- An external archive stored non-dominated solutions.

Universidad Veracruzana

Outline

Universidad Veracruzana

Ensemble of constraint-handling techniques

- Mallipeddi and Suganthan [78] proposed an ensemble of four constraint techniques (ECHT):
  - Feasibility rules.
  - Stochastic ranking.
  - A self-adaptive penalty function.
  - The ε-constrained method.
- A four sub-population scheme was considered.
- One EP-based and one DE-based versions were designed.
- Each constraint-handling technique was used to evolve an specific sub-population.
- All sub-populations share all of their offspring.

Universidad Veracruzana

Ensemble of constraint-handling techniques

- Mallipeddi and Suganthan [78] proposed an ensemble of four constraint techniques (ECHT):
  - Feasibility rules.
  - Stochastic ranking.
  - A self-adaptive penalty function.
  - The ε-constrained method.
- A four sub-population scheme was considered.
- One EP-based and one DE-based versions were designed.
- Each constraint-handling technique was used to evolve an specific sub-population.
- All sub-populations share all of their offspring.

Ensemble of constraint-handling techniques

- Mallipeddi and Suganthan [78] proposed an ensemble of four constraint techniques (ECHT):
  - Feasibility rules.
  - Stochastic ranking.
  - A self-adaptive penalty function.
  - The ε-constrained method.
- A four sub-population scheme was considered.
- One EP-based and one DE-based versions were designed.
- Each constraint-handling technique was used to evolve an specific sub-population.
- All sub-populations share all of their offspring,

Universidad Veracruzana

Ensemble of constraint-handling techniques

- Mallipeddi and Suganthan [78] proposed an ensemble of four constraint techniques (ECHT):
  - Feasibility rules.
  - Stochastic ranking.
  - A self-adaptive penalty function.
  - The ε-constrained method.
- A four sub-population scheme was considered.
- One EP-based and one DE-based versions were designed.
- Each constraint-handling technique was used to evolve an specific sub-population.
- All sub-populations share all of their offspring,

Ensemble of constraint-handling techniques

- Mallipeddi and Suganthan [78] proposed an ensemble of four constraint techniques (ECHT):
  - Feasibility rules.
  - Stochastic ranking.
  - A self-adaptive penalty function.
  - The ε-constrained method.
- A four sub-population scheme was considered.
- One EP-based and one DE-based versions were designed.
- Each constraint-handling technique was used to evolve an specific sub-population.
- All sub-populations share all of their offspring,

Universidad Veracruzana

Ensemble of constraint-handling techniques

- Elsayed et al. [29] proposed a DE-based algorithm where the combination of four DE-mutations, two DE recombinations and two constraint-handling techniques (feasibility rules and ε-constrained method) generated sixteen variants which were assigned to each individual in a single-population algorithm.
- The rate of usage for each variant was based on its improvement measured by its ability to generate better solutions.
- A local search algorithm was applied.

Universidad Veracruzana

Ensemble of constraint-handling techniques

- Elsayed et al. [29] proposed a DE-based algorithm where the combination of four DE-mutations, two DE recombinations and two constraint-handling techniques (feasibility rules and ε-constrained method) generated sixteen variants which were assigned to each individual in a single-population algorithm.
- The rate of usage for each variant was based on its improvement measured by its ability to generate better solutions.
- A local search algorithm was applied.

Ensemble of constraint-handling techniques

- Elsayed et al. [29] proposed a DE-based algorithm where the combination of four DE-mutations, two DE recombinations and two constraint-handling techniques (feasibility rules and ε-constrained method) generated sixteen variants which were assigned to each individual in a single-population algorithm.
- The rate of usage for each variant was based on its improvement measured by its ability to generate better solutions.
- A local search algorithm was applied.

Ensemble of constraint-handling techniques

- A similar idea was presented in a combination of two DE variants and a variable neighborhood search with three constraint-handling techniques (feasibility rules, ε-constrained method, and an adaptive penalty function) by Tasgetiren et al. [131].

Ensemble of constraint-handling techniques

- The ECHT opens a new paradigm in constraint-handling techniques.
- The design of mechanisms which allow the combination of approaches that can be seen as complementary (in terms of the way in which they operate).
- However, as the combination of several techniques considerably enhances the capabilities of an approach, it is also required to define parameter values for each of these techniques.
- Parameter control [74] becomes an important issue when designing ensemble approaches.

Universidad Veracruzana

Ensemble of constraint-handling techniques

- The ECHT opens a new paradigm in constraint-handling techniques.
- The design of mechanisms which allow the combination of approaches that can be seen as complementary (in terms of the way in which they operate).
- However, as the combination of several techniques considerably enhances the capabilities of an approach, it is also required to define parameter values for each of these techniques.
- Parameter control [74] becomes an important issue when designing ensemble approaches.

Universidad Veracruzana

Ensemble of constraint-handling techniques

- The ECHT opens a new paradigm in constraint-handling techniques.
- The design of mechanisms which allow the combination of approaches that can be seen as complementary (in terms of the way in which they operate).
- However, as the combination of several techniques considerably enhances the capabilities of an approach, it is also required to define parameter values for each of these techniques.
- Parameter control [74] becomes an important issue when designing ensemble approaches.

Ensemble of constraint-handling techniques

- The ECHT opens a new paradigm in constraint-handling techniques.
- The design of mechanisms which allow the combination of approaches that can be seen as complementary (in terms of the way in which they operate).
- However, as the combination of several techniques considerably enhances the capabilities of an approach, it is also required to define parameter values for each of these techniques.
- Parameter control [74] becomes an important issue when designing ensemble approaches.

Universidad Veracruzana

Outline

Universidad Veracruzana

A bird's eye view

Technique	Core concept	Pros	Cons
FR	Three criteria for pairwise selection	Simple to add into a NIA No extra parameters	May cause premature convergence
SR	Ranking process	Easy to implement	Not all NIAs have ordering in their processes One extra parameter
ε-CM	Transforms a constrained problem into an unconstrained problem	Very competitive performance	Extra parameters Local search for high performance
NPF	Focus on adaptive and dynamic approaches	Well-known transformation process	Some of them add extra parameters
NSO	Focus on boundary operators and equality constraints	Tendency to design easy to generalize operators	Still limited usage
MOC	Focused on bi-objective transformation of a CNOP Pareto dominance	Both, Pareto ranking and dominance still popular	May require an additional constraint-handling technique
ECHT	Combination of two or more constraint-handling techniques	Very competitive performance	Requires the definition of several parameter values

Universidad Veracruzana

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

Universidad Veracruzana

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

Universidad Veracruzana

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

How important is the search algorithm?

- DE is the most preferred algorithm, usually coupled with the feasibility rules.
- GAs are popular when coupled with penalty functions.
- PSO has been mainly coupled with the feasibility rules as well.
- ES has been usually coupled with the stochastic ranking.
- EP, ACO scarcely used.
- Among novel algorithms, ABC with feasibility rules has been particularly popular.
- AIS recently coupled with the feasibility rules.
- Gradient-based local search frequently found.
- Special operators focused on equality constraints.
- Multi-operator algorithms preferred over hybrid approaches.

Benchmarks

The first one

Function	*n*	Type of function	ρ	LI	NI	LE	NE	*a*
g01	13	quadratic	0.0003%	9	0	0	0	6
g02	20	nonlinear	99.9973%	2	0	0	0	1
g03	10	nonlinear	0.0026%	0	0	0	1	1
g04	5	quadratic	27.0079%	4	2	0	0	2
g05	4	nonlinear	0.0000%	2	0	0	3	3
g06	2	nonlinear	0.0057%	0	2	0	0	2
g07	10	quadratic	0.0000%	3	5	0	0	6
g08	2	nonlinear	0.8581%	0	2	0	0	0
g09	7	nonlinear	0.5199%	0	4	0	0	2
g10	8	linear	0.0020%	6	0	0	0	6
g11	2	quadratic	0.0973%	0	0	0	1	1
g12	3	quadratic	4.7697%	0	1	0	0	0
g13	5	nonlinear	0.0000%	0	0	1	2	3
g14	10	nonlinear	0.0000%	0	0	3	0	3
g15	3	quadratic	0.0000%	0	0	1	1	2
g16	5	nonlinear	0.0204%	4	34	0	0	4
g17	6	nonlinear	0.0000%	0	0	0	4	4
g18	9	quadratic	0.0000%	0	13	0	0	6
g19	15	nonlinear	33.4761%	0	5	0	0	0
g20	24	linear	0.0000%	0	6	2	12	16
g21	7	linear	0.0000%	0	1	0	5	6
g22	22	linear	0.0000%	0	1	8	11	19
g23	9	linear	0.0000%	0	2	3	1	6
g24	2	linear	79.6556%	0	2	0	0	2

Benchmarks

The second one

Problem/Search Range	Type of Objective	Number of Constraints		Feasibility Region (ρ)	
		E	I	10D	30D
C01 $[0,10]^D$	Non Separable	0	2 Non Separable	0.997689	1.000000
C02 $[-5.12,5.12]^D$	Separable	1 Separable	2 Separable	0.000000	0.000000
C03 $[-1000,1000]^D$	Non Separable	1 Non Separable	0	0.000000	0.000000
C04 $[-50,50]^D$	Separable	4 2 Non Separable, 2 Separable	0	0.000000	0.000000
C05 $[-600,600]^D$	Separable	2 Separable	0	0.000000	0.000000
C06 $[-600,600]^D$	Separable	2 Rotated	0	0.000000	0.000000

Benchmarks

The second one

C07 [-140,140]^D	Non Separable	0	1 Separable	0.505123	0.503725
C08 [-140,140]^D	Non Separable	0	1 Rotated	0.379512	0.375278
C09 [-500500]^D	Non Separable	1 Separable	0	0.000000	0.000000
C10 [-500,500]^D	Non Separable	1 Rotated	0	0.000000	0.000000
C11 [-100,100]^D	Rotated	1 Non Separable	0	0.000000	0.000000
C12 [-1000,1000]^D	Separable	1 Non Separable	1 Separable	0.000000	0.000000
C13 [-500,500]^D	Separable	0	3 2 Separable, 1 Non Separable	0.000000	0.000000
C14 [-1000,1000]^D	Non Separable	0	3 Separable	0.003112	0.006123
C15 [-1000,1000]^D	Non Separable	0	3 Rotated	0.003210	0.006023
C16 [-10,10]^D	Non Separable	2 Separable	2 1 Separable, 1 Non Separable	0.000000	0.000000
C17 [-10,10]^D	Non Separable	1 Separable	2 Non Separable	0.000000	0.000000
C18 [-50,50]^D	Non Separable	1 Separable	1 Separable	0.000010	0.000000

Benchmarks

The most recent

Problem/Search Range	Type of Objective	Number of Constraints	
		E	I
C01 $[-100,100]^D$	Non Separable	0	1 Separable
C02 $[-100,100]^D$	Non Separable, Rotated	0	1 Non Separable, Rotated
C03 $[-100,100]^D$	Non Separable	1 Separable	1 Separable
C04 $[-10,10]^D$	Separable	0	2 Separable
C05 $[-10,10]^D$	Non Separable	0	2 Non Separable, Rotated
C06 $[-20,20]^D$	Separable	6	0 Separable
C07 $[-50,50]^D$	Separable	2 Separable	0
C08 $[-100,100]^D$	Separable	2 Non Separable	0
C09 $[-10,10]^D$	Separable	2 Non Separable	0
C10 $[-100,100]^D$	Separable	2 Non Separable	0
C11 $[-100,100]^D$	Separable	1 Non Separable	1 Non Separable
C12 $[-100,100]^D$	Separable	0	2 Separable
C13 $[-100,100]^D$	Non Separable	0	3 Separable
C14 $[-100,100]^D$	Non Separable	1 Separable	1 Separable
C15 $[-100,100]^D$	Separable	1	1
C16 $[-100,100]^D$	Separable	1 Non Separable	1 Separable
C17 $[-100,100]^D$	Non Separable	1 Non Separable	1 Separable
C18	Separable	1	2

Benchmarks

The most recent

$[-100,100]^D$			Non Separable
C19 $[-50,50]^D$	Separable	0	2 Non Separable
C20 $[-100,100]^D$	Non Separable	0	2
C21 $[-100,100]^D$	Rotated	0	2 Rotated
C22 $[-100,100]^D$	Rotated	0	3 Rotated
C23 $[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C24 $[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C25 $[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C26 $[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C27 $[-100,100]^D$	Rotated	1 Rotated	2 Rotated
C28 $[-50,50]^D$	Rotated	0	2 Rotated

Performance measures

- Evals (number of solution evaluations to find a feasible solution).
- Progress ratio (difference between the objective function value of the first and best feasible solutions found).
- AFES (average number of solution evaluations in a set of successful runs).
- FP (percentage of feasible runs).
- P (percentage of successful runs).
- SP (successful performance computed by AFES divided by P).

Universidad Veracruzana

Performance measures

- Evals (number of solution evaluations to find a feasible solution).
- Progress ratio (difference between the objective function value of the first and best feasible solutions found).
- AFES (average number of solution evaluations in a set of successful runs).
- FP (percentage of feasible runs).
- P (percentage of successful runs).
- SP (successful performance computed by AFES divided by P).

Universidad Veracruzana

Performance measures

- Evals (number of solution evaluations to find a feasible solution).
- Progress ratio (difference between the objective function value of the first and best feasible solutions found).
- AFES (average number of solution evaluations in a set of successful runs).
- FP (percentage of feasible runs).
- P (percentage of successful runs).
- SP (successful performance computed by AFES divided by P).

Universidad Veracruzana

Performance measures

- Evals (number of solution evaluations to find a feasible solution).
- Progress ratio (difference between the objective function value of the first and best feasible solutions found).
- AFES (average number of solution evaluations in a set of successful runs).
- FP (percentage of feasible runs).
- P (percentage of successful runs).
- SP (successful performance computed by AFES divided by P).

Performance measures

- Evals (number of solution evaluations to find a feasible solution).
- Progress ratio (difference between the objective function value of the first and best feasible solutions found).
- AFES (average number of solution evaluations in a set of successful runs).
- FP (percentage of feasible runs).
- P (percentage of successful runs).
- SP (successful performance computed by AFES divided by P).

Universidad Veracruzana

Performance measures

- Evals (number of solution evaluations to find a feasible solution).
- Progress ratio (difference between the objective function value of the first and best feasible solutions found).
- AFES (average number of solution evaluations in a set of successful runs).
- FP (percentage of feasible runs).
- P (percentage of successful runs).
- SP (successful performance computed by AFES divided by P).

Outline

Universidad Veracruzana

Constraint-handling for EMO

- EMO approaches usually adopt constraint-handling techniques for single-objective optimization.
- Topics of interest:
  - Performance measures.
  - Diversity mechanisms.
  - Boundary operators.
  - Many-objective multi-constrained optimization.

Current trends

Constraint-handling for EMO

- EMO approaches usually adopt constraint-handling techniques for single-objective optimization.
- Topics of interest:
  - Performance measures.
  - Diversity mechanisms.
  - Boundary operators.
  - Many-objective multi-constrained optimization.

Universidad Veracruzana

Current trends

Constraint approximation

- Fitness approximation methods have been extensively applied to unconstrained optimization problems.
- Jin [47] proposed to enlarge the feasible region by using surrogates to ease the generation of fesible solutions.
- Regis [110] used radial basis functions as surrogates to approximate constraints and objective functions in constrained multi-objective optimization problems.

Current trends

Constraint approximation

- Fitness approximation methods have been extensively applied to unconstrained optimization problems.
- Jin [47] proposed to enlarge the feasible region by using surrogates to ease the generation of fesible solutions.
- Regis [110] used radial basis functions as surrogates to approximate constraints and objective functions in constrained multi-objective optimization problems.

Constraint approximation

- Fitness approximation methods have been extensively applied to unconstrained optimization problems.
- Jin [47] proposed to enlarge the feasible region by using surrogates to ease the generation of fesible solutions.
- Regis [110] used radial basis functions as surrogates to approximate constraints and objective functions in constrained multi-objective optimization problems.

Current trends

Constraint approximation

- Datta and Regis [24] proposed an evolution strategy coupled with cubic radial basis functions to solve constrained multi-objective optimization problems.
- Miranda-Varela and Mezura-Montes [97] added feasibility information in the evolution control of a surrogate-assisted differential evolution to solve constrained optimization problems.

Universidad Veracruzana

Constraint approximation

- Datta and Regis [24] proposed an evolution strategy coupled with cubic radial basis functions to solve constrained multi-objective optimization problems.
- Miranda-Varela and Mezura-Montes [97] added feasibility information in the evolution control of a surrogate-assisted differential evolution to solve constrained optimization problems.

Dynamic constraints

- There is a considerable amount of research devoted to deal with unconstrained dynamic optimization problems.
- Initial efforts have focused on constrained dynamic optimization problems.

Universidad Veracruzana

Current trends

Dynamic constraints

- There is a considerable amount of research devoted to deal with unconstrained dynamic optimization problems.
- Initial efforts have focused on constrained dynamic optimization problems.

Current trends

Dynamic constraints

- Nguyen and Yao [99] started the research on DCOPs, by providing a benchmark and an initial comparison of algorithms based mainly on hypermutation and repair methods.

Current trends

Benchmark

Table 1: Main features of the test problems (Nguyen and Yao, 2012).

Problem	Obj. Function	Constraints	DFR	SwO	bNAO	OICB	OISB	Path
g24_u	Dynamic	No Constraints	1	No	No	No	Yes	N/A
g24_1	Dynamic	Static	2	Yes	No	Yes	No	N/A
g24_f	Static	Static	2	No	No	Yes	No	N/A
g24_uf	Static	No Constraints	1	No	No	No	Yes	N/A
g24_2*	Dynamic	Static	2	Yes	No	Yes and No	Yes and No	N/A
g24_2u	Dynamic	No Constraints	1	No	No	No	Yes	N/A
g24_3	Static	Dynamic	2-3	No	Yes	Yes	No	N/A
g24_3b	Dynamic	Dynamic	2-3	Yes	No	Yes	No	N/A
g24_3f	Static	Static	1	No	No	Yes	No	N/A
g24_4	Dynamic	Dynamic	2-3	Yes	No	Yes	No	N/A
g24_5*	Dynamic	Dynamic	2-3	Yes	No	Yes and No	Yes and No	N/A
g24_6a	Dynamic	Static	2	Yes	No	No	Yes	Hard
g24_6b	Dynamic	Static	1	No	No	No	Yes	N/A
g24_6c	Dynamic	Static	2	Yes	No	No	Yes	Easy
g24_6d	Dynamic	Static	2	Yes	No	No	Yes	Hard
g24_7	Static	Dynamic	2	No	No	Yes	No	N/A
g24_8a	Dynamic	No Constraints	1	No	No	No	No	N/A
g24_8b	Dynamic	Static	2	Yes	No	Yes	No	N/A

DFR	Number of disconnected feasible regions
SwO	Switched global optimum between disconnected regions
bNAO	Better newly appear optimum without changing existing ones
OICB	Global optimum is in the constraint boundary
OISB	Global optimum is in the search boundary
Path	Indicate if it is easy or difficult to use mutation to travel between feasible regions
Dynamic	The function is dynamic
Static	There is no change
*	In some change periods, the landscape either is a plateau or contains infinite number of optima and all optima (including the existing optimum) lie in a line parallel to one of the axes

Current trends

Dynamic constraints

- Pal et al. [101] proposed one of the first competitive algorithms for DCOPs based on the gravitational search algorithm and a repair method.
- Ameca-Alducin et al. [3] proposed a DE-based approach with a repair mechanism based on sampling to solve DCOPs. Immigrants and change of DE variants were used as well.

Current trends

Dynamic constraints

- Pal et al. [101] proposed one of the first competitive algorithms for DCOPs based on the gravitational search algorithm and a repair method.
- Ameca-Alducin et al. [3] proposed a DE-based approach with a repair mechanism based on sampling to solve DCOPs. Immigrants and change of DE variants were used as well.

Current trends

Dynamic constraints

- Sharma & Sharma [119] used special operators and Tabu search concepts to deal with DCOPs.
- Aragón et al. [5] proposed a T-cell-inspired approach to solve DCOPs where four sub-populations with different goals interacted in the dynamic search space.
- Bu et al. [15] proposed two new benchmarks and a dynamic species-based PSO with an ensemble of tracking feasible regions strategies.

Current trends

Dynamic constraints

- Sharma & Sharma [119] used special operators and Tabu search concepts to deal with DCOPs.
- Aragón et al. [5] proposed a T-cell-inspired approach to solve DCOPs where four sub-populations with different goals interacted in the dynamic search space.
- Bu et al. [15] proposed two new benchmarks and a dynamic species-based PSO with an ensemble of tracking feasible regions strategies.

Universidad Veracruzana

Current trends

Dynamic constraints

- Sharma & Sharma [119] used special operators and Tabu search concepts to deal with DCOPs.
- Aragón et al. [5] proposed a T-cell-inspired approach to solve DCOPs where four sub-populations with different goals interacted in the dynamic search space.
- Bu et al. [15] proposed two new benchmarks and a dynamic species-based PSO with an ensemble of tracking feasible regions strategies.

Universidad Veracruzana

Current trends

Ensembles/multi-operator NIAs

- This topic is still in its starting phase.
- More combinations and adaptive mechanisms within the ensembles of constraint-handling techniques and/or multi-operator NIAs are expected.

Universidad Veracruzana

Current trends

Ensembles/multi-operator NIAs

- This topic is still in its starting phase.
- More combinations and adaptive mechanisms within the ensembles of constraint-handling techniques and/or multi-operator NIAs are expected.

Current trends

Theory

- There is some work on runtime analysis in constrained search spaces with EAs [158] and also in the usefulness of infeasible solutions in the search process [153].
- Other theoretical studies have focused on some ES variants, such as the (1+1)-ES [10] and more recently the $(1,\lambda)$-ES [9].
- More research in this area is required.

Current trends

Theory

- There is some work on runtime analysis in constrained search spaces with EAs [158] and also in the usefulness of infeasible solutions in the search process [153].
- Other theoretical studies have focused on some ES variants, such as the (1+1)-ES [10] and more recently the $(1,\lambda)$-ES [9].
- More research in this area is required.

Universidad Veracruzana

Current trends

Theory

- There is some work on runtime analysis in constrained search spaces with EAs [158] and also in the usefulness of infeasible solutions in the search process [153].
- Other theoretical studies have focused on some ES variants, such as the (1+1)-ES [10] and more recently the (1,λ)-ES [9].
- More research in this area is required.

References I

S. Akhtar, K. Tai, and T. Ray.
A Socio-Behavioural Simulation Model for Engineering Design Optimization.
Engineering Optimization, 34(4):341–354, 2002.

M. M. Ali and Z. Kajee-Bagdadi.
A local exploration-based differential evolution algorithm for constrained global optimization.
Applied Mathematics and Computation, 208(1):31–48, February 1 2009.

M.-Y. Ameca-Alducin, E. Mezura-Montes, and N. Cruz-Ramírez.
Differential evolution with combined variants and a repair method to solve dynamic constrained optimization problems.
Soft Computing, pages 1–30, 2016.

A. Angantyr, J. Andersson, and J.-O. Aidanpaa.
Constrained Optimization based on a Multiobjective Evolutionary Algorithms.
In *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*, volume 3, pages 1560–1567, Piscataway, New Jersey, December 2003. Canberra, Australia, IEEE Service Center.

V. Aragón, S. Esquivel, and C. Coello-Coello.
Artificial immune system for solving dynamic constrained optimization problems.
In E. Alba, A. Nakib, and P. Siarry, editors, *Metaheuristics for dynamic optimization*, Studies in Computational Intelligence, pages 225–263. Springer, Heidelberg, Germany, 2013.

V. S. Aragón, S. C. Esquivel, and C. A. Coello Coello.
Artificial immune system for solving constrained optimization problems.
Revista Iberoamericana de Inteligencia Artificial, 11(35):55–66, 2007.

Universidad Veracruzana

References II

V. S. Aragón, S. C. Esquivel, and C. A. Coello Coello.
A novel model of artificial immune system for solving constrained optimization problems with dynamic tolerance factor.
In A. Gelbukh and Ángel Fernando Kuri Morales, editors, *Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI'2007)*, pages 19–29, Aguascalientes, México, November 2007. Springer. Lecture Notes in Artificial Intelligence Vol. 4827.

M. C. Araujo, E. F. Wanner, F. G. G. aes, and R. H. C. Takahashi.
Constrained Optimization Based on Quadratic Approximations in Genetic Algorithms.
In E. Mezura-Montes, editor, *Constraint-Handling in Evolutionary Computation*, chapter 9, pages 193–217. Springer. Studies in Computational Intelligence, Volume 198, Berlin, 2009.
ISBN 978-3-642-00618-0.

D. Arnold.
On the behaviour of the (1,λ)-ES for a simple constrained problem.
In *Proceedings of the 2011 ACM/SIGEVO Foundations of Genetic Algorithms (FOGA'2011)*, pages 15–24. ACM Press, June 2011.

D. V. Arnold and D. Brauer.
On the Behaviour of the (1+1)-ES for a Simple Constrained Problem.
In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature–PPSN X*, pages 1–10. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008.

H. J. Barbosa and A. C. Lemonge.
An adaptive penalty scheme in genetic algorithms for constrained optimization problems.
In W. Langdon and et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pages 287–294, San Francisco, California, July 2002. Morgan Kaufmann Publishers.

References III

J. Brest.
Constrained real-parameter optimization with ϵ-self-adaptive differential evolution.
In E. Mezura-Montes, editor, *Constraint-Handling in Evolutionary Optimization*, volume 198, pages 73–93. Springer-Verlag, Studies in Computational Intelligence Series, ISBN:978-3-642-00618-0, 2009.

J. Brest, B. Bošković, and V. Žumer.
An improved self-adaptive differential evolution algorithm in single objective constrained real-parameter optimization.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 1073–1078, Barcelona, Spain, July 2010. IEEE Service Center.

J. Brest, V. Zumer, and M. S. Maucec.
Self-Adaptative Differential Evolution Algorithm in Constrained Real-Parameter Optimization.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 919–926, Vancouver, BC, Canada, July 2006. IEEE.

C. Bu, W. Luo, and L. Yue.
Continuous dynamic constrained optimization with ensemble of locating and tracking feasible regions strategies.
IEEE Transactions on Evolutionary Computation, 21(1):14–33, 2017.

L. Cagnina, S. Esquivel, and C. Coello-Coello.
A Bi-population PSO with a Shake-Mechanism for Solving Constrained Numerical Optimization.
In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 670–676, Singapore, September 2007. IEEE Press.

L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello.
A Particle Swarm Optimizer for Constrained Numerical Optimization.
In T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature (PPSN IX). 9th International Conference*, pages 910–919, Reykjavik, Iceland, September 2006. Reykjavik, Iceland, Springer.
Lecture Notes in Computer Science Vol. 4193.

Universidad Veracruzana

References IV

C. A. C. Coello.
Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art.
Computer Methods in Applied Mechanics and Engineering, 191(11-12):1245–1287, January 2002.

C. A. Coello Coello.
Treating Constraints as Objectives for Single-Objective Evolutionary Optimization.
Engineering Optimization, 32(3):275–308, 2000.

C. A. Coello Coello.
Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems.
Computers in Industry, 41(2):113–127, January 2000.

W. A. Crossley and E. A. Williams.
A Study of Adaptive Penalty Functions for Constrained Genetic Algorithm Based Optimization.
In *AIAA 35th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1997. AIAA Paper 97-0083.

N. Cruz-Cortés, D. Trejo-Pérez, and C. A. C. Coello.
Handling Constraints in Global Optimization using an Artificial Immune System.
In C. Jacob, M. L. Pilat, P. J. Bentley, and J. Timmis, editors, *Artificial Immune Systems. 4th International Conference, ICARIS 2005*, pages 234–247, Banff, Canada, August 2005. Springer.
Lecture Notes in Computer Science Vol. 3627.

R. Datta and K. Deb.
A bi-objective based hybrid evolutionary-classical algorithm for handling equality constraints.
In R. Takahashi and et al., editors, *2011 Evolutionary Multicriterion Optimization Conference (EMO'2011)*, pages 313–327, Heidelberg, Germany, April 2011. Springer-Verlag.
LNCS Vol. 6576.

Universidad Veracruzana

References V

R. Datta and R. G. Regis.
A surrogate-assisted evolution strategy for constrained multi-objective optimization.
Journal of Computational Science, 57:270–284, 2016.

K. Deb.
An Efficient Constraint Handling Method for Genetic Algorithms.
Computer Methods in Applied Mechanics and Engineering, 186(2/4):311–338, 2000.

K. Deb and R. Datta.
A Fast and Accurate Solution of Constrained Optimization Problems Using a Hybrid Bi-Objective and Penalty Function Approach.
In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, pages 165–172, Barcelona, Spain, July 18–23 2010. IEEE Press.

S. Elsayed, R. Sarker, and D. Essam.
A comparative study of different variants of genetic algorithms for constrained optimization.
In *2010 International Conference on Simulated Evolution and Learning (SEAL'2010)*, pages 177–186, Kanpur, India, December 2010. Springer LNCS Vol. 6457.

S. Elsayed, R. Sarker, and D. Essam.
Ga with a new multi-parent crossover for constrained optimization.
In *2011 IEEE Congress on Evolutionary Computation (CEC'2011)*, pages 857–864, New Orleans, USA, July 2011. IEEE Press.

S. Elsayed, R. Sarker, and D. Essam.
Integrated strategies differential evolution algorithm with a local search for constrained optimization.
In *2011 IEEE Congress on Evolutionary Computation (CEC'2011)*, pages 2618–2625, New Orleans, USA, July 2011. IEEE Press.

Universidad Veracruzana

References VI

Z. Fan, J. Liu, T. Sorensen, and P. Wang.
Improved differential evolution based on stochastic ranking for robust layout synthesis of mems components.
IEEE Transactions on Industrial Electronics, 56(4):937–948, April 2009.

R. Farmani and J. A. Wright.
Self-Adaptive Fitness Formulation for Constrained Optimization.
IEEE Transactions on Evolutionary Computation, 7(5):445–455, October 2003.

L. Fonseca, P. Capriles, H. Barbosa, and A. Lemonge.
A stochastic rank-based ant system for discrete structural optimization.
In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pages 68–75, Honolulu, Hawaii, USA, April 2007. IEEE Press.

W. Gong and Z. Cai.
A Multiobjective Differential Evolution Algorithm for Constrained Optimization.
In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 181–188, Hong Kong, June 2008. IEEE Service Center.

A. B. Hadj-Alouane and J. C. Bean.
A Genetic Algorithm for the Multiple-Choice Integer Program.
Operations Research, 45:92–101, 1997.

H. Hamda and M. Schoenauer.
Adaptive techniques for Evolutionary Topological Optimum Design.
In I. Parmee, editor, *Proceedings of the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2000)*, pages 123–136, University of Plymouth, Devon, UK, April 2000. Springer-Verlag.

Universidad Veracruzana

References VII

S. B. Hamida and M. Schoenauer.
ASCHEA: New Results Using Adaptive Segregational Constraint Handling.
In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 1, pages 884–889, Piscataway, New Jersey, May 2002. IEEE Service Center.

N. Hamza, S. Elsayed, D. Essam, and R. Sarker.
Differential evolution combined with constraint consensus for constrained optimization.
In *2011 IEEE Congress on Evolutionary Computation (CEC'2011)*, pages 865–871, New Orleans, USA, July 2011. IEEE Press.

Q. He and L. Wang.
A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization.
Applied Mathematics and Computation, 186(2):1407–1422, March 15th 2007.

Q. He, L. Wang, and F.-Z. Huang.
Nonlinear Constrained Optimization by Enhanced Co-evolutionary PSO.
In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 83–89, Hong Kong, June 2008. IEEE Service Center.

A. Hernández-Aguirre, S. Botello-Rionda, C. A. Coello Coello, G. Lizárraga-Lizárraga, and E. Mezura-Montes.
Handling Constraints using Multiobjective Optimization Concepts.
International Journal for Numerical Methods in Engineering, 59(15):1989–2017, April 2004.

R. Hinterding and Z. Michalewicz.
Your Brains and My Beauty: Parent Matching for Constrained Optimisation.
In *Proceedings of the 5th International Conference on Evolutionary Computation*, pages 810–815, Anchorage, Alaska, May 1998.

Universidad Veracruzana

F. Hoffmeister and J. Sprave.
Problem-independent handling of constraints by use of metric penalty functions.
In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP'96)*, pages 289–294, San Diego, California, February 1996. The MIT Press.

A. Homaifar, S. H. Y. Lai, and X. Qi.
Constrained Optimization via Genetic Algorithms.
Simulation, 62(4):242–254, 1994.

G. Huan-Tong, S. Qing-Xi, J. Feng, and S. Yi-Jie.
An evolution strategy with stochastic ranking for solving reactive power optimization.
In *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS'2009)*, pages 14–17, Shenzhen, Dec 19-20 2009. IEEE Press.
ISBN 978-1-4244-4544-8.

F.-Z. Huang, L. Wang, and Q. He.
A Hybrid Differential Evolution with Double Populations for Constrained Optimization.
In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 18–25, Hong Kong, June 2008. IEEE Service Center.

V. L. Huang, A. K. Qin, and P. N. Suganthan.
Self-adaptative Differential Evolution Algorithm for Constrained Real-Parameter Optimization.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 324–331, Vancouver, BC, Canada, July 2006. IEEE.

Y. Jin.
Surrogate-assisted evolutionary computation: Recent advances and future challenges.
Swarm and Evolutionary Computation, 1(2):61–70, 2011.

References IX

J. Joines and C. Houck.
On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs.
In D. Fogel, editor, *Proceedings of the first IEEE Conference on Evolutionary Computation*, pages 579–584, Orlando, Florida, 1994. IEEE Press.

D. Karaboga and B. Akay.
A modified artificial bee colony (abc) algorithm for constrained optimization problems.
Applied Soft Computing, 11(3):3021–3031, 2011.

D. Karaboga and B. Basturk.
Artificial bee colony(abc) optimization algorithm for solving constrained optimization problems.
In P. Melin, O. Castillo, L. T. Aguilar, J. Kacptrzyk, and W. Pedrycz, editors, *Foundations of Fuzzy Logic and Soft Computing, 12th International Fuzzy Systems Association, World Congress, IFSA 2007*, pages 789–798, Cancun, Mexico, June 2007. Springer, Lecture Notes in Artificial Intelligence Vol. 4529.

S. Kazarlis and V. Petridis.
Varying Fitness Functions in Genetic Algorithms: Studying the Rate of Increase of the Dynamic Penalty Terms.
In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)*, pages 211–220, Heidelberg, Germany, September 1998. Amsterdan, The Netherlands, Springer-Verlag.
Lecture Notes in Computer Science Vol. 1498.

D. G. Kim and P. Husbands.
Riemann Mapping Constraint Handling Method for Genetic Algorithms.
Technical Report CSRP 469, COGS, University of Sussex, UK, 1997.

References X

D. G. Kim and P. Husbands.
Landscape Changes and the Performance of Mapping Based Constraint Handling Methods.
In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)*, pages 221–230, Heidelberg, Germany, September 1998. Amsterdan, The Netherlands, Springer-Verlag.
Lecture Notes in Computer Science Vol. 1498.

D. G. Kim and P. Husbands.
Mapping Based Constraint Handling for Evolutionary Search; Thurston's Circle Packing and Grid Generation.
In I. Parmee, editor, *The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation*, pages 161–173. Springer-Verlag, Plymouth, United Kingdom, April 1998.

R. Kowalczyk.
Constraint Consistent Genetic Algorithms.
In *Proceedings of the 1997 IEEE Conference on Evolutionary Computation*, pages 343–348, Indianapolis, USA, April 1997. IEEE.

S. Koziel and Z. Michalewicz.
A Decoder-based Evolutionary Algorithm for Constrained Parameter Optimization Problems.
In T. Bäck, A. E. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)*, pages 231–240, Heidelberg, Germany, September 1998. Amsterdan, The Netherlands, Springer-Verlag.
Lecture Notes in Computer Science Vol. 1498.

S. Koziel and Z. Michalewicz.
Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization.
Evolutionary Computation, 7(1):19–44, 1999.

Universidad Veracruzana

References XI

S. Kukkonen and J. Lampinen.
Constrained Real-Parameter Optimization with Generalized Differential Evolution.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 911–918, Vancouver, BC, Canada, July 2006. IEEE.

A. Kuri-Morales and C. V. Quezada.
A Universal Eclectic Genetic Algorithm for Constrained Optimization.
In *Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT'98*, pages 518–522, Aachen, Germany, September 1998. Verlag Mainz.

J. Lampinen.
A Constraint Handling Approach for the Differential Evolution Algorithm.
In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 2, pages 1468–1473, Piscataway, New Jersey, May 2002. IEEE Service Center.

R. Landa Becerra and C. A. Coello Coello.
Cultured differential evolution for constrained optimization.
Computer Methods in Applied Mechanics and Engineering, 195(33–36):4303–4322, July 1 2006.

G. Leguizamón and C. A. C. Coello.
Boundary Search for Constrained Numerical Optimization Problems with an Algorithm Inspired on the Ant Colony Metaphor.
IEEE Transactions on Evolutionary Computation, 13(2):350–368, 2009.

G. Leguizamón and C. Coello-Coello.
A Boundary Search based ACO Algorithm Coupled with Stochastic Ranking.
In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 165–172, Singapore, September 2007. IEEE Press.

Universidad Veracruzana

References XII

L. D. Li, X. Li, and X. Yu.
A Multi-Objective Constraint-Handling Method with PSO Algorithm for Constrained Engineering Optimization Problems.
In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 1528–1535, Hong Kong, June 2008. IEEE Service Center.

S. Li and Y. Li.
A new self-adaption differential evolution algorithm based component model.
In *Advances in Computation and Intelligence*, pages 54–63. Springer. Lecture Notes in Computer Science, Vol. 6362, 2010.

Z. Li, J. Liang, X. He, and Z. Shang.
Differential evolution with dynamic constraint-handling mechanism.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 1899–1906, Barcelona, Spain, July 2010. IEEE Service Center.

J. J. Liang and P. N. Suganthan.
Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constrain-Handling Mechanism.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 316–323, Vancouver, BC, Canada, July 2006. IEEE.

J. J. Liang, S. Zhigang, and L. Zhihui.
Coevolutionary comprehensive learning particle swarm optimizer.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 1505–1512, Barcelona, Spain, July 2010. IEEE Service Center.

B. Liu, H. Ma, X. Zhang, B. Liu, and Y. Zhou.
A memetic co-evolutionary differential evolution algorithm for constrained optimization.
In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 2996–3002, Singapore, September 2007. IEEE Press.

Universidad Veracruzana

References XIII

H. Liu, Z. Cai, and Y. Wang.
Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization.
Applied Soft Computing, 10(2):629–640, 2010.

J. Liu, Z. Fan, and E. D. Goodman.
SRaDE: An Adaptive Differential Evolution Based on Stochastic Ranking.
In *2009 Genetic and Evolutionary Computation Conference (GECCO'2009)*, pages 1871–1872, Montreal, Canada, July 8–12 2009. ACM Press.
ISBN 978-1-60558-325-9.

J. Liu, W. Zhong, and L. Hao.
An organizational evolutionary algorithm for numerical optimization.
IEEE Transactions on Systems, Man, and Cybernetics Part B—Cybernetics, 37(4):1052–1064, August 2007.

R. Liu, Y. Li, W. Zhang, and L. Jiao.
Stochastic ranking based differential evolution algorithm for constrained optimization problem.
In *2009 ACM SIGEVO Summit on Genetic and Evolutionary Computation (GEC'2009)*, pages 887–890, Shanghai, China, June 12-14 2009. ACM Press.
ISBN 978-1-60558-326-6.

F. G. Lobo, C. F. Lima, and Z. Michalewicz, editors.
Parameter Setting in Evolutionary Algorithms.
Springer, Berlin, Germany, 2007.

H. Lu and W. Chen.
Self-adaptive velocity particle swarm optimization for solving constrained optimization problems.
Journal of Global Optimization, 41(3):427–445, July 2008.

Universidad Veracruzana

References XIV

H. Ma and D. Simon.
Blended biogeography-based optimization for constrained optimization.
Engineering Applications of Artificial Intelligence, 24(3):517–525, April 2011.

R. Mallipeddi, P. Suganthan, and B. Qu.
Diversity enhanced adaptive evolutionary programming for solving single objective constrained problems.
In *IEEE 2009 Congress on Evolutionary Computation (CEC'2009)*, pages 2106–2113, Trondheim, Norway, May 2009. IEEE Service Center.

R. Mallipeddi and P. N. Suganthan.
Ensemble of Constraint Handling Techniques.
IEEE Transactions On Evolutionary Computation, 14(4):561–579, August 2010.

A. Mani and C. Patvardhan.
A novel hybrid constraint-handling technique for evolutionary optimization.
In *IEEE 2009 Congress on Evolutionary Computation (CEC'2009)*, pages 2577–2583, Trondheim, Norway, May 2009. IEEE Service Center.

S. M.Elsayed, R. A. Sarker, and D. L. Essam.
Multi-operator based evolutionary algorithms for solving constrained optimization problems.
Computers & Operations Research, 38(12):1877–1896, 2011.

A. Menchaca-Méndez and C. A. Coello Coello.
A new proposal to hybridize the nelder-mead method to a differential evolution algorithm for constrained optimization.
In *IEEE 2009 Congress on Evolutionary Computation (CEC'2009)*, pages 2598–2605, Trondheim, Norway, May 2009. IEEE Service Center.

Universidad Veracruzana

References XV

E. Mezura-Montes and O. Cetina-Domínguez.
Exploring promising regions of the search space with the scout bee in the artificial bee colony for constrained optimization.
In C. H. Dagli and et al., editors, *Proceedings of the Artificial Neural Networks in Enginnering Conference (ANNIE'2009)*, volume 19 of *Intelligent Engineering Systems Through Artificial Neural Networks*, pages 253–260, St. Louis, MO, USA, November 2009. ASME Press.

E. Mezura-Montes and C. A. Coello Coello.
A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems.
IEEE Transactions on Evolutionary Computation, 9(1):1–17, February 2005.

E. Mezura-Montes and C. A. Coello-Coello.
Constrained optimization via multiobjective evolutionary algorithms.
In D. C. in Joshua Knowles and K. Deb, editors, *Multiobjective Problems Solving from Nature: From Concepts to Applications*, pages 53–76. Springer-Verlag, Natural Computing Series, 2008, ISBN: 978-3-540-72963-1., 2008.

E. Mezura-Montes, C. A. Coello Coello, and E. I. Tun-Morales.
Simple Feasibility Rules and Differential Evolution for Constrained Optimization.
In R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, editors, *Proceedings of the 3rd Mexican International Conference on Artificial Intelligence (MICAI'2004)*, pages 707–716, Heidelberg, Germany, April 2004. Springer Verlag. Lecture Notes in Artificial Intelligence No. 2972.

E. Mezura-Montes, M. Damian-Araoz, and O. Cetina-Domínguez.
Smart flight and dynamic tolerances in the artificial bee colony for constrained optimization.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 4118–4125, Barcelona, Spain, July 2010. IEEE Service Center.

E. Mezura-Montes and J. I. Flores-Mendoza.
Improved particle swarm optimization in constrained numerical search spaces.
In R. Chiong, editor, *Nature-Inspired Algorithms for Optimization*, volume 193, pages 299–332. Springer-Verlag, Studies in Computational Intelligence Series, 2009, ISBN: 978-3-540-72963-1., 2009.

Universidad Veracruzana

References XVI

E. Mezura-Montes and B. Hernández-Ocaña.
Modified bacterial foraging optimization for engineering design.
In C. H. Dagli and et al., editors, *Proceedings of the Artificial Neural Networks in Enginnering Conference (ANNIE'2009)*, volume 19 of *Intelligent Engineering Systems Through Artificial Neural Networks*, pages 357–364, St. Louis, MO, USA, November 2009. ASME Press.

E. Mezura-Montes and A. G. Palomeque-Ortiz.
Parameter Control in Differential Evolution for Constrained Optimization.
In *2009 Congress on Evolutionary Computation (CEC'2009)*, pages 1375–1382, Tronheim, Norway, May 2009. IEEE Service Center.

E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello.
Promising Infeasibility and Multiple Offspring Incorporated to Differential Evolution for Constrained Optimization.
In H.-G. Beyer, U.-M. O'Reilly, D. Arnold, W. Banzhaf, C. Blum, E. Bonabeau, E. Cant Paz, D. Dasgupta, K. Deb, J. Foste r, E. de Jong, H. Lipson, X. Llora, S. Mancoridis, M. Pelikan, G. Raidl, T. Soule, A. Tyrrell, J.-P. Watson, and E. Zitzler, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 225–232, New York, June 2005. Washington DC, USA, ACM Press.
ISBN 1-59593-010-8.

E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello.
Modified Differential Evolution for Constrained Optimization.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 332–339, Vancouver, BC, Canada, July 2006. IEEE.

E. Mezura-Montes and R. E. Velez-Koeppel.
Elitist artificial bee colony for constrained real-parameter optimization.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 2068–2075, Barcelona, Spain, July 2010. IEEE Service Center.

Universidad Veracruzana

References XVII

Z. Michalewicz.
Genetic Algorithms + Data Structures = Evolution Programs.
Springer-Verlag, third edition, 1996.

Z. Michalewicz and N. F. Attia.
Evolutionary Optimization of Constrained Problems.
In *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pages 98–108. World Scientific, 1994.

Z. Michalewicz and G. Nazhiyath.
Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints.
In D. B. Fogel, editor, *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pages 647–651, Piscataway, New Jersey, 1995. IEEE Press.

Z. Michalewicz and M. Schoenauer.
Evolutionary Algorithms for Constrained Parameter Optimization Problems.
Evolutionary Computation, 4(1):1–32, 1996.

M.-E. Miranda-Varela and E. Mezura-Montes.
Surrogate-assisted differential evolution with an adaptive evolution control based on feasibility to solve constrained optimization problems.
In *Proceedings of the 5th International Conference on Soft Computing for Problem Solving, (SocProsS 2015)*, pages 809–822, Saharanpur, INDIA, 2015. Springer.

A. E. Muñoz-Zavala, A. Hernández-Aguirre, E. R. Villa-Diharce, and S. Botello-Rionda.
PESO+ for Constrained Optimization.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 935–942, Vancouver, BC, Canada, July 2006. IEEE.

Universidad Veracruzana

References XVIII

L.-T. Nguyen and X. Yao.
Continuous dynamic constrained optimization the challenges.
IEEE Transactions on Evolutionary Computation, 16(6):769–786, 2012.

A. I. Oyman, K. Deb, and H.-G. Beyer.
An Alternative Constraint Handling Method for Evolution Strategies.
In *Proceedings of the Congress on Evolutionary Computation 1999 (CEC'99)*, volume 1, pages 612–619, Piscataway, New Jersey, July 1999. IEEE Service Center.

K. Pal, C. Saha, S. Das, and C. Coello-Coello.
Dynamic constrained optimization with offspring repair based gravitational search algorithm.
In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 2414–2421, Cancun, MEXICO, 2013. IEEE Press.

G. Peconick, E. F. Wanner, and R. H. C. Takahashi.
Projection-based local search operator for multiple equality constraints within genetic algorithms.
In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 3043–3049, Singapore, September 2007. IEEE Press.

D. Powell and M. M. Skolnick.
Using genetic algorithms in engineering design optimization with non-linear constraints.
In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, pages 424–431, San Mateo, California, July 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers.

S. Puzzi and A. Carpinteri.
A double-multiplicative dynamic penalty approach for constrained evolutionary optimization.
Structural and Multidisciplinary Optimization, 35(5):431–445, May 2008.

Universidad Veracruzana

References XIX

K. Rasheed.
An Adaptive Penalty Approach for Constrained Genetic-Algorithm Optimization.
In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. L. Riolo, editors, *Proceedings of the Third Annual Genetic Programming Conference*, pages 584–590, San Francisco, California, 1998. Morgan Kaufmann Publishers.

T. Ray, T. Kang, and S. K. Chye.
An Evolutionary Algorithm for Constrained Optimization.
In D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pages 771–777, San Francisco, California, July 2000. Morgan Kaufmann.

T. Ray and K. Liew.
A Swarm with an Effective Information Sharing Mechanism for Unconstrained and Constrained Single Objective Optimization Problems.
In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 1, pages 75–80, Piscataway, New Jersey, May 2001. IEEE Service Center.

T. Ray and K. Liew.
Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior.
IEEE Transactions on Evolutionary Computation, 7(4):386–396, August 2003.

T. Ray, H. K. Singh, A. Isaacs, and W. Smith.
Infeasibility driven evolutionary algorithm for constrained optimization.
In E. Mezura-Montes, editor, *Constraint-Handling in Evolutionary Optimization*, volume 198, pages 145–165. Springer-Verlag, Studies in Computational Intelligence Series, ISBN:978-3-642-00618-0, 2009.

Universidad Veracruzana

References XX

R. G. Regis.
Multi-objective constrained black-box optimization using radial basis function surrogates.
Journal of Computational Science, 16:140–155, 2016.

G. Reynoso-Meza, X. Blasco, J. Sanchis, and M. Martínez.
Multiobjective optimization algorithm for solving constrained single objective problems.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 3418–3424, Barcelona, Spain, July 2010. IEEE Service Center.

R. G. L. Riche, C. Knopf-Lenoir, and R. T. Haftka.
A Segregated Genetic Algorithm for Constrained Structural Optimization.
In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA-95)*, pages 558–565, San Mateo, California, July 1995. University of Pittsburgh, Morgan Kaufmann Publishers.

T. P. Runarsson.
Constrained Evolutionary Optimization by Approximate Ranking and Surrogate Models.
In X. Yao, E. Burke, J. A. Lozano, J. Smith, , J. J. Merelo-Guervós, J. A. Bullinaria, J. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, editors, *Proceedings of 8th Parallel Problem Solving From Nature (PPSN VIII)*, pages 401–410, Heidelberg, Germany, September 2004. Birmingham, UK, Springer-Verlag.
Lecture Notes in Computer Science Vol. 3242.

T. P. Runarsson and X. Yao.
Stochastic Ranking for Constrained Evolutionary Optimization.
IEEE Transactions on Evolutionary Computation, 4(3):284–294, September 2000.

T. P. Runarsson and X. Yao.
Search biases in constrained evolutionary optimization.
IEEE Transactions on Systems, Man, and Cybernetics Part C—Applications and Reviews, 35(2):233–243, May 2005.

Universidad Veracruzana

References XXI

M. Schoenauer and Z. Michalewicz.
Evolutionary Computation at the Edge of Feasibility.
In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the Fourth Conference on Parallel Problem Solving from Nature (PPSN IV)*, pages 245–254, Heidelberg, Germany, September 1996. Berlin, Germany, Springer-Verlag.

M. Schoenauer and Z. Michalewicz.
Boundary Operators for Constrained Parameter Optimization Problems.
In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-97)*, pages 322–329, San Francisco, California, July 1997. Morgan Kaufmann.

M. Schoenauer and S. Xanthakis.
Constrained GA Optimization.
In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, pages 573–580, San Mateo, California, July 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

A. Sharma and D. Sharma.
A constraint guided search for improving evolutionary algorithms.
In *Proceedings of the 19th international conference in Neural information processing (ICONIP)*, pages 269–279, Doha, QATAR, 2012. Springer.

H. K. Singh, T. Ray, and W. Smith.
Performance of infeasibility empowered memetic algorithm for cec 2010 constrained optimization problems.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 3770–3777, Barcelona, Spain, July 2010. IEEE Service Center.

References XXII

M. Spadoni and L. Stefanini.
Handling box, linear and quadratic-convex constraints for boundary optimization with differential evolution algorithms.
In *Proceedings of the Ninth International Conference on Intelligence Systems Design adn Applications (ISDA'2009)*, pages 7–12. IEEE Computer Society, 2009.

J. Sun and J. M. Garibaldi.
A Novel Memetic Algorithm for Constrained Optimization.
In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, pages 549–556, Barcelona, Spain, July 18–23 2010. IEEE Press.

T. Takahama and S. Sakai.
Constrained Optimization by α Constrained Genetic Algorithm (αGA).
Systems and Computers in Japan, 35(5):11–22, May 2004.

T. Takahama and S. Sakai.
Constrained Optimization by Applying the α Constrained Method to the Nonlinear Simplex Method with Mutations.
IEEE Transactions on Evolutionary Computation, 9(5):437–451, October 2005.

T. Takahama and S. Sakai.
Constrained optimization by ε constrained particle swarm optimizer with ε-level control.
In *Proceedings of the 4th IEEE InternationalWorkshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05)*, pages 1019–1029. Muroran, Japan, 2005.

T. Takahama and S. Sakai.
Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 308–315, Vancouver, BC, Canada, July 2006. IEEE.

References XXIII

T. Takahama and S. Sakai.
Constrained optimization by ϵ constrained differential evolution with dynamic ϵ-level control.
In U. K. Chakraborty, editor, *Advances in Differential Evolution*, pages 139–154. Springer, Berlin, 2008.
ISBN 978-3-540-68827-3.

T. Takahama and S. Sakai.
Solving difficult constrained optimization problems by the ϵ constrained differential evolution with gradient-based mutation.
In E. Mezura-Montes, editor, *Constraint-Handling in Evolutionary Optimization*, volume 198, pages 51–72.
Springer-Verlag, Studies in Computational Intelligence Series, ISBN:978-3-642-00618-0, 2009.

T. Takahama and S. Sakai.
Constrained optimization by the ε-constrained differential evolution with an archive and gradient-based mutation.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 1680–1688, Barcelona, Spain, July 2010. IEEE Service Center.

T. Takahama, S. Sakai, and N. Iwane.
Constrained optimization by the epsilon constrained hybrid algorithm of particle swarm optimization and genetic algorithm.
In *AI 2005: Advances in Artificial Intelligence*, pages 389–400. Springer-Verlag, 2005.
Lecture Notes in Artificial Intelligence Vol. 3809.

M. Tasgetiren, P. Suganthan, Q. Pan, R. Mallipeddi, and S. Sarman.
An ensemble of differential evolution algorithms for constrained function optimization.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 967–975, Barcelona, Spain, July 2010. IEEE Service Center.

M. F. Tasgetiren and P. N. Suganthan.
A Multi-Populated Differential Evolution Algorithm for Solving Constrained Optimization Problem.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 340–354, Vancouver, BC, Canada, July 2006. IEEE.

Universidad Veracruzana

References XXIV

B. Tessema and G. G. Yen.
An adaptive penalty formulation for constrained evolutionary optimization.
IEEE Transactions on Systems, Man and Cybernetics Part A—Systems and Humans, 39(3), 2009.

A. S. B. Ullah, R. Sarker, and C. Lokan.
An agent-based memetic algorithm (ama) for nonlinear optimization with equality constraints.
In *IEEE 2009 Congress on Evolutionary Computation (CEC'2009)*, pages 70–77, Trondheim, Norway, May 2009. IEEE Service Center.

A. S. S. M. B. Ullah, R. Sarker, and D. Cornforth.
Search Space Reduction Technique for Constrained Optimization with Tiny Feasible Space.
In *2008 Genetic and Evolutionary Computation Conference (GECCO'2008)*, pages 881–888, Atlanta, USA, July 2008. ACM Press.
ISBN 978-1-60558-131-6.

A. S. S. M. B. Ullah, R. Sarker, D. Cornforth, and C. Lokan.
An Agent-based Memetic Algorithm (AMA) for Solving constrainted Optimization Problems.
In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 999–1006, Singapore, September 2007. IEEE Press.

A. S. S. M. B. Ullah, R. Sarker, D. Cornforth, and C. Lokan.
Ama: a new approach for solving constrained real-valued optimization problems.
Soft Computing, 13(8–9):741–762, 2008.

S. Venkatraman and G. G. Yen.
A Generic Framework for Constrained Optimization Using Genetic Algorithms.
IEEE Transactions on Evolutionary Computation, 9(4), August 2005.

References XXV

G. Venter and R. T. Haftka.
Constrained particle searm optimization using a bi-objective formulation.
Structural and Multidisciplinary Optimization, 40(1-6):65–76, January 2010.

L. Wang and L. po Li.
An effective differential evolution with level comparison for constrained engineering design.
Structural and Multidisciplinary Optimization, 41(6):947–963, June 2010.

Q. Wang, X.-Z. Gao, and C. Wang.
An Adaptive Bacterial Foraging Algorithm For Constrained Optimization.
International Journal of Innovative Computing Information and Control, 6(8):3585–3593, August 2010.

Y. Wang and Z. Cai.
A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems.
Frontiers of Computer Science in China, 3(1):38–52, 2009.

Y. Wang and Z. Cai.
Hybrid differential evolution and adaptive trade-off model to solve constrained optimization problems.
In *2010 Congress on Evolutionary Computation (CEC'2010)*, pages 2846–2850, Barcelona, Spain, July 2010. IEEE Service Center.

Y. Wang, Z. Cai, G. Guo, and Y. Zhou.
Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems.
IEEE Transactions on Systems, Man and Cybernetics Part B–Cybernetics, 37(3):560–575, June 2007.

Y. Wang, Z. Cai, and Y. Zhou.
Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization.
International Journal for Numerical Methods in Engineering, 77(11):1501–1534, 2009.

Universidad Veracruzana

References XXVI

Y. Wang, Z. Cai, Y. Zhou, and Z. Fan.
Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique.
Structural and Multidisciplinary Optimization, 37(4):395–413, January 2009.

Y. Wang, Z. Cai, Y. Zhou, and W. Zeng.
An Adaptive Tradeoff Model for Constrained Evolutionary Optimization.
IEEE Transactions on Evolutionary Computation, 12(1):80–92, February 2008.

Y. Wang, H. Liu, Z. Cai, and Y. Zhou.
An orthogonal design based constrained evolutionary optimization algorithm.
Engineering Optimization, 39(6):715–736, September 2007.

E. F. Wanner, F. G. Guimarï¿½s, R. H. Takahashi, R. R. Saldanha, and P. J. Fleming.
Constraint quadratic approximation operator for treating equality constraints with genetic algorithms.
In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 3, pages 2255–2262, Edinburgh, Scotland, September 2005. IEEE Press.

B. Wu and X. Yu.
Fuzzy Penalty Function Approach for Constrained Function Optimization with Evolutionary Algorithms.
In *Proceedings of the 8th International Conference on Neural Information Processing*, pages 299–304, Shanghai, China, November 2001. Fudan University Press.

J.-Y. Wu.
Solving Constrained Global Optimization Via Artificial Immune System.
International Journal on Artificial Intelligence Tools, 20(1):1–27, February 2011.

Universidad Veracruzana

References XXVII

Y. Wu, Y. Li, and X. Xu.
A Novel Component-Based Model and Ranking Strategy in Constrained Evolutionary Optimization.
In R. Huang, Q. Yang, J. Pei, J. ao Gama, X. Meng, and X. Li, editors, *Advanced Data Mining and Applications, 5th International Conference (ADMA'2009)*, pages 362–373. Springer. Lecture Notes in Computer Science, Vol. 5678, Beijing, China, 2009.

Y. Yu and Z.-H. Zhou.
On the Usefulness of Infeasible Solutions in Evolutionary Search: A Theoretical Study.
In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 835–840, Hong Kong, June 2008. IEEE Service Center.

S. Zeng, R. Jiao, C. Li, X. Li, and J. S. Alkasassbeh.
A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization.
IEEE Transactions on Cybernetics, 2017.

S. Zeng, H. Shi, H. Li, G. Chen, L. Ding, and L. Kang.
A Lower-dimensional-Search Evolutionary Algorithm and Its Application in Constrained Optimization Problem.
In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 1255–1260, Singapore, September 2007. IEEE Press.

M. Zhang, W. Luo, and X. Wang.
Differential evolution with dynamic stochastic selection for constrained optimization.
Information Sciences, 178(15):3043–3074, August 1 2008.

Q. Zhang, S. Zeng, R. Wang, H. Shi, G. Chen, L. Ding, and L. Kang.
Constrained Optimization by the Evolutionary Algorithm with Lower Dimensional Crossover and Gradient-Based Mutation.
In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 273–279, Hong Kong, June 2008. IEEE Service Center.

Universidad Veracruzana

References XXVIII

Y. Zhou and J. He.
A runtime analysis of evolutionary algorithms for constrained optimization problems.
IEEE Transactions on Evolutionary Computation, 11(5):608–619, October 2007.

K. Zielinski and R. Laur.
Constrained Single-Objective Optimization Using Differential Evolution.
In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 927–934, Vancouver, BC, Canada, July 2006. IEEE.

K. Zielinski and R. Laur.
Stopping criteria for differential evolution in constrained single-objective optimization.
In U. K. Chakraborty, editor, *Advances in Differential Evolution*, pages 111–138. Springer, Berlin, 2008.
ISBN 978-3-540-68827-3.

K. Zielinski, S. P. Vudathu, and R. Laur.
Influence of Different Deviations Allowed for Equality Constraints on Particle Swarm Optimization and Differential Evolution.
In N. Krasnogor, G. Nicosia, M. Pavone, and D. Pelta, editors, *Nature Inspired Cooperative Strategies for Optimization*, pages 249–259. Springer, Berlin, 2008.
ISBN 978-3-540-78986-4.

K. Zielinski, X. Wang, and R. Laur.
Comparison of Adaptive Approaches for Differential Evolution.
In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature–PPSN X*, pages 641–650. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008.

Universidad Veracruzana