Metaheuristic Optimization
16. Constraint Handling

Thomas Weise · 汤卫思
tweise@hfuu.edu.cn · http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Faculty of Computer Science and Technology
Institute of Applied Optimization
230601 Shushan District, Hefei, Anhui, China
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99

合肥学院 南艳湖校区/南2区
计算机科学与技术系
应用优化研究所
中国 安徽省 合肥市 蜀山区 230601
经济技术开发区 锦绣大道99号
• Besides optimization criteria, there often are feasibility criteria defining whether a candidate solution makes sense or not
Besides optimization criteria, there often are feasibility criteria defining whether a candidate solution makes sense or not.

**Definition (Feasibility)**

In an optimization problem, \( q \geq 0 \) inequality constraints \( g \) and \( r \geq 0 \) equality constraints \( h \) may be imposed additionally to the objective functions. Then, a candidate solution \( x \) is **feasible**, if and only if it fulfills all constraints:

\[
\text{isFeasible}(x) \iff \begin{align*}
  g_i(x) & \geq 0 & \forall i \in 1 \ldots q \\
  h_j(x) & = 0 & \forall j \in 1 \ldots r
\end{align*}
\]  

(1)
• Ensure that all phenotypes are always feasible
• Ensure that all phenotypes are always feasible
  ① Coding: Use a GPM that constructs feasible phenotypes only \cite{1-3}
Ensure that all phenotypes are always feasible

1. Coding: Use a GPM that constructs feasible phenotypes only \[1^{-3}\]
2. Repairing: Repair infeasible phenotypes, i.e., turn them feasible \[3, 4\]
• Ensure that all phenotypes are always feasible
  1 Coding: Use a GPM that constructs feasible phenotypes only [1–3]
  2 Repairing: Repair infeasible phenotypes, i.e., turn them feasible [3, 4]

✔ Feasibility “removed” from optimization process $\implies$ problem complexity kept low
Genotype/Phenotype Methods

- Ensure that all phenotypes are always feasible
  1. Coding: Use a GPM that constructs feasible phenotypes only \(^{[1-3]}\)
  2. Repairing: Repair infeasible phenotypes, i.e., turn them feasible \(^{[3, 4]}\)

✔ Feasibility “removed” from optimization process \(\implies\) problem complexity kept low

✘ Requires knowledge about “what makes a candidate solution feasible”
• Throw away infeasible candidate solutions \cite{5, 6}
• Throw away infeasible candidate solutions \cite{5, 6}

✔ Very easy to implement
- Throw away infeasible candidate solutions [5, 6]

✔ Very easy to implement

✘ Only possible if many (most) candidate solutions are feasible, otherwise
• Throw away infeasible candidate solutions $^[5, 6]$ 

✔ Very easy to implement

✘ Only possible if many (most) candidate solutions are feasible, otherwise
  • Much effort may be wasted just to discover feasible individuals
• Throw away infeasible candidate solutions \cite{5, 6}

✔ Very easy to implement

✘ Only possible if many (most) candidate solutions are feasible, otherwise
  • Much effort may be wasted just to discover feasible individuals
  • The transition from one feasible individual to another one may be unlikely, the objective landscape becomes rugged with large neutral planes at the worst possible fitness levels
- Throw away infeasible candidate solutions \cite{5, 6}

- Very easy to implement

- Only possible if many (most) candidate solutions are feasible, otherwise
  - Much effort may be wasted just to discover feasible individuals
  - The transition from one feasible individual to another one may be unlikely, the objective landscape becomes rugged with large neutral planes at the worst possible fitness levels
  - The information gained from sampling infeasible individuals is lost!
• Apply evolutionary pressure to guide the search from infeasible to feasible individuals
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals
- Idea by Courant [7] for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value $\nu'$ [7–11]
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals
- Idea by Courant $^7$ for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value $\nu'$ $^{[7-11]}$
- Examples
  1. If $h > 0$ or $h < 0$, there should be a penalty:

\[
\nu(p) = \nu'(p) + \sum_{i=1}^{r} z_i \ast [h_i(p.x)]^2
\]  

(2)
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals
- Idea by Courant \(^{[7]}\) for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value \(\nu'\) \(^{[7-11]}\)
- Examples
  1. If \(h > 0\) or \(h < 0\), there should be a penalty:

\[
\nu(p) = \nu'(p) + \sum_{i=1}^{r} z_i \times [h_i(p.x)]^2 \tag{2}
\]

  2. The closer \(g\) gets to 0, the larger should the penalty be (works if \(g\) is always > 0)

\[
\nu(p) = \nu'(p) + \sum_{i=1}^{q} z \times [g_i(p.x)]^{-1} \tag{3}
\]
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals
- Idea by Courant \cite{7} for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value $\nu'$ \cite{7–11}
- Examples
  1. If $h > 0$ or $h < 0$, there should be a penalty:
  2. The closer $g$ gets to 0, the larger should the penalty be (works if $g$ is always $> 0$)
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals
- Idea by Courant [7] for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value $\nu'$ [7–11]
- Examples
  1. If $h > 0$ or $h < 0$, there should be a penalty:
  2. The closer $g$ gets to 0, the larger should the penalty be (works if $g$ is always $> 0$)
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals
- Idea by Courant [7] for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value $\nu'$ [7–11]
  ✔ Easy to implement
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals
- Idea by Courant [7] for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value $\nu'$ [7–11]

✔ Easy to implement

✘ Knowledge about behavior of objective functions, fitness, and constraint functions necessary
Penalty Function

- Apply evolutionary pressure to guide the search from infeasible to feasible individuals

- Idea by Courant [7] for single-objective optimization in the 1940s: Add a penalty to original objective or fitness value $\nu'$ [7–11]

✔ Easy to implement

✘ Knowledge about behavior of objective functions, fitness, and constraint functions necessary

- Good and a nice method for single-objective optimization, but a bit harder to understand in multi-objective scenarios
- Treat constraints as additional objectives
• Treat constraints as additional objectives

Examples

1. The objective representing the “greater-equal constraint” \( g \) as should be 0 (best) as long as \( g \) is met (\( \geq 0 \)) and > 0 otherwise:

\[
 f^\geq(x) = - \min\{g(x), 0\} \tag{2}
\]
• Treat constraints as additional objectives

• Examples

1. The objective representing the “greater-equal constraint” \(g\) as should be 0 (best) as long as \(g\) is met (\(\geq 0\)) and \(> 0\) otherwise:

\[
f^{\geq}(x) = -\min\{g(x), 0\}
\]  

2. The objective representing “equality constraint” should be \(\epsilon\) (or 0) as long as the constraint is met with a given precision \(\epsilon\), and larger otherwise

\[
f^= (x) = \max\{|h(x)|, \epsilon\} \text{ with an } \epsilon > 0
\]
Constraints as Objectives

- Treat constraints as additional objectives

Examples

1. The objective representing the "greater-equal constraint" \( g \) as should be 0 (best) as long as \( g \) is met (\( \geq 0 \)) and \( > 0 \) otherwise:

\[
f^\geq(x) = -\min\{g(x), 0\}
\]  

(2)

2. The objective representing "equality constraint" should be \( \epsilon \) (or 0) as long as the constraint is met with a given precision \( \epsilon \), and larger otherwise

\[
f^= (x) = \max\{|h(x)|, \epsilon\} \text{ with an } \epsilon > 0
\]  

(3)

✔ Very easy to realize in a MOEA
Constraints as Objectives

- Treat constraints as additional objectives
- Examples
  1. The objective representing the “greater-equal constraint” $g$ as should be 0 (best) as long as $g$ is met ($\geq 0$) and $> 0$ otherwise:

$$f^\geq(x) = -\min\{g(x), 0\}$$

(2)

2. The objective representing “equality constraint” should be $\epsilon$ (or 0) as long as the constraint is met with a given precision $\epsilon$, and larger otherwise

$$f^= (x) = \max\{|h(x)|, \epsilon\} \text{ with an } \epsilon > 0$$

(3)

✔ Very easy to realize in a MOEA

✘ Too many objectives may make the problem very hard to solve

(many-objective optimization optimization [3, 12–19])
Method of Inequalities

- Method of Inequalities (MOI) [20–25]: instead of defining constraints explicitly, goal ranges \([r_i, \overline{r}_i]\) are defined for each of the \(n\) objective function \(f_i\)
Method of Inequalities (MOI) \cite{20-25}: instead of defining constraints explicitly, goal ranges \([r_i, \overline{r}_i]\) are defined for each of the \(n\) objective function \(f_i\).

Pohlheim \cite{26} combines this with Pareto ranking by introducing three classes of candidate solutions.
Method of Inequalities

- Method of Inequalities (MOI)\(^{[20–25]}\): instead of defining constraints explicitly, goal ranges \([\underline{r}_i, \overline{r}_i]\) are defined for each of the \(n\) objective function \(f_i\).

- Pohlheim\(^ { [26]}\) combines this with Pareto ranking by introducing three classes of candidate solutions: a candidate solution either...
Method of Inequalities

- Method of Inequalities (MOI)\textsuperscript{[20–25]}: instead of defining constraints explicitly, goal ranges $[r_{i}, \bar{r}_{i}]$ are defined for each of the $n$ objective function $f_{i}$
- Pohlheim\textsuperscript{[26]} combines this with Pareto ranking by introducing three classes of candidate solutions: a candidate solution either...
  1. fulfills all goals, i.e.,

$$r_{i} \leq f_{i}(x) \leq \bar{r}_{i} \forall i \in \{1 \ldots n\}$$

\text{(4)}
Method of Inequalities

- Method of Inequalities (MOI)\(^{[20–25]}\): instead of defining constraints explicitly, goal ranges \([r_i, \bar{r}_i]\) are defined for each of the \(n\) objective function \(f_i\).

- Pohlheim\(^{[26]}\) combines this with Pareto ranking by introducing three classes of candidate solutions: a candidate solution either...

1. fulfills all goals, i.e.,

\[
\underline{r}_i \leq f_i(x) \leq \bar{r}_i \forall i \in 1 \ldots n \tag{4}
\]

2. fulfills some (but not all) of the goals

\[
(\exists i \in 1 \ldots n : \underline{r}_i \leq f_i(x) \leq \bar{r}_i) \land (\exists j \in 1 \ldots n : (f_j(x) < \underline{r}_j) \lor (f_j(x) > \bar{r}_j)) \tag{5}
\]
Method of Inequalities

- **Method of Inequalities (MOI)** \[^{[20\text{-}25]}]: instead of defining constraints explicitly, goal ranges \([\underline{r}_i, \overline{r}_i]\) are defined for each of the \(n\) objective functions \(f_i\).

- **Pohlheim** \[^{[26]}\] combines this with Pareto ranking by introducing three classes of candidate solutions: a candidate solution either...
  
  1. fulfills all goals, i.e.,

      \[
      \underline{r}_i \leq f_i(x) \leq \overline{r}_i \forall i \in 1 \ldots n
      \]  

  2. fulfills some (but not all) of the goals

      \[(\exists i \in 1 \ldots n : \underline{r}_i \leq f_i(x) \leq \overline{r}_i) \land (\exists j \in 1 \ldots n : (f_j(x) < \underline{r}_j) \lor (f_j(x) > \overline{r}_j))\]

  3. fulfills none of the goals, i.e.,

      \[
      (f_i(x) < \underline{r}_i) \lor (f_i(x) > \overline{r}_i) \forall i \in 1 \ldots n
      \]
• New comparison mechanism
• New comparison mechanism
  Solutions that fulfill all goals are preferred to solutions which fulfill only some goals
Method of Inequalities

- New comparison mechanism
  1. Solutions that fulfill all goals are preferred to solutions which fulfill only some goals
  2. Solutions which fulfill only some goals are preferred to solutions which fulfill no goals
Method of Inequalities

New comparison mechanism

1. Solutions that fulfill all goals are preferred to solutions which fulfill only some goals
2. Solutions which fulfill only some goals are preferred to solutions which fulfill no goals
3. Only solutions in the same group are compared according to the Pareto relationship
• New comparison mechanism
  1. Solutions that fulfill all goals are preferred to solutions which fulfill only some goals
  2. Solutions which fulfill only some goals are preferred to solutions which fulfill no goals
  3. Only solutions in the same group are compared according to the Pareto relationship

• Pareto ranking is performed based on this comparison method
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  \[ \vec{f} = \{ f_1, f_2 \}, f_i : \mathbb{R} \mapsto \mathbb{R} \ \forall i \in \{1, 2\} \]
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  \( \mathbf{f} = \{ f_1, f_2 \}, f_i : \mathbb{R} \mapsto \mathbb{R} \forall i \in \{1, 2\} \)

\[
\begin{align*}
\mathbf{y} &= f_1(x) \\
\mathbf{y} &= f_2(x)
\end{align*}
\]
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  \( \vec{f} = \{ f_1, f_2 \}, f_i : \mathbb{R} \mapsto \mathbb{R} \forall i \in \{1, 2\} \)

MOI with goal ranges \([\underline{r}_i, \bar{r}_i]\) – results change
Example B: Two 2d-Functions

- Two 2-dimensional functions to minimization:
  \[ \vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R} \ \forall i \in \{3, 4\} \]
Example B: Two 2d-Functions

- Two 2-dimensional functions to minimization:
  \[ \vec{f} = \{ f_3, f_4 \}, f_i : \mathbb{R}^2 \mapsto \mathbb{R} \ \forall i \in \{3, 4\} \]

Goal ranges \([\underline{r}_i, \overline{r}_i] \]
Example B: Two 2d-Functions

- Two 2-dimensional functions to minimization:
  \[ \vec{f} = \{f_3, f_4\}, \quad f_i : \mathbb{R}^2 \mapsto \mathbb{R} \quad \forall i \in \{3, 4\} \]

The three different classes
Example B: Two 2d-Functions

- Two 2-dimensional functions to minimization:
  \[ \vec{f} = \{f_3, f_4\}, \, f_i : \mathbb{R}^2 \mapsto \mathbb{R} \, \forall i \in \{3, 4\} \]

\[ \#\text{dom}(\vec{x}, \vec{X}) = |\{ \vec{x}' : (\vec{x}' \in \vec{X}) \land (\vec{x}' \nleq_c \vec{x}) \}| \]

The MOI-domination ranks and optima
Easy to implement
Method of Inequalities

✔ Easy to implement
✔ Fits perfectly well to Pareto ranking / can easily be integrated into this process
Method of Inequalities

✔ Easy to implement
✔ Fits perfectly well to Pareto ranking / can easily be integrated into this process
✘ Constraints must be formulated as objective function ranges
• **Constraint Domination**\(^{[27–29]}\) adapts the Pareto comparison to also consider constraints
- Constraint Domination \([27–29]\) adapts the Pareto comparison to also consider constraints
- A candidate solution \(x_1\) is preferred in comparison to an element \(x_2\) if...
• **Constraint Domination**\textsuperscript{[27–29]} adapts the Pareto comparison to also consider constraints

• A candidate solution \( x_1 \) is preferred in comparison to an element \( x_2 \) if...
  1. \( x_1 \) is feasible while \( x_2 \) is not,
Constraint Domination adapts the Pareto comparison to also consider constraints. A candidate solution $x_1$ is preferred in comparison to an element $x_2$ if

1. $x_1$ is feasible while $x_2$ is not,
2. $x_1$ and $x_2$ both are infeasible but $x_1$ has a smaller overall constraint violation, or
• **Constraint Domination** \([27–29]\) adapts the Pareto comparison to also consider constraints

• A candidate solution \(x_1\) is prefered in comparison to an element \(x_2\) if...

  1. \(x_1\) is feasible while \(x_2\) is not,
  2. \(x_1\) and \(x_2\) both are infeasible but \(x_1\) has a smaller overall constraint violation, or
  3. \(x_1\) and \(x_2\) are both feasible but \(x_1\) dominates \(x_2\).
• Constraint Domination\(^{[27–29]}\) adapts the Pareto comparison to also consider constraints

• A candidate solution \(x_1\) is prefered in comparison to an element \(x_2\) if...
  1. \(x_1\) is feasible while \(x_2\) is not,
  2. \(x_1\) and \(x_2\) both are infeasible but \(x_1\) has a smaller overall constraint violation, or
  3. \(x_1\) and \(x_2\) are both feasible but \(x_1\) dominates \(x_2\).

✔ Easy to implement
Constraint Domination [27–29] adapts the Pareto comparison to also consider constraints

A candidate solution $x_1$ is prefered in comparison to an element $x_2$ if...

1. $x_1$ is feasible while $x_2$ is not,
2. $x_1$ and $x_2$ both are infeasible but $x_1$ has a smaller overall constraint violation, or
3. $x_1$ and $x_2$ are both feasible but $x_1$ dominates $x_2$.

✔ Easy to implement
✔ Fits perfectly well to Pareto ranking and existing MOEAs
Constraint Domination adapts the Pareto comparison to also consider constraints.

A candidate solution $x_1$ is preferred in comparison to an element $x_2$ if...

1. $x_1$ is feasible while $x_2$ is not,
2. $x_1$ and $x_2$ both are infeasible but $x_1$ has a smaller overall constraint violation, or
3. $x_1$ and $x_2$ are both feasible but $x_1$ dominates $x_2$.

✔ Easy to implement
✔ Fits perfectly well to Pareto ranking and existing MOEAs
✔ Constraints can be of arbitrary nature
Section Outline

1 Introduction
2 Methods
3 Summary
Summary

- Constraints represent limitations on the possible solutions: require special treatment too
• Constraints represent limitations on the possible solutions: require special treatment too
• They are different from objectives: Objectives put “always” optimization pressure, whereas constraints only put pressure as long as they are not satisfied.
谢谢

Thank you

Thomas Weise [汤卫思]
tweise@hfuu.edu.cn
http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui, China


Bibliography III


