合肥學院
HEFEI UNIVERSITY

Metaheuristic Optimization

3. Exhaustive Enumeration

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

Outline

website

Baseline: Test all possible solutions

- Let us now start with some very simple ideas for optimization algorithms.

Baseline: Test all possible solutions

- Let us now start with some very simple ideas for optimization algorithms.
- First, we know that a computer can only represent finitely many values in its memory, because its memory is finite

Baseline: Test all possible solutions

- Let us now start with some very simple ideas for optimization algorithms.
- First, we know that a computer can only represent finitely many values in its memory, because its memory is finite
- In other words, *any* search space $\mathbb{G}$ or solution space $\mathbb{X}$ in practice is always finite, even if we solve numerical problems defined over $\mathbb{R}^n$

Baseline: Test all possible solutions

- Let us now start with some very simple ideas for optimization algorithms.
- First, we know that a computer can only represent finitely many values in its memory, because its memory is finite
- In other words, *any* search space $\mathbb{G}$ or solution space $\mathbb{X}$ in practice is always finite, even if we solve numerical problems defined over $\mathbb{R}^n$
- The very baseline, the most primitive optimization algorithm possible, would therefore simply enumerate all potentical candidate solutions and return the best one.

Exhaustive Enumeration

- *Exhaustive Enumeration* is a method for finite search spaces

Exhaustive Enumeration

- *Exhaustive Enumeration* is a method for finite search spaces
- It enumerates all possible solutions.

Exhaustive Enumeration

- *Exhaustive Enumeration* is a method for finite search spaces
- It enumerates all possible solutions.
- It will definitely find the global optimum $x^{\bigstar}$.

Exhaustive Enumeration

- *Exhaustive Enumeration* is a method for finite search spaces
- It enumerates all possible solutions.
- It will definitely find the global optimum $x^{\bigstar}$.
- Usually it has exponential complexity: the number of possible solutions usually rises exponentially with input size

Exhaustive Enumeration

- *Exhaustive Enumeration* is a method for finite search spaces
- It enumerates all possible solutions.
- It will definitely find the global optimum $x^{\bigstar}$.
- Usually it has exponential complexity: the number of possible solutions usually rises exponentially with input size
- If lower bound $\underline{y}$ for objective values is known, we can maybe stop earlier

Exhaustive Enumeration

- *Exhaustive Enumeration* is a method for finite search spaces
- It enumerates all possible solutions.
- It will definitely find the global optimum $x^{\bigstar}$.
- Usually it has exponential complexity: the number of possible solutions usually rises exponentially with input size
- If lower bound $\underline{y}$ for objective values is known, we can maybe stop earlier: If an element $x^{\bigstar}$ with $f(x^{\bigstar}) = \underline{y}$ is discovered, it is the global optimum and we can stop

Exhaustive Enumeration

- *Exhaustive Enumeration* is a method for finite search spaces
- It enumerates all possible solutions.
- It will definitely find the global optimum $x^{\star}$.
- Usually it has exponential complexity: the number of possible solutions usually rises exponentially with input size
- If lower bound $\underline{y}$ for objective values is known, we can maybe stop earlier: If an element $x^{\star}$ with $f(x^{\star}) = \underline{y}$ is discovered, it is the global optimum and we can stop
- In general, however, this method is not feasible because it takes too long

Exhaustive Enumeration

$\tilde{x} \longleftarrow \text{exhaustiveEnumeration} f \underline{y}$

Input: f: the objective/fitness function
Input: $\underline{y}$: the lowest possible objective value, $-\infty$ if unknown
Data: x: the current candidate solution
Output: $\tilde{x}$: the best currently known solution

begin
  $\tilde{x} \longleftarrow$ the first solution in the solution space
  while not all solutions checked $\wedge \left(f(\tilde{x}) > \underline{f}\right)$ **do**
    $x \longleftarrow$ next candidate solution
    if $f(\tilde{x}) > f(x)$ **then** $\tilde{x} \longleftarrow x$
  return $\tilde{x}$

Exhaustive Enumeration

$\tilde{x} \longleftarrow \text{exhaustiveEnumeration} f \underline{y}$

Input: f: the objective/fitness function
Input: $\underline{y}$: the lowest possible objective value, $-\infty$ if unknown
Data: x: the current candidate solution
Output: $\tilde{x}$: the best currently known solution

begin
    $\tilde{x} \longleftarrow$ the first solution in the solution space
    while not all solutions checked $\wedge \left(f(\tilde{x}) > \underline{f}\right)$ **do**
        $x \longleftarrow$ next candidate solution
        if $f(\tilde{x}) > f(x)$ **then** $\tilde{x} \longleftarrow x$
    return $\tilde{x}$

1. step-by-step test each point in the $\mathbb{G}$ (or all candidate solutions)

Exhaustive Enumeration

$\tilde{x} \longleftarrow \text{exhaustiveEnumeration} f \underline{y}$

Input: f: the objective/fitness function
Input: $\underline{y}$: the lowest possible objective value, $-\infty$ if unknown
Data: x: the current candidate solution
Output: $\tilde{x}$: the best currently known solution

begin
  $\tilde{x} \longleftarrow$ the first solution in the solution space
  while not all solutions checked $\wedge \left(f(\tilde{x}) > \underline{f}\right)$ **do**
    $x \longleftarrow$ next candidate solution
    if $f(\tilde{x}) > f(x)$ **then** $\tilde{x} \longleftarrow x$
  return $\tilde{x}$

1. step-by-step test each point in the $\mathbb{G}$ (or all candidate solutions)
2. remember the best candidate solution $\tilde{x}$ encountered

Exhaustive Enumeration

$\tilde{x} \longleftarrow \text{exhaustiveEnumeration} f \underline{y}$

Input: f: the objective/fitness function
Input: $\underline{y}$: the lowest possible objective value, $-\infty$ if unknown
Data: x: the current candidate solution
Output: $\tilde{x}$: the best currently known solution

begin
- $\tilde{x} \longleftarrow$ the first solution in the solution space
- **while** not all solutions checked $\wedge \left(f(\tilde{x}) > \underline{f}\right)$ **do**
  - $x \longleftarrow$ next candidate solution
  - **if** $f(\tilde{x}) > f(x)$ **then** $\tilde{x} \longleftarrow x$
- **return** $\tilde{x}$

1. step-by-step test each point in the $\mathbb{G}$ (or all candidate solutions)
2. remember the best candidate solution $\tilde{x}$ encountered
3. after all possible solutions have been tested or $\underline{y}$ is reached: $\tilde{x}$ must be global optimum

Summary

- Exhaustive enumeration is the most primitive way we can search in the solution space

Summary

- Exhaustive enumeration is the most primitive way we can search in the solution space
- It is an exact method that will eventually find the global optimum, but it will take way too long.

Summary

- Exhaustive enumeration is the most primitive way we can search in the solution space
- It is an exact method that will eventually find the global optimum, but it will take way too long.
- This method is never ever used in practice.

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

Caspar David Friedrich, “Der Wanderer über dem Nebelmeer”, 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog