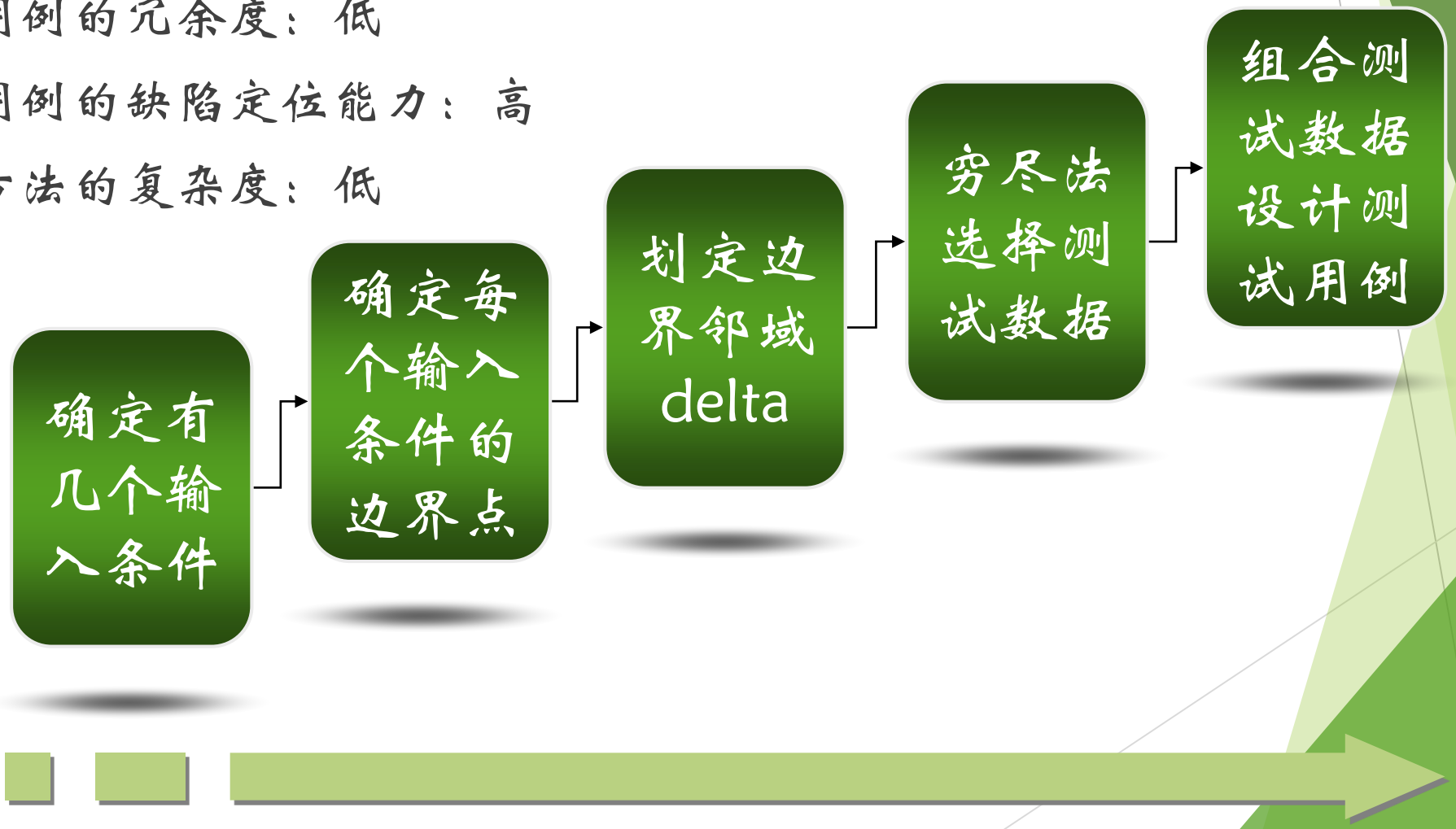
# 从输入看边界值测试（下）

- 测试方法的评价
  - 测试用例的数量：少
  - 测试用例的覆盖度：高
  - 测试用例的冗余度：低
  - 测试用例的缺陷定位能力：高
  - 测试方法的复杂度：低



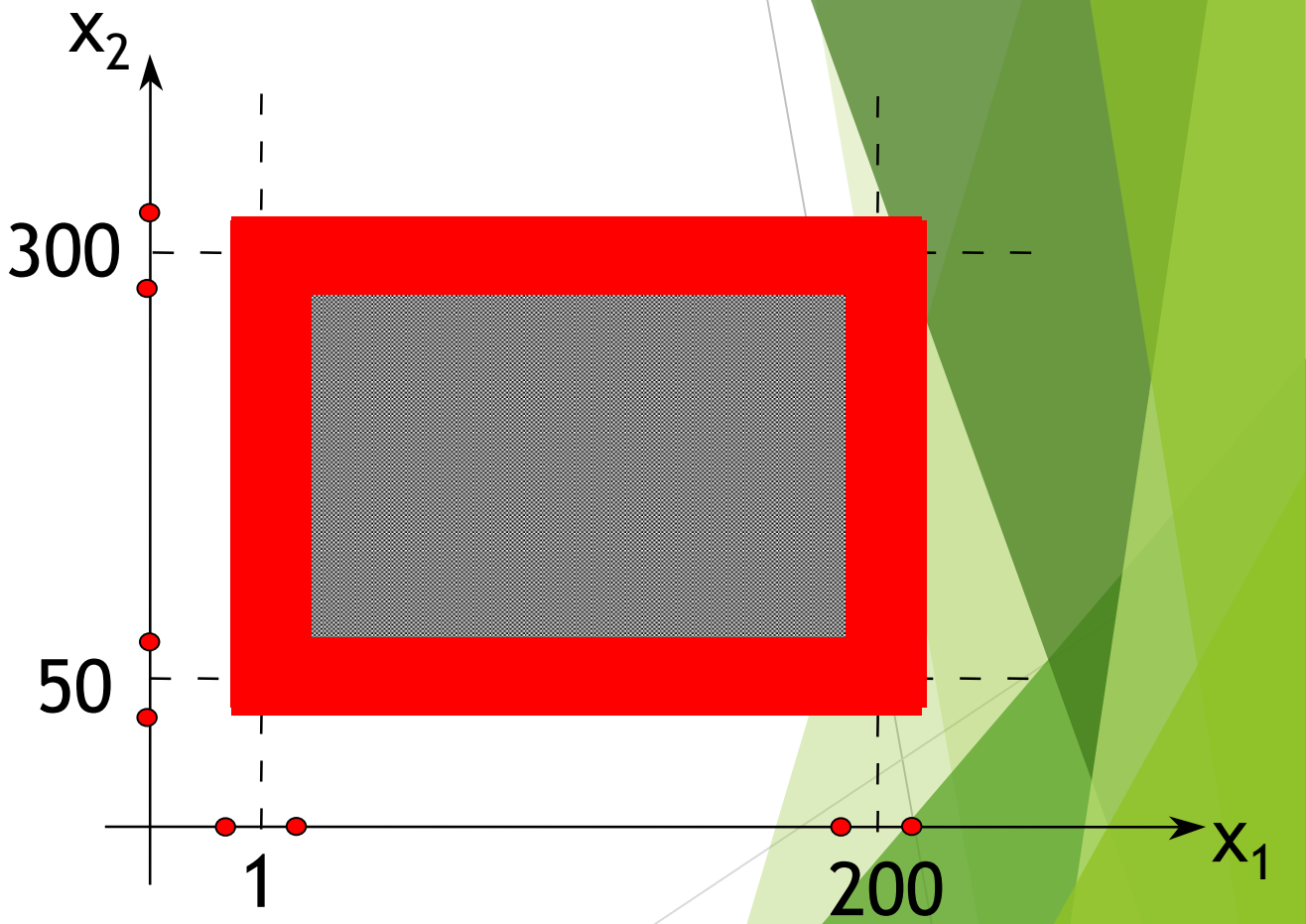
# 如何有效组织测试数据，设计测试用例？

- int Add(int x1, int x2)
  - 1 ≤ x1 ≤ 200
  - 50 ≤ x2 ≤ 300

| 输入条件 | x1 | x2 |
| --- | --- | --- |
| 边界点 | 1, 200 | 50, 300 |
| 边界测试数据 | 0,1,2,<br>199,200,201 | 49, 50, 51,<br>199, 300, 301 |

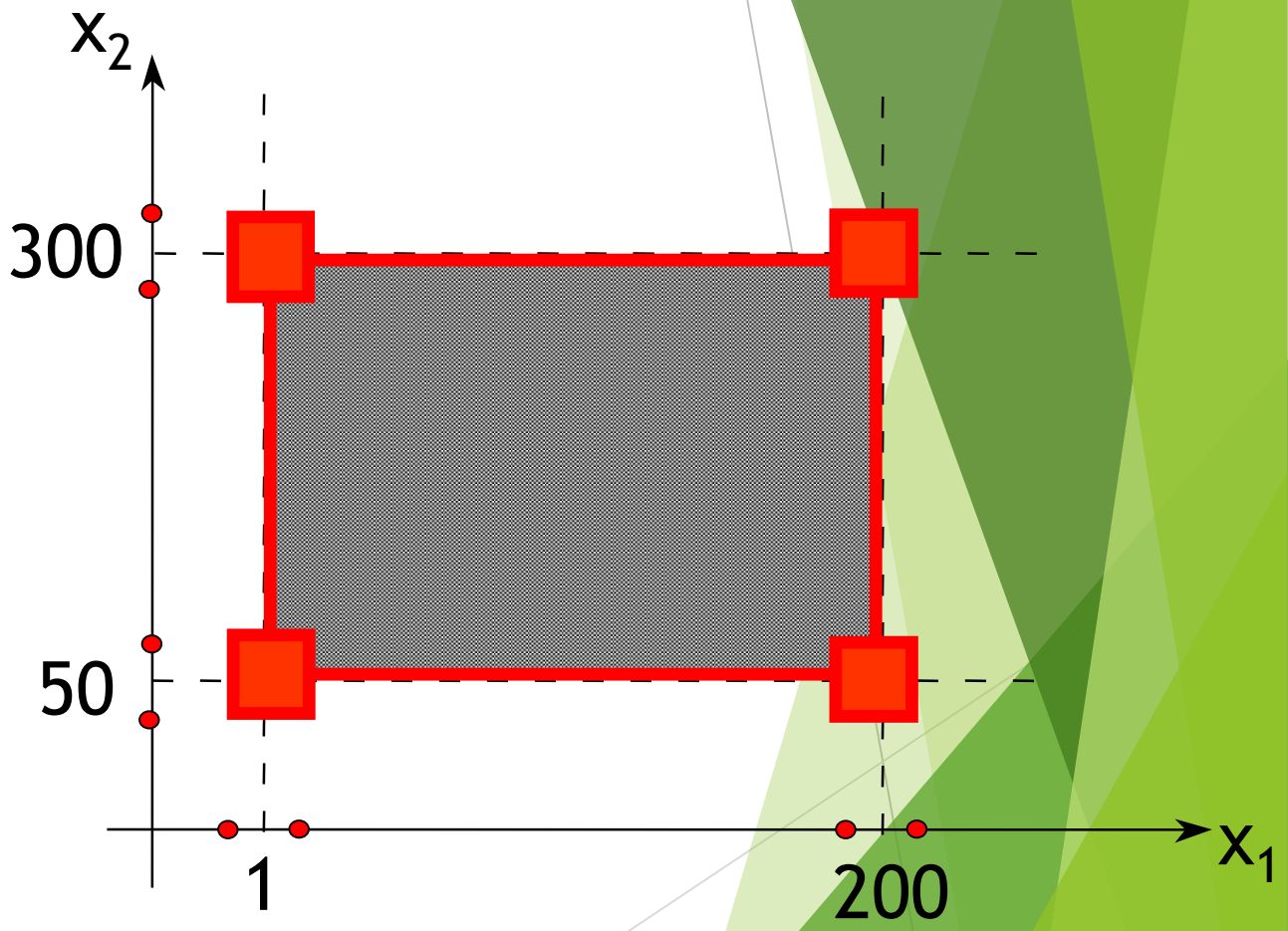
# 方案一：总体效果不好

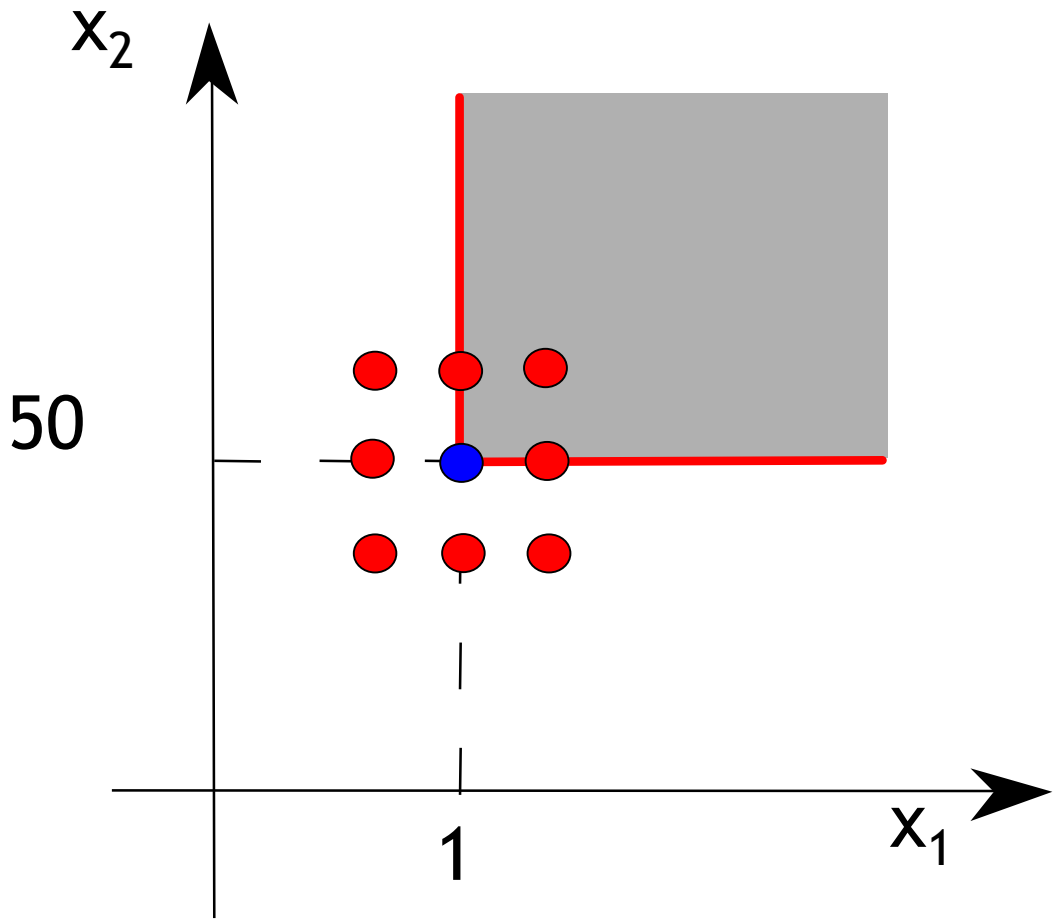
- 测试用例数目：2730
- 测试用例覆盖度：100%
- 测试用例冗余度：非常高
- 缺陷定位能力：低
- 测试方法复杂度：低



# 方案二

- 测试用例数目：36
- 测试用例覆盖度：100%
- 测试用例冗余度：不高
- 缺陷定位能力：低
- 测试方法复杂度：低

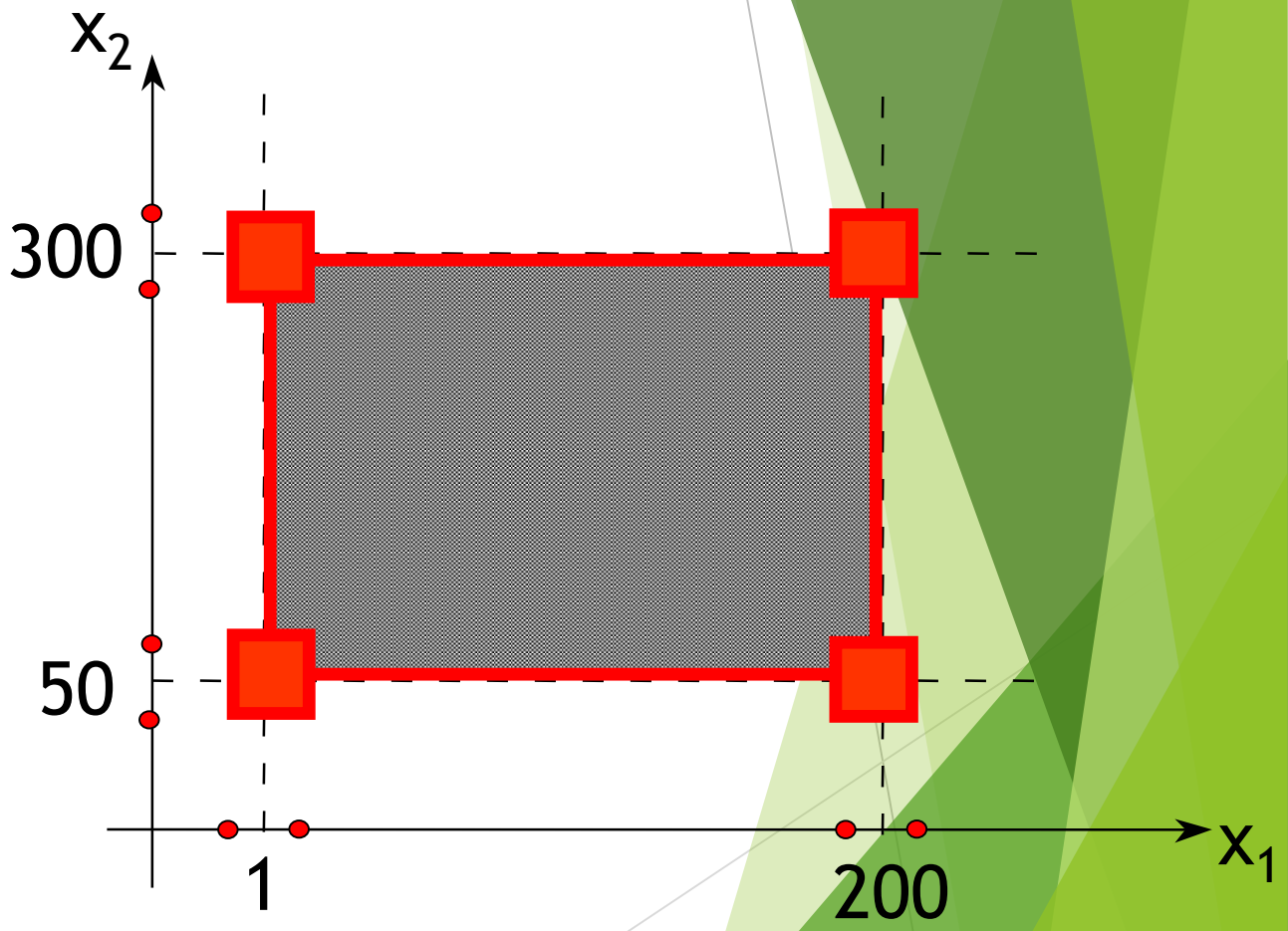



x2
50
1
x1

| 序号 | x1 | x2 | 预期输出 | 因边界点1有缺陷而导致的实际输出 | 因边界点50有缺陷而导致的实际输出 |
|---|---|---|---|---|---|
| 1 | 0 | 49 | -1 | -1 | -1 |
| 2 | 0 | 50 | -1 | -1 | -1 |
| 3 | 0 | 51 | -1 | -1 | -1 |
| 4 | 1 | 49 | -1 | -1 | -1 |
| 5 | 1 | 50 | 51 | -1 | -1 |
| 6 | 1 | 51 | 52 | -1 | 52 |
| 7 | 2 | 49 | -1 | -1 | -1 |
| 8 | 2 | 50 | 52 | 52 | -1 |
| 9 | 2 | 51 | 53 | 53 | 53 |

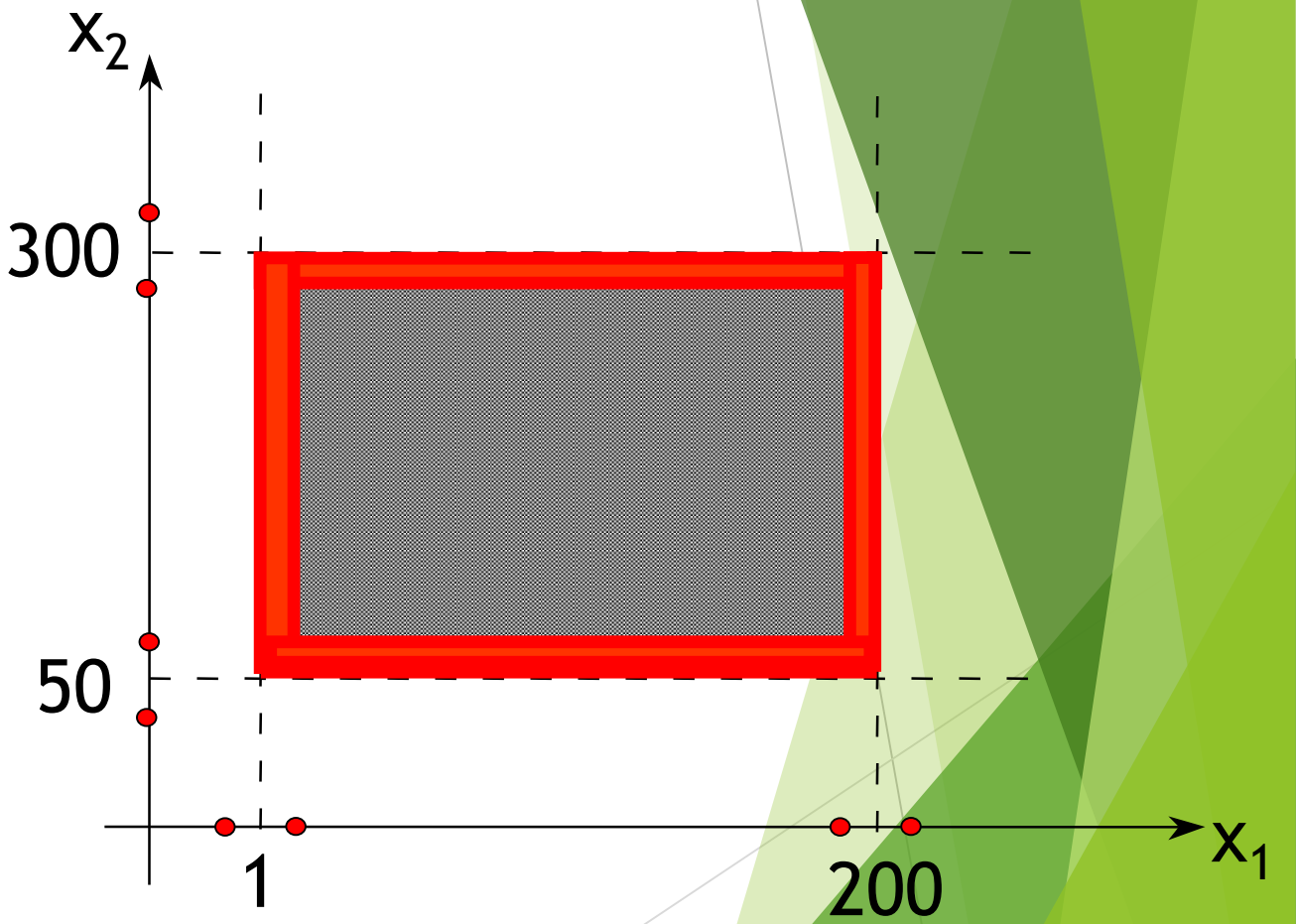
# 方案二

- 测试用例数目：36
- 测试用例覆盖度：100%
- 测试用例冗余度：不高
- 缺陷定位能力：低
- 测试方法复杂度：低



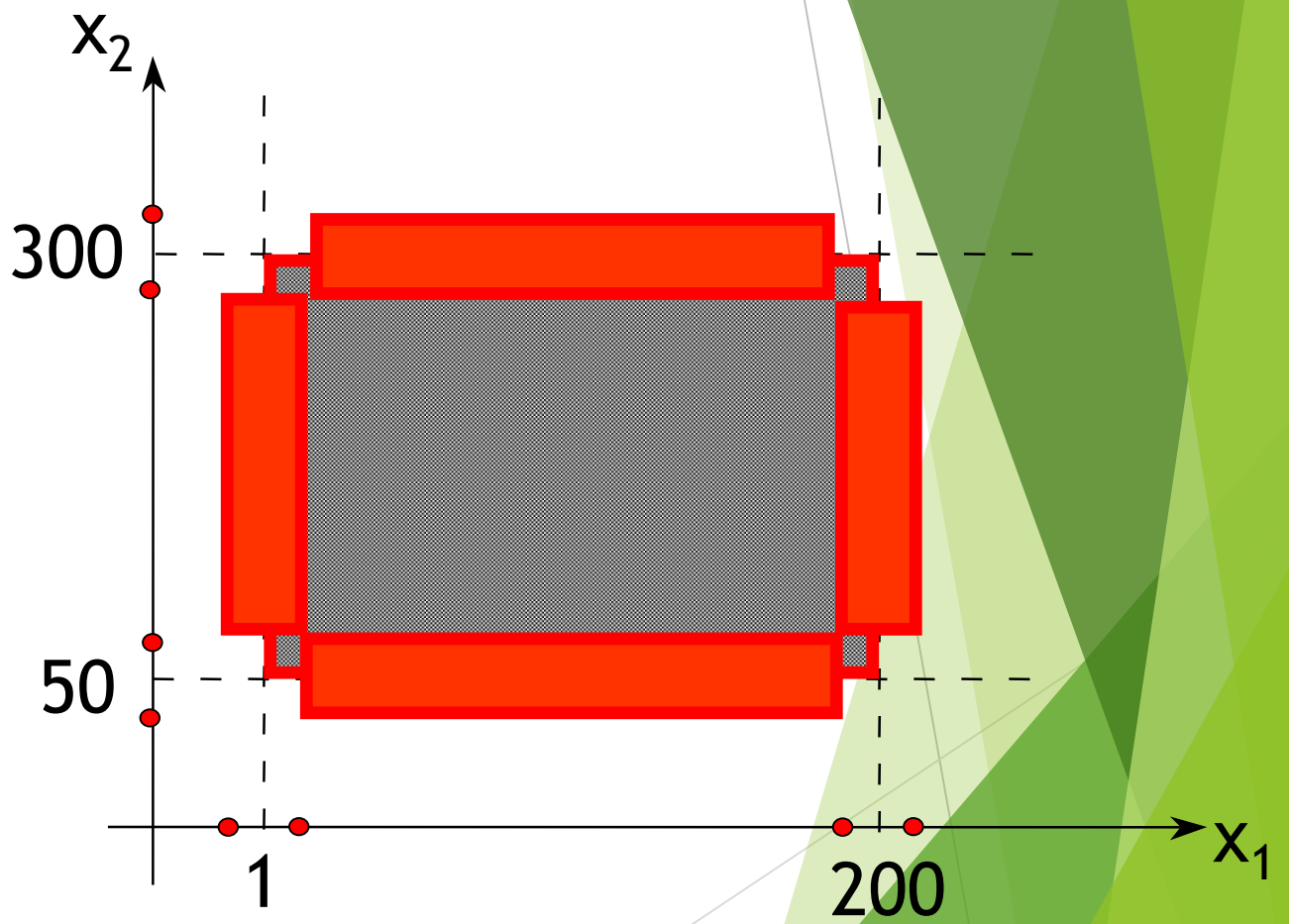
# 方案三

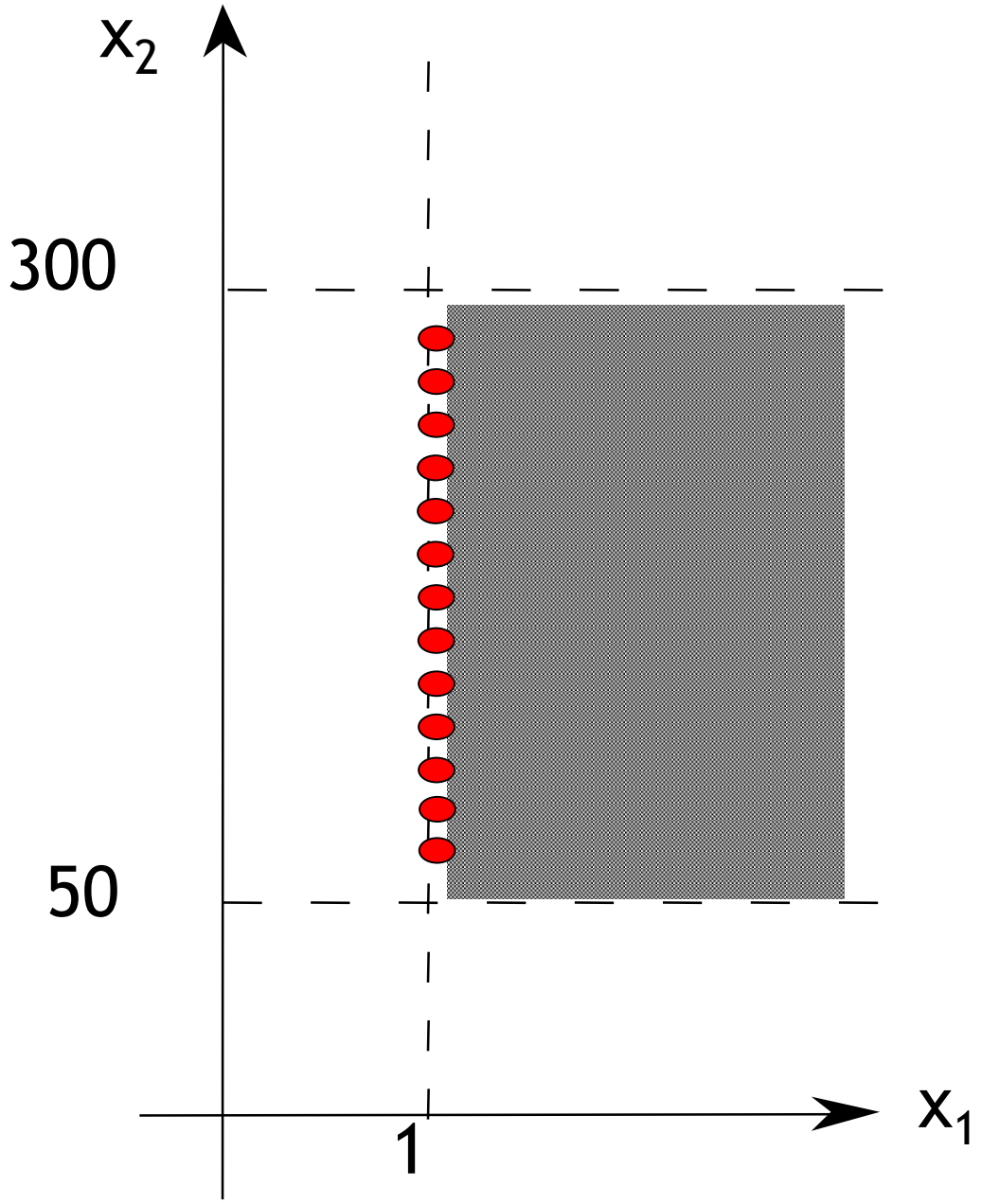
- 测试用例数目：1804
- 测试用例覆盖度：100%
- 测试用例冗余度：较高
- 缺陷定位能力：低
- 测试方法复杂度：低



# 方案四

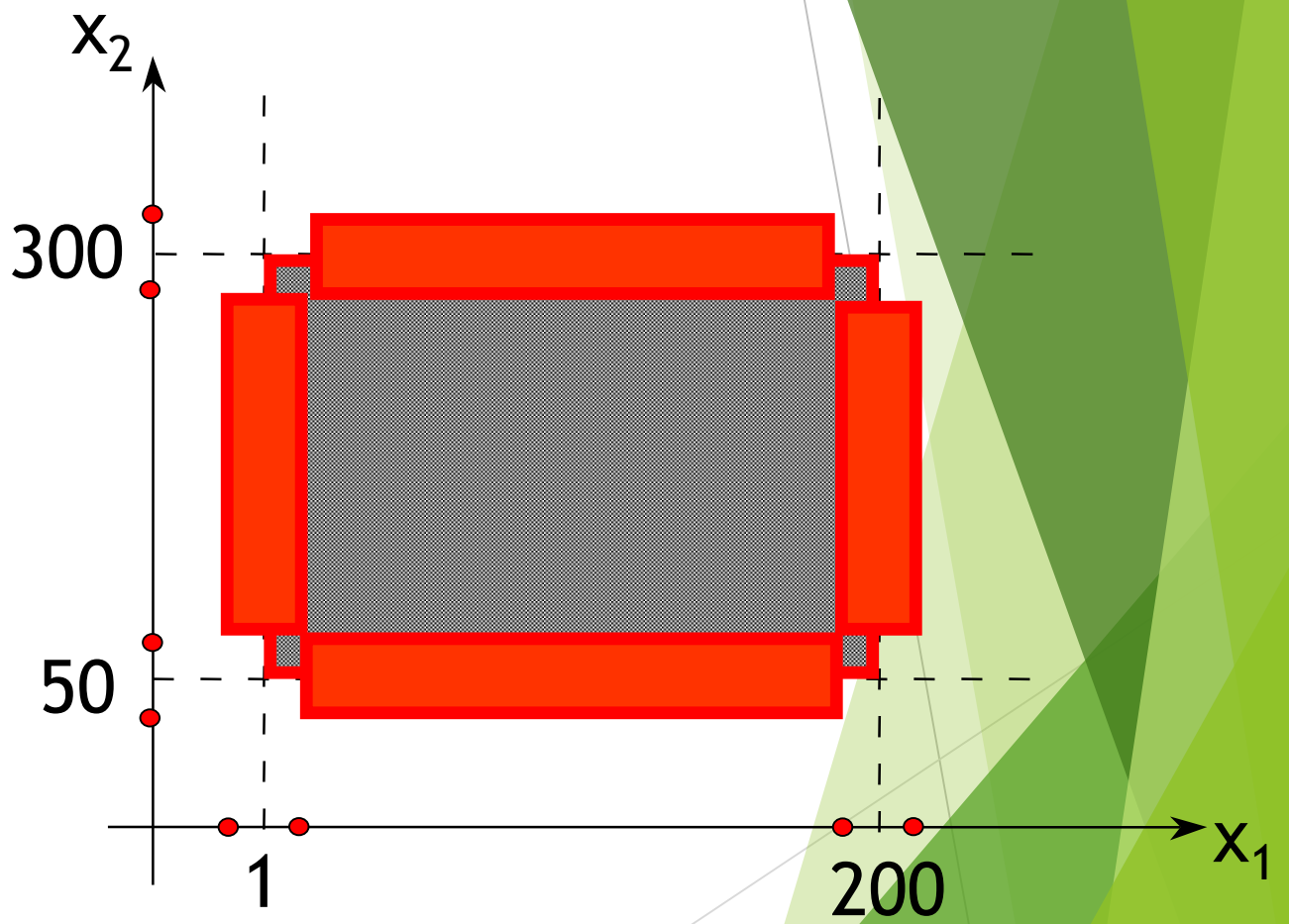
- 测试用例数目：2652
- 测试用例覆盖度：100%
- 测试用例冗余度：较高
- 缺陷定位能力：高
- 测试方法复杂度：低

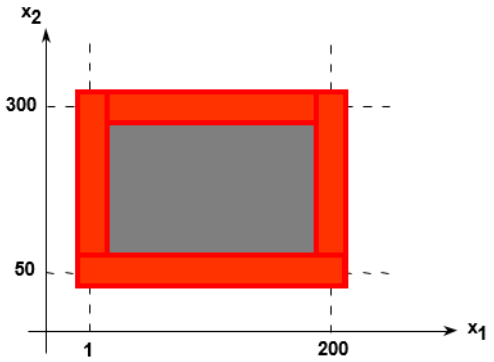


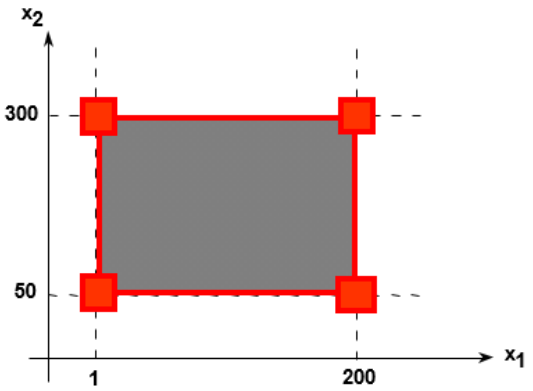

$x_2$
300
50
1
$x_1$

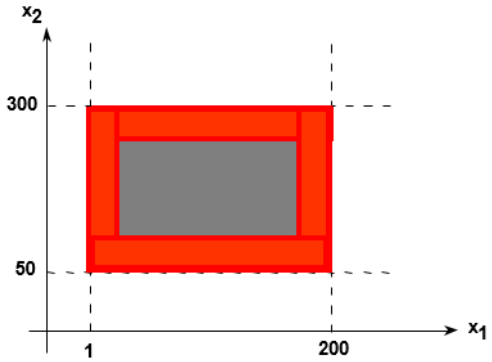
# 方案四

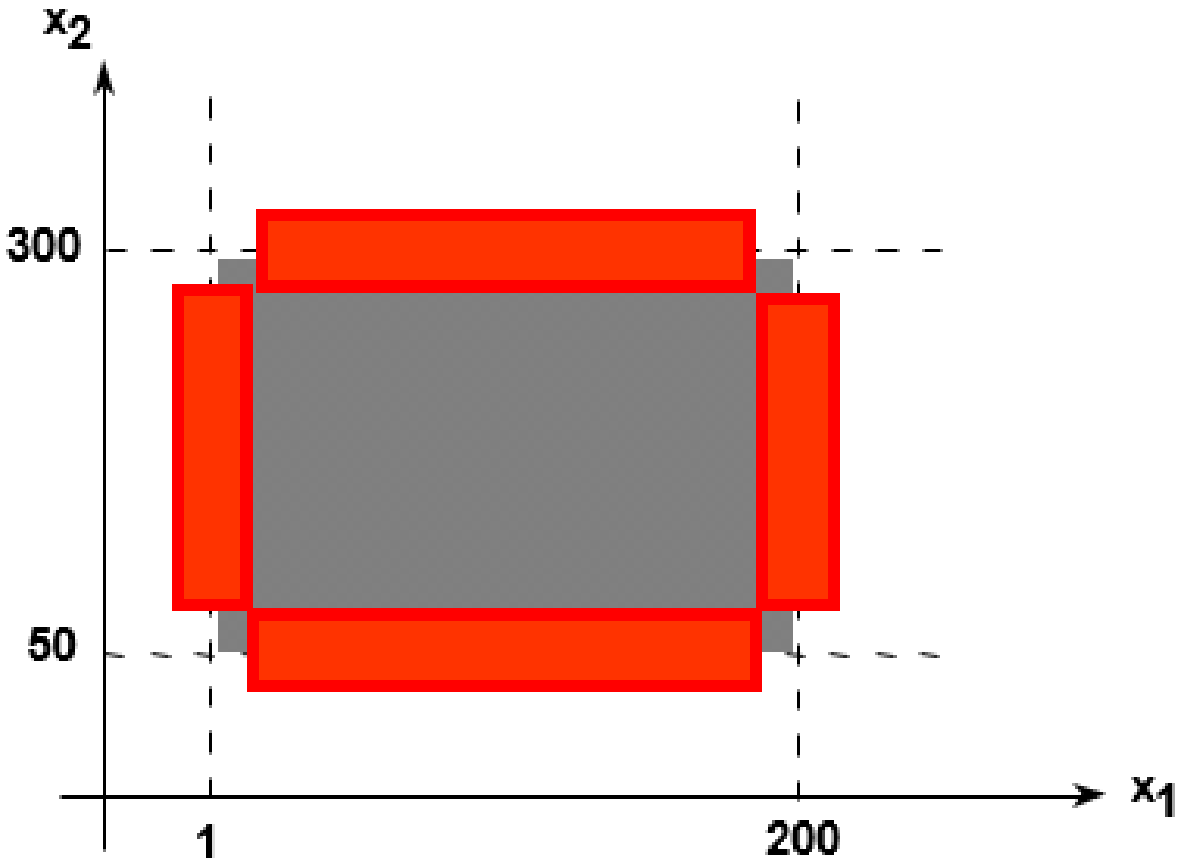
- 测试用例数目：2652
- 测试用例覆盖度：100%
- 测试用例冗余度：较高
- 缺陷定位能力：高
- 测试方法复杂度：低



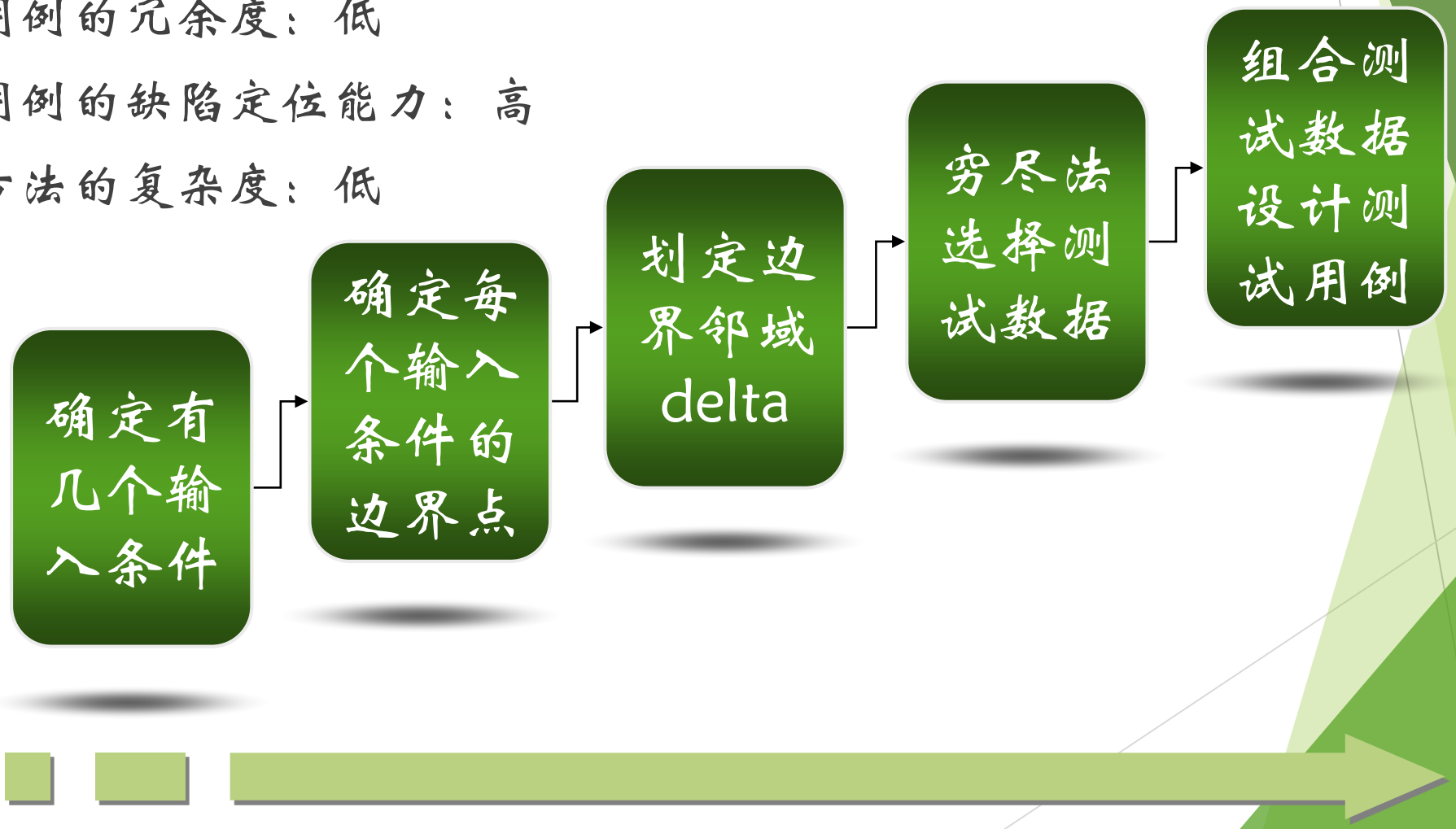

x2
300
50
1
200
x1


x2
300
50
1
200
x1

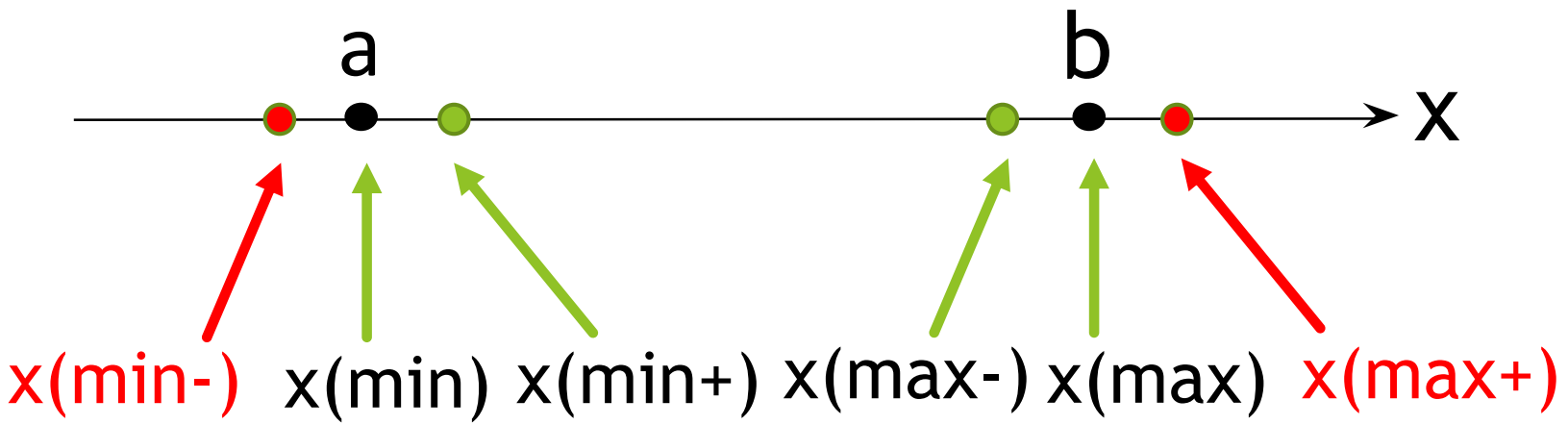

x2
300
50
1
200
x1


x2
300
50
1
200
x1

- 测试方法的评价
  - 测试用例的数量：少
  - 测试用例的覆盖度：高
  - 测试用例的冗余度：低
  - 测试用例的缺陷定位能力：高
  - 测试方法的复杂度：低



# 3. 如何选择测试数据?

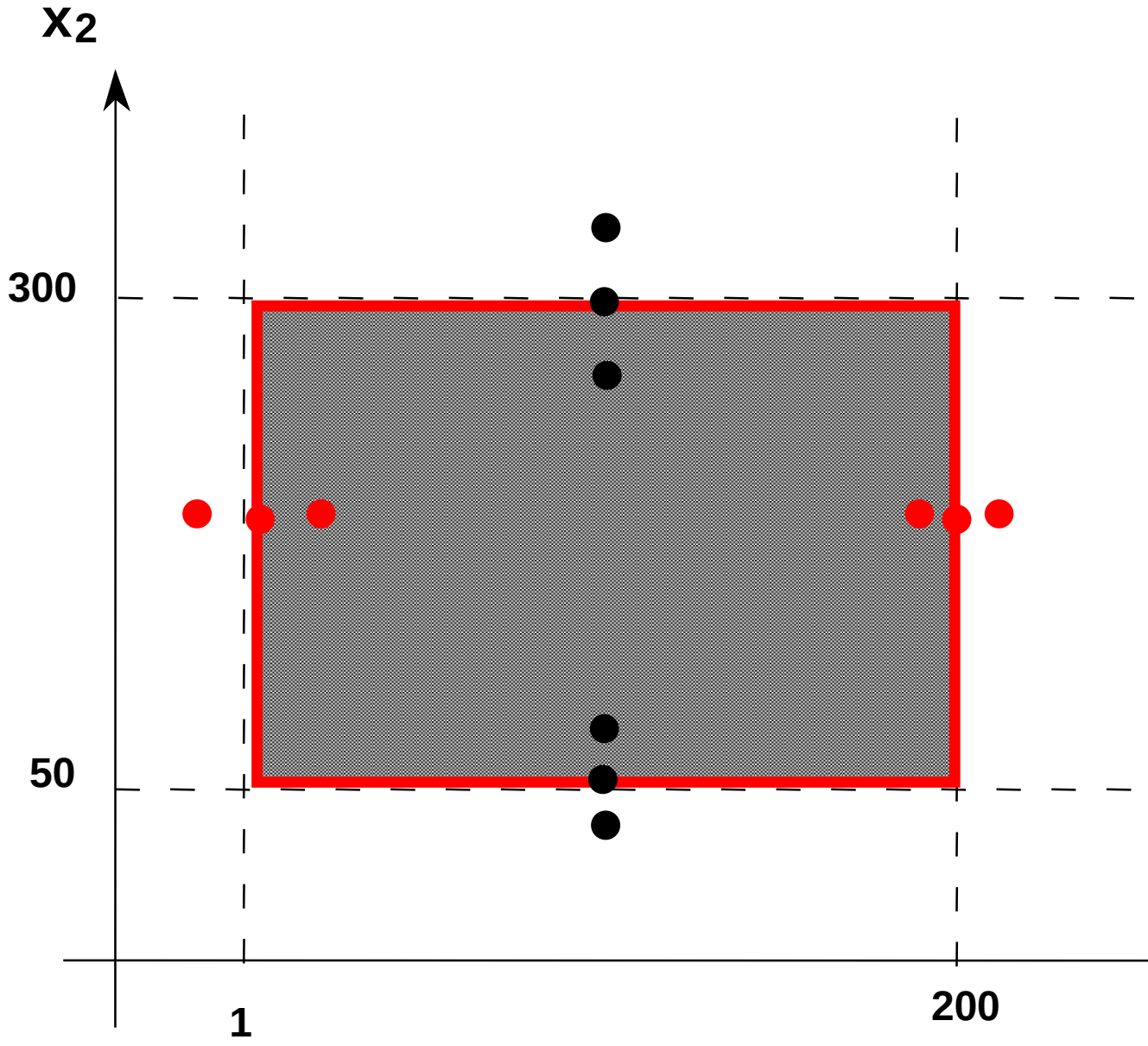



给定: $a \le x \le b$

x的边界点: a, b

邻域: [a-δ, a+δ], [b-δ, b+δ]

测试数据: a-δ, a, a+δ, b-δ, b, b+δ




- int Add(int x1, int x2)
  - 1 ≤ x1 ≤ 200
  - 50 ≤ x2 ≤ 300
- 规模：12
- 覆盖度：100%
- 冗余：无
- 缺陷定位：容易
- 复杂度：低


确定有几个输入条件

确定每个输入条件的边界点

划定边界邻域delta

每个边界对应3个测试数据

单边界设计测试用例