

The background is a dark blue-grey color. It is decorated with various geometric elements: teal circles of different sizes, some with white dotted patterns; white circles, some with teal dotted patterns; teal hexagons and polygons; and white dotted lines and patterns. The overall style is modern and abstract.

# Branch and Bound

NHÓM 1

**01.** Ý tưởng

**02.** Đặc điểm bài toán

**03.** Dạng phổ quát

**04.** Ưu/Nhược điểm

**05.** Ví dụ minh họa



## Ý TƯỞNG

**Cải tiến** giải thuật quay lui (backtracking)

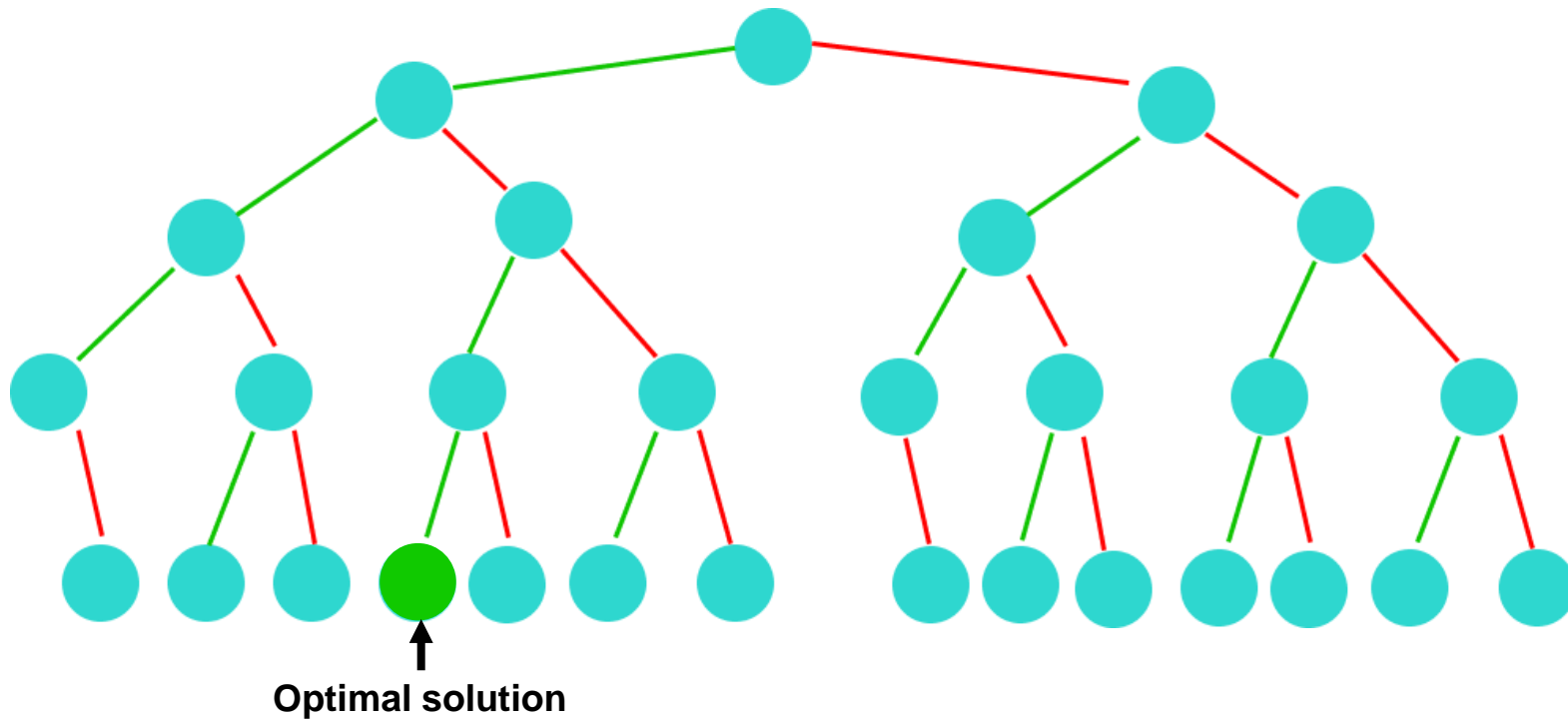
Là kỹ thuật xây dựng cây không gian trạng thái để tìm lời giải **tối ưu**. Sử dụng **giá trị cận** để cắt bớt các nhánh trong quá trình duyệt.

**Cận** của một nút là giá trị **tối đa/tối thiểu** mà các **nút con** của nó có thể đạt tới.



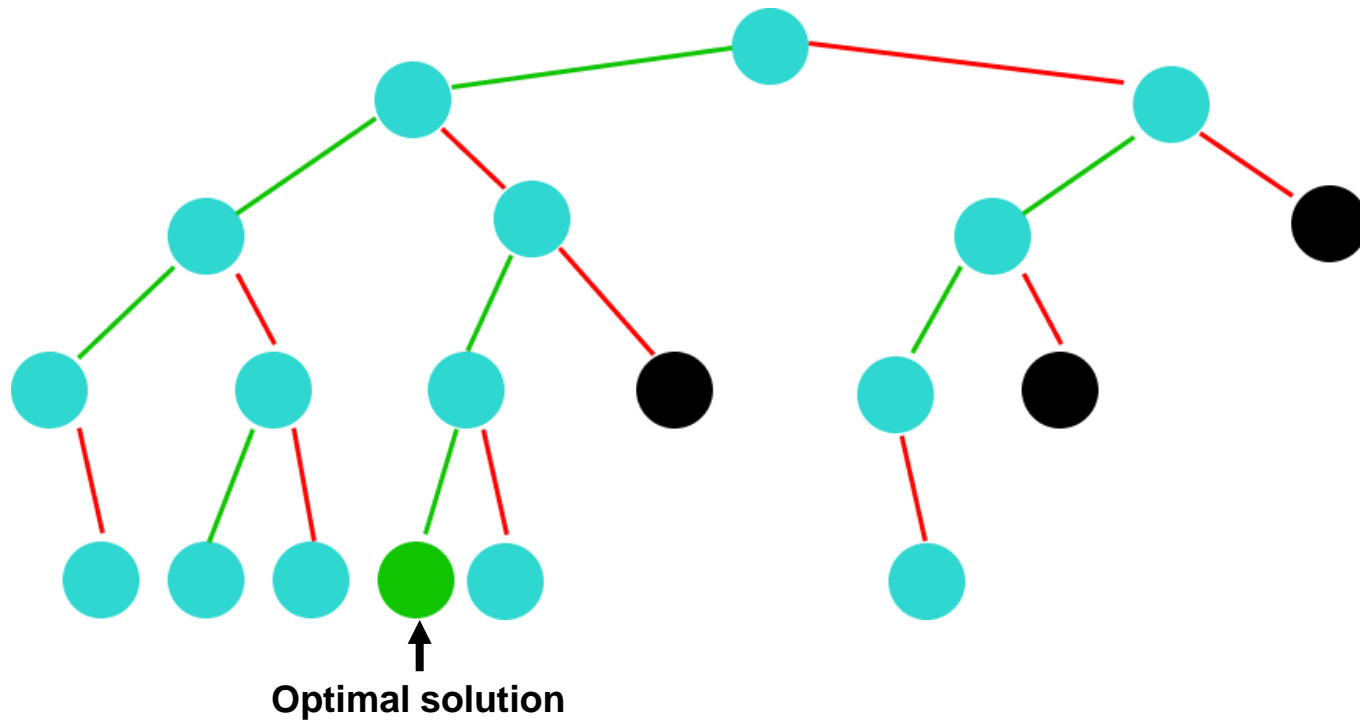
# Minh họa ý tưởng

## BACKTRACKING



# Minh họa ý tưởng

## BRANCH AND BOUND





# Dạng thuật toán phổ quát

## Khởi tạo:

- Hàm tính cận :  $f$
- Phương án tối ưu tạm thời :  $x$

## Cách chọn cận:

- Phương án tối thiểu
  - > cận trên
- Phương án tối đa
  - > cận dưới

## BnB(i)

```
{  
  For (j thuộc tập lựa chọn của của i)  
    If (j thỏa mãn các ràng buộc của bài toán)  
      {  
        Ghi nhận trạng thái nút hiện tại:  $x'$   
        If (  $i == n$  ) // trạng thái hiện tại là nút lá  
          Cập nhật lời giải tối ưu  
        Else  
          {  
            Xác định cận của nút hiện tại // tính  $f$   
            If (  $f(x') > x$  ) // hoặc <  
              BnB(i+1)  
          }  
        Trả về trạng thái cũ  
      }  
}
```

## ĐẶC ĐIỂM BÀI TOÁN

Nhánh cận thường được sử dụng để giải quyết những **bài toán tối ưu tổ hợp**.

Bài toán tối ưu tổ hợp là bài toán yêu cầu tìm ra một giải pháp **tối ưu** giữa các giải pháp khả dĩ trong một tập hữu hạn các giải pháp.

Từ khóa: **nhất**. ( có thể là nhỏ nhất / lớn nhất )



# ĐẶC ĐIỂM BÀI TOÁN

## *Các kỹ thuật tìm kiếm*

- 1.FIFO Branch and Bound
- 2.LIFO Branch and Bound
- 3.Least Cost-Branch and Bound





## ƯU ĐIỂM

Đảm bảo tìm được  
giải pháp tối ưu

Giảm được chi phí tìm kiếm  
so với backtracking

## NHƯỢC ĐIỂM

Phương pháp phụ thuộc  
vào hàm tính cận

Tốn nhiều thời gian do  
vẫn là duyệt toàn bộ  
trạng thái có thể



# Ví dụ minh họa

## ***Bài toán Knapsack:***

item	weight	value
1	4	\$40
2	7	\$42
3	5	\$25
4	3	\$12

Khả năng chứa của balo : 10



# Ví dụ minh họa

## ***Bài toán Knapsack:***

item	weight	value
1	4	\$40
2	7	\$42
3	5	\$25
4	3	\$12

$$v_1/w_1 \geq v_2/w_2 \geq \dots \geq v_n/w_n.$$

Khả năng chứa của balo : 10

# Ví dụ minh họa

## Bài toán Knapsack:

item	weight	value	$\frac{\text{value}}{\text{weight}}$
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

Sắp xếp giảm dần theo value/weight

$$v_1/w_1 \geq v_2/w_2 \geq \dots \geq v_n/w_n.$$

Khả năng chứa của balo : 10

# Ví dụ minh họa

## Bài toán Knapsack:

item	weight	value	$\frac{\text{value}}{\text{weight}}$
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

Khả năng chứa của balo : 10

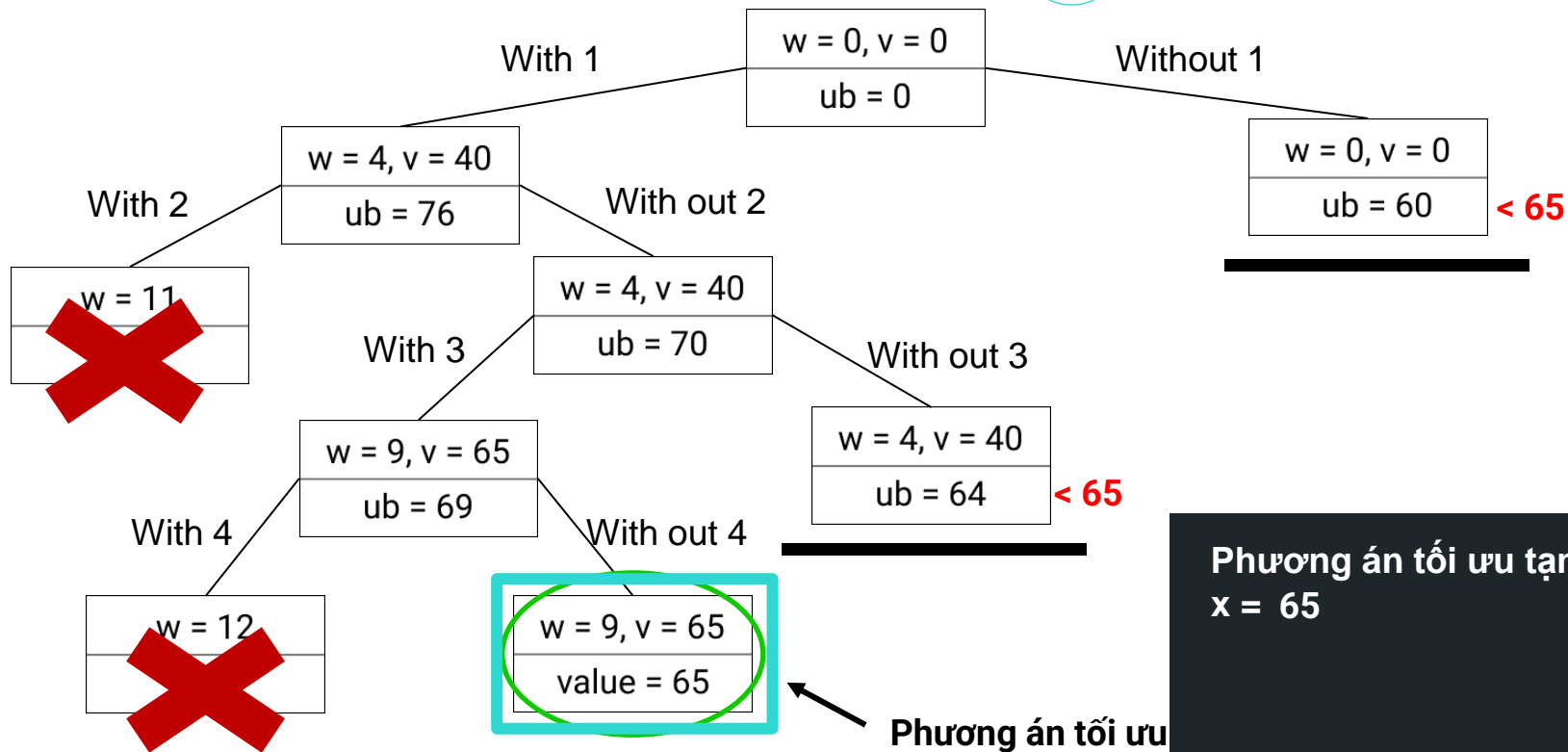
## Hàm tính cận:

$$ub = v + (W - w)(v_{i+1}/w_{i+1}).$$

Giá trị hiện tại

Cân nặng hiện tại

# Ví dụ minh họa



Phương án tối ưu tạm thời:  
 $x = 65$

Phương án tối ưu



# Thanks!

---

