

Introduction to Artificial Intelligence in Games

Lecture 5

Informed Search Algorithms: Applications

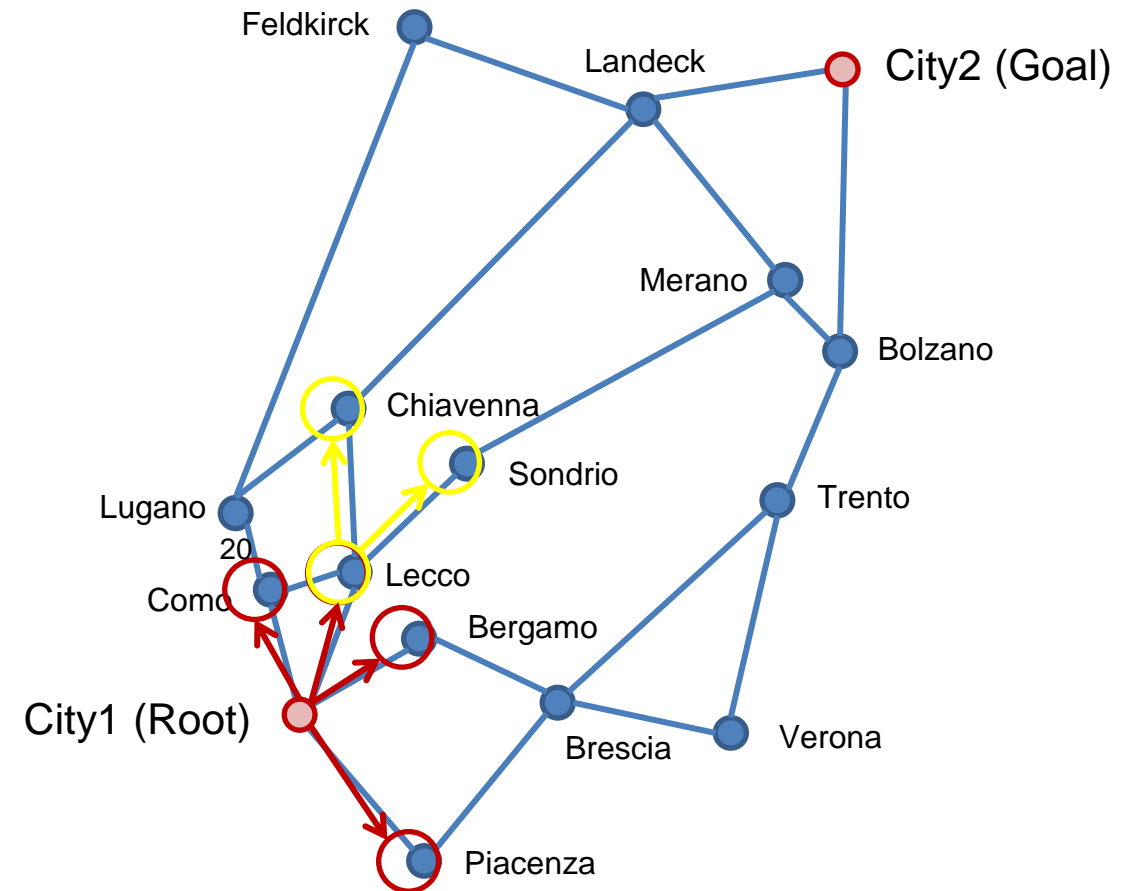
- Get the most cost-effective route to a destination location on a map.
- Suppose we want to get the best path from City1 to City2
- Other cities that we may traverse are the nodes on the graph
- The actual distances between every pair of cities (nodes) are known.
- The straight line distance between cities and the goal city is also known (by Google maps)



Informed Search Algorithms: Applications

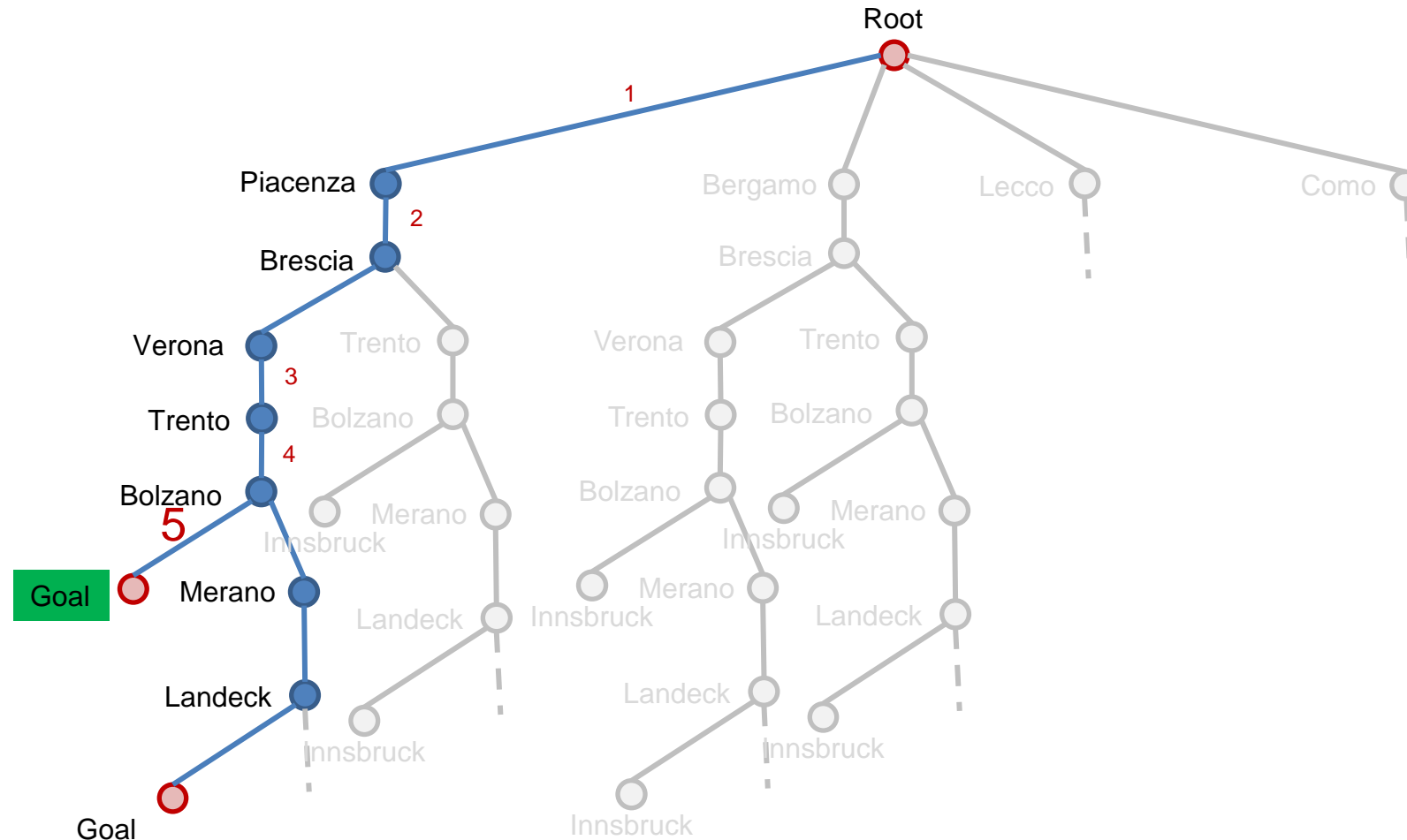
Graph Representation:

- Root node: City1
- Goal node: City2
- Nodes: other cities
- $g(n)$: cumulative distance between node n to the root node
- $h(n)$: heuristic value (straight-line distance) between the node n to the goal node
- Apply search algorithm such as A* to determine the best path from the root to the goal node



Informed Search Algorithms: Applications

Building Search Tree:

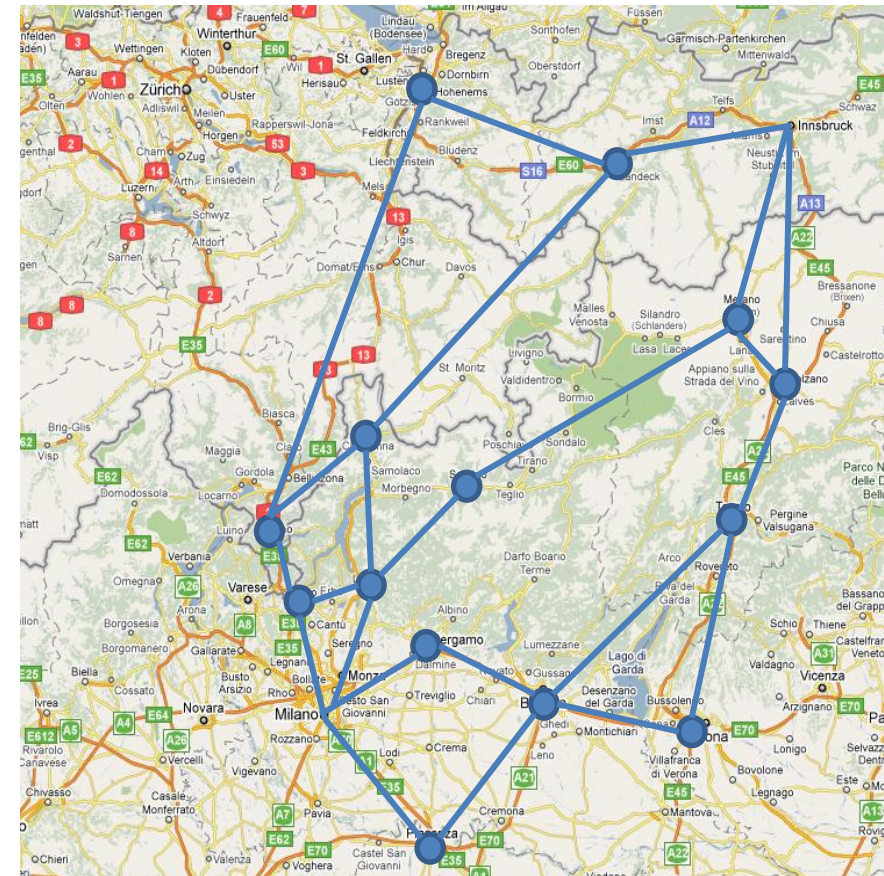


Travelling Salesman Problem

- Travelling Salesman Problem (TSP): Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible route that the salesman visits every city exactly once and returns to the starting point.

Solution Steps:

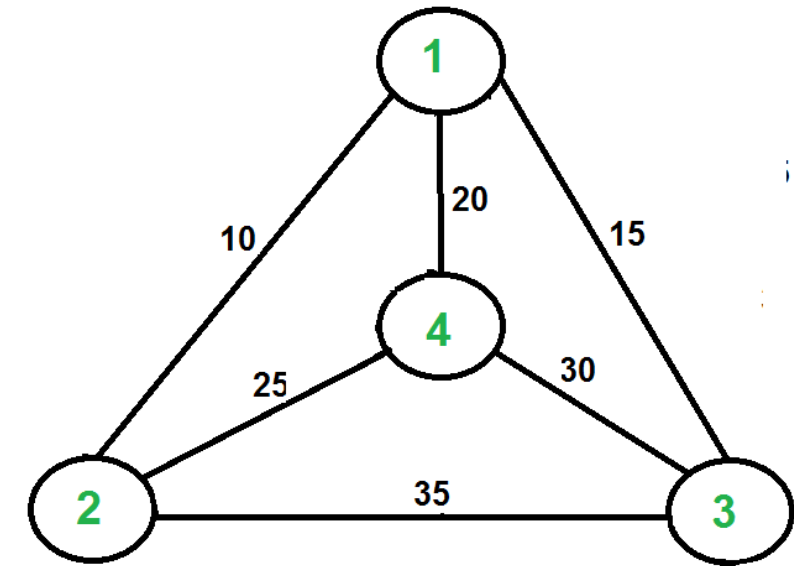
- Suppose there are n cities needed to be traversed.
- For each city:
 - Consider the city as the starting and ending point.
 - Generate all $(n - 1)!$ permutations of cities.
 - Calculate cost of every permutation.
 - Repeat, starting with another city
- Return the permutation with minimum cost.
- Number of all permutations in this problem is $n!$



Travelling Salesman Problem

Consider the following graph:

- No. of nodes (cities)=4.
- No. of permutations = $4! = 24$



Starting with node 1

No.	Permutation	Cost
1	1-2-3-4-1	95
2	1-2-4-3-1	80
3	1-3-2-4-1	95
4	1-3-4-2-1	80
5	1-4-2-3-1	95
6	1-4-3-2-1	95

Starting with node 2

No.	Permutation	Cost
1	2-1-3-4-2	80
2	2-1-4-3-2	95
3	2-3-4-1-2	95
4	2-3-1-4-2	95
5	2-4-1-3-2	95
6	2-4-3-1-2	80

Travelling Salesman Problem

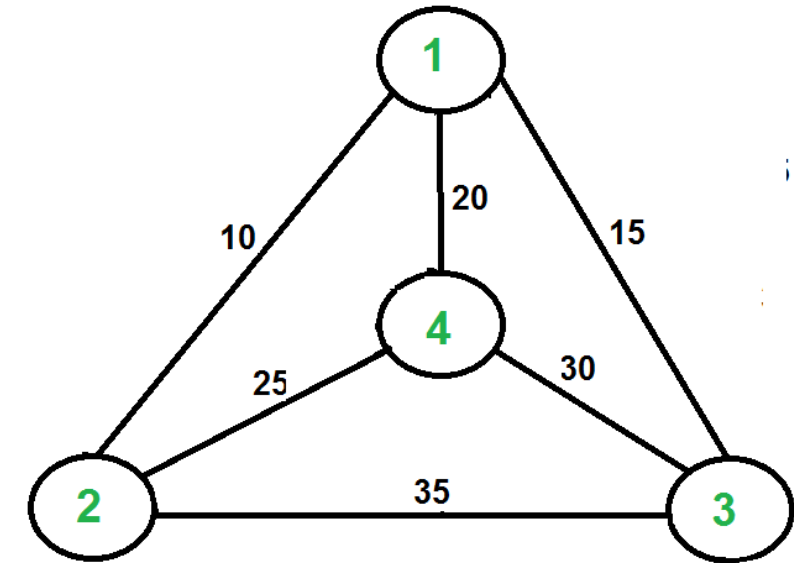
Cont.

Starting with node 3

No.	Permutation	Cost
1	3-1-2-4-3	80
2	3-1-4-2-3	95
3	3-2-1-4-3	95
4	3-2-4-1-3	95
5	3-4-1-2-3	95
6	3-4-2-1-3	80

Starting with node 4

No.	Permutation	Cost
1	4-1-2-3-4	95
2	4-1-3-2-4	95
3	4-2-1-3-4	80
4	4-2-3-1-4	95
5	4-3-1-2-4	80
6	4-3-2-1-4	95



- Thus, among 24 possible paths, there are 8 paths which have the lowest cost which is 80.

Travelling Salesman Problem

Time Complexity of the travelling salesman problem:

- The number of possible paths to travel along n cities and visit every city exactly once and returns to the starting point is $n!$
- Suppose $n = 20$ cities, no. of paths = $20! = 2.43 \times 10^{18}$
- Suppose each path is examined and its cost is calculated in 1 second
- Thus, the total time needed to explore all paths is 2.43×10^{18} seconds
- This is equivalent to about 7.7×10^{10} years.
- This problem is non-deterministic

AO* Search Algorithm

Problem Reduction (Decomposition) :

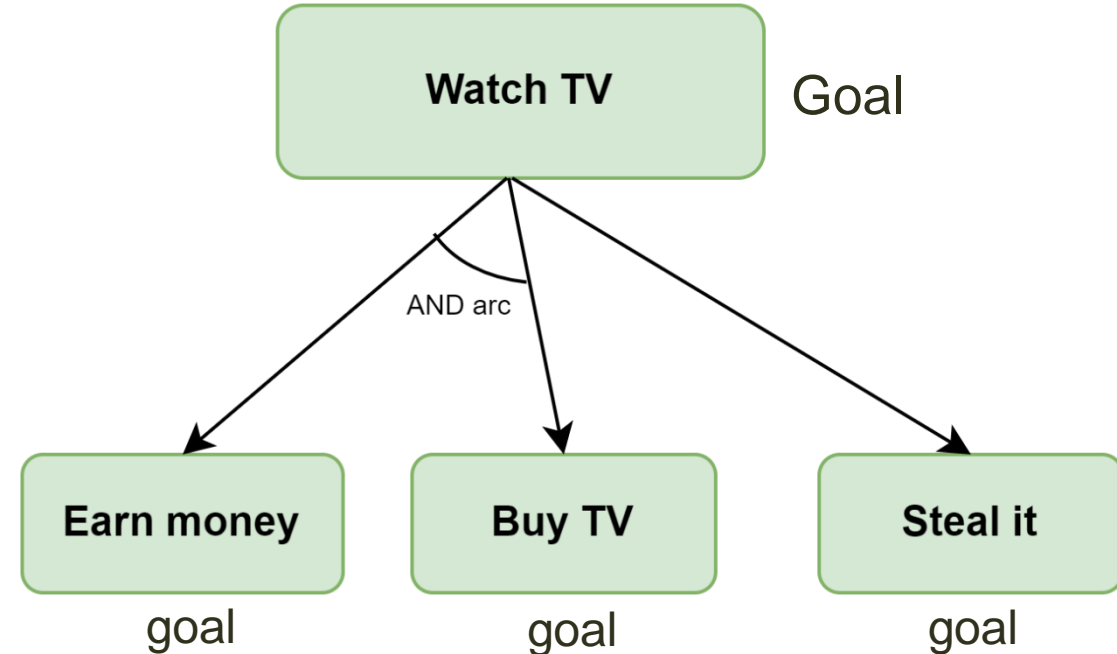
- AO* is informed search algorithm, work based on heuristic
- We already know about the divide and conquer strategy, a solution to a problem can be obtained by decomposing it into smaller sub-problems.
- Each of this sub-problem can then be solved to get its sub solution.
- These sub solutions can then recombined to get a solution as a whole. That is called is **Problem Reduction**. AND-OR graphs or AND - OR trees are used for representing the solution.
- AND-OR graph is used to represent various kind of complex problem solutions.
- AO* search algorithm is based on AND-OR graph so ,it is called AO* search algorithm

AND-OR Tree

- AND-OR graph is useful for representing the solution of problems that can be solved by decomposing them into a set of smaller problems, all of which must be then solved.

Example

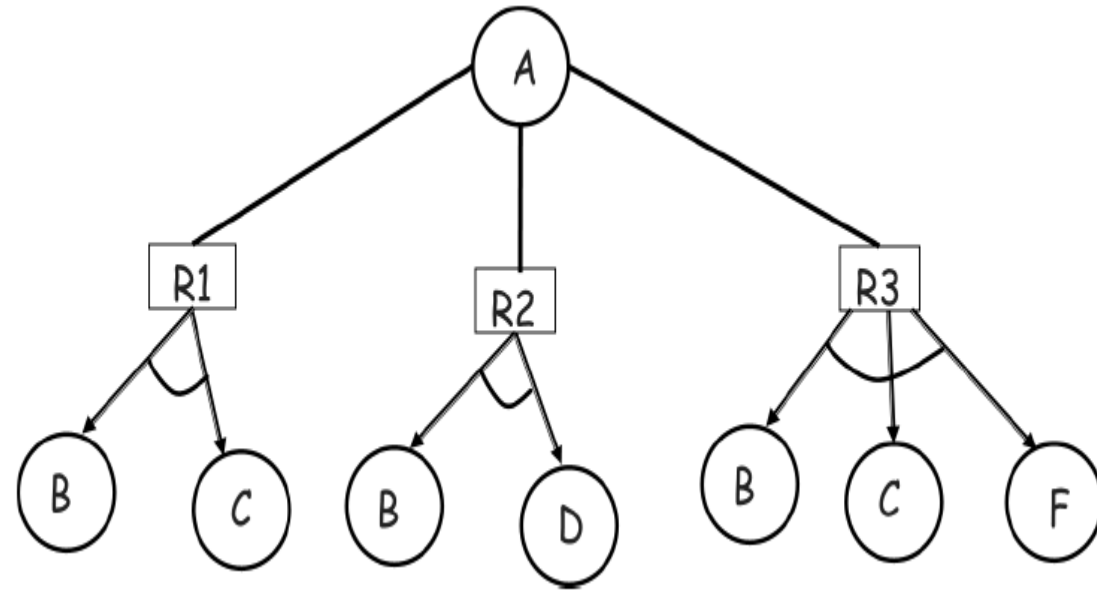
- Problem: Watch TV
- This problem can be solved by dividing it into smaller problems and then solving these smaller problems separately.
- If the sub-problems are solvable, then we can solve the main problem



AND-OR Tree

Example: represent the following graph

- A is solved if R1 OR R2 OR R3 is solved
- R1 is solved if B AND C are solved
- R2 is solved if B AND D are solved
- R3 is solved if B AND C AND F are solved



AND-OR Tree

Example: draw the AND-OR search tree for the following rules:

P if Q and R

P if S

Q if T

Q if U

AND-OR Tree

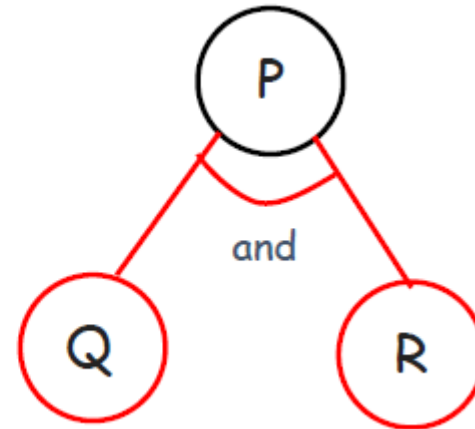
Solution

P if Q and R

P if S

Q if T

Q if U

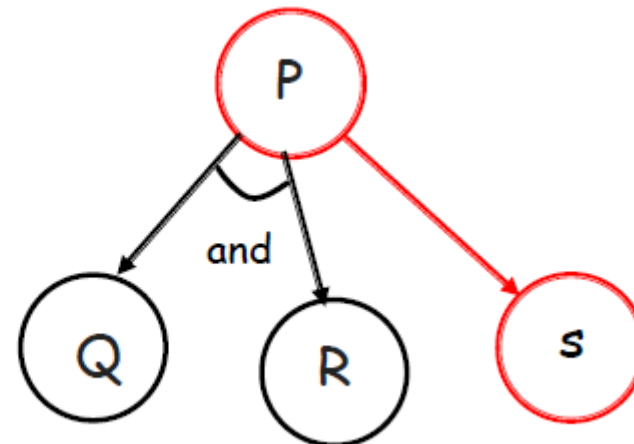


P if Q and R

P if S

Q if T

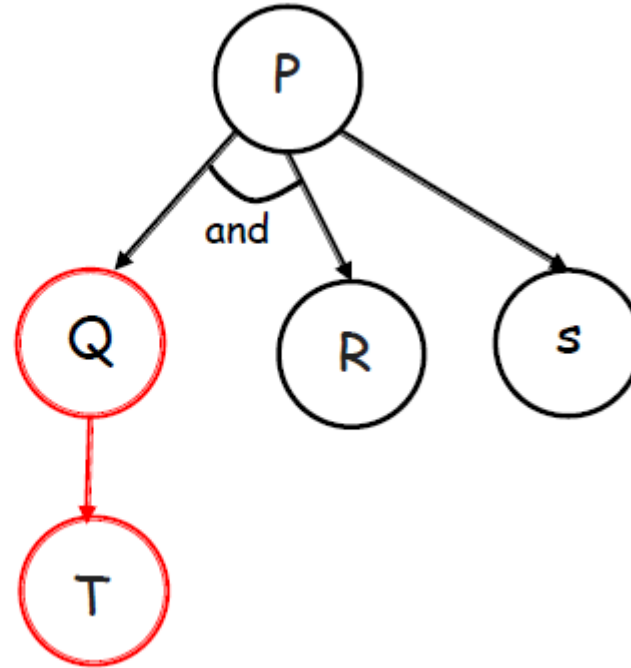
Q if U



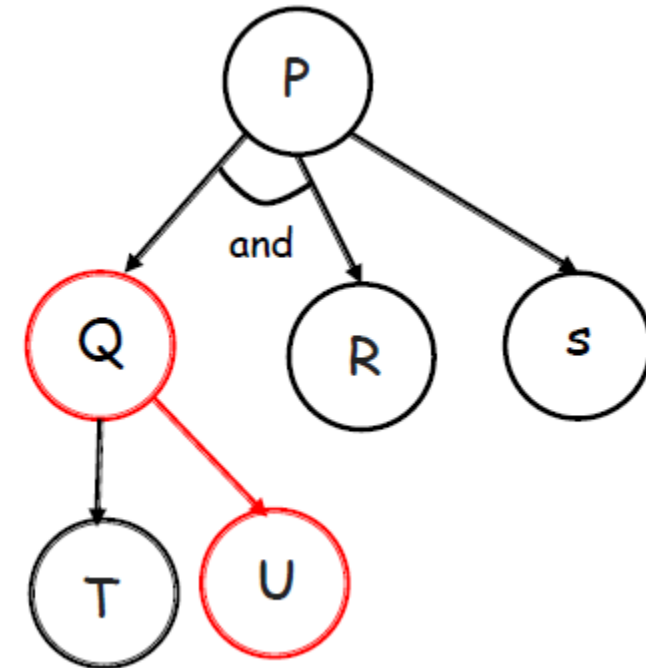
AND-OR Tree

Solution

P if Q and R
P if S
Q if T
Q if U



P if Q and R
P if S
Q if T
Q if U



AO* Search Algorithm

Example

- Represent the AND-OR search tree and find the solution for the problem.
- Main problem: P_0
- Solvable sub-problems are:
 $P_1, P_5, P_6, P_{13}, P_{14}, P_{15}$
- Heuristic value for solvable problems = 0

Operation	Cost for Op.	Heuristic value
Op ₁ : $P_0 \Rightarrow p_1 \& p_2$	$K(op_1)=5$	$h(p_2)=50$
Op ₂ : $P_0 \Rightarrow p_3$	$k(op_2)=19$	$h(p_3)=28$
Op ₃ : $P_0 \Rightarrow p_4 \& p_5$	$K(op_3)=8$	$h(p_4)=40$
Op ₄ : $P_2 \Rightarrow p_{15} \& p_{10}$	$K(op_4)=5$	$h(p_7)=30$
Op ₅ : $P_3 \Rightarrow p_6 \& p_7$	$K(op_5)=20$	$h(p_8)=22$
Op ₆ : $P_3 \Rightarrow p_8 \& p_9$	$K(op_6)=10$	$h(p_9)=20$
Op ₇ : $P_4 \Rightarrow p_{10}$	$K(op_7)=10$	$h(p_{10})=30$
Op ₈ : $P_4 \Rightarrow p_{11} \& p_{12}$	$K(op_8)=20$	$h(p_{11})=15$
Op ₉ : $P_{10} \Rightarrow p_{13} \& p_{14} \& p_6$	$k(op_9)=45$	$h(p_{12})=15$

AO* Search Algorithm

Solution: Search Tree

Op₁: $P_0 \Rightarrow p_1 \& p_2$

Op₂: $P_0 \Rightarrow p_3$

Op₃: $P_0 \Rightarrow p_4 \& p_5$

$K(op_1)=5$

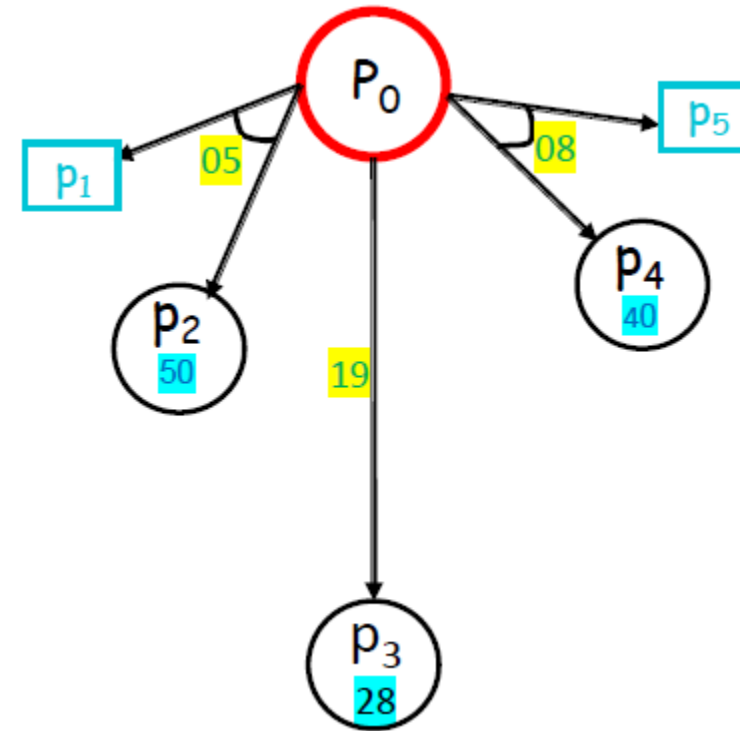
$k(op_2)=19$

$K(op_3)=8$

$h(p_2)=50$

$h(p_3)=28$

$h(p_4)=40$



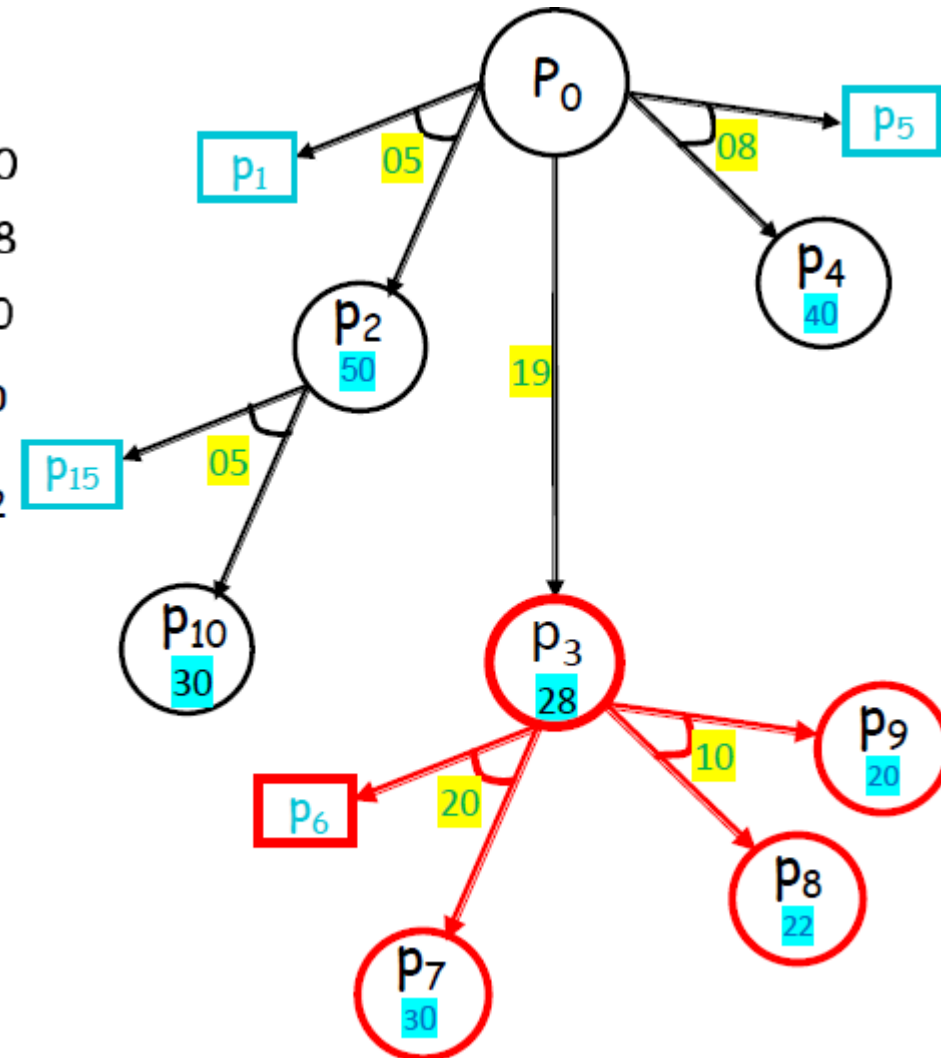
Solution: Search Tree Cont.

$$h(p_2)=50$$
$$h(p_3)=28$$
$$h(p_4)=40$$
$$h(p_{10})=30$$


AO* Search Algorithm

Solution: Search Tree Cont.

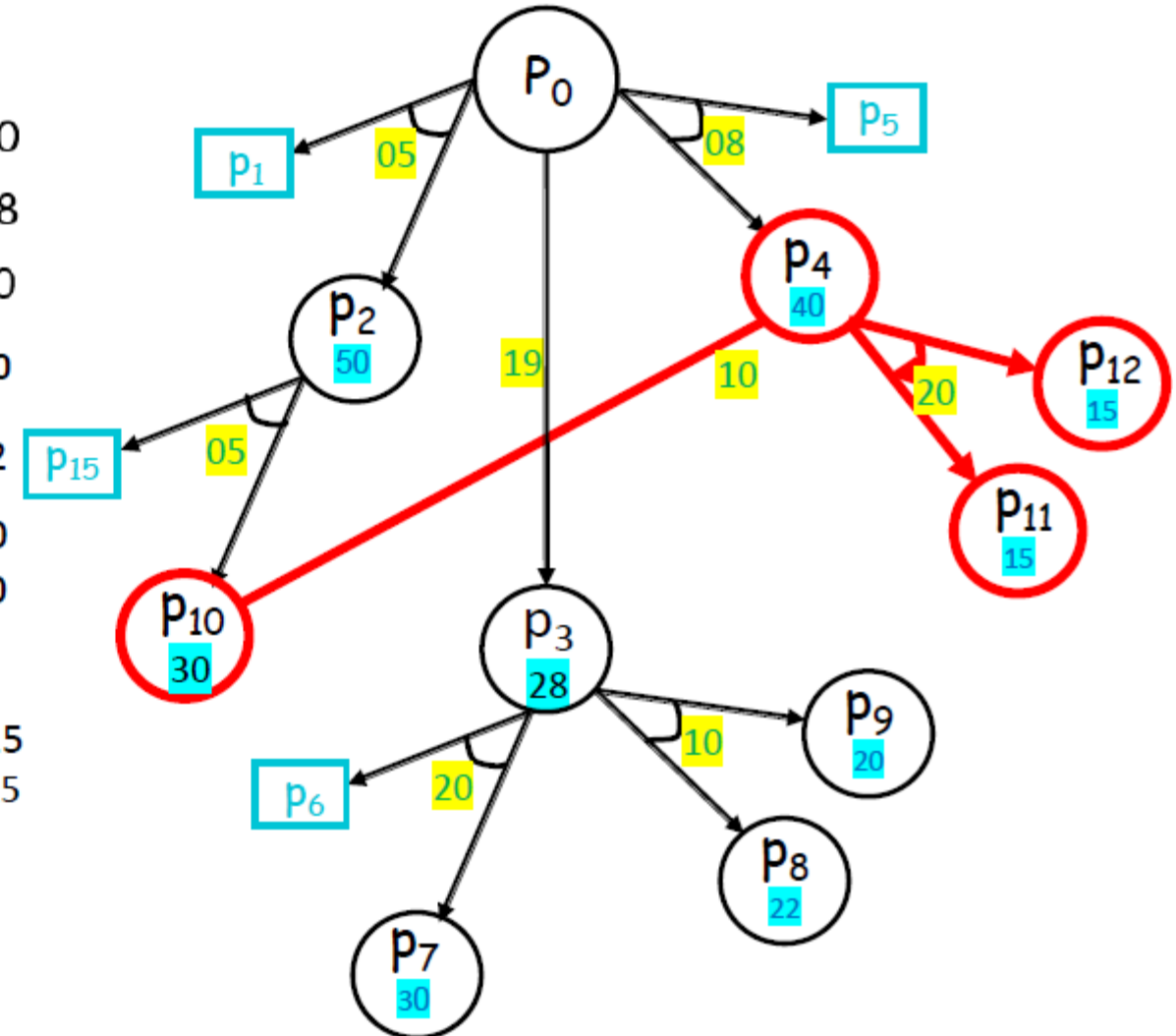
Op ₁ : $P_0 \Rightarrow p_1 \& p_2$	$K(op_1)=5$	$h(p_2)=50$
Op ₂ : $P_0 \Rightarrow p_3$	$K(op_2)=19$	$h(p_3)=28$
Op ₃ : $P_0 \Rightarrow p_4 \& p_5$	$K(op_3)=8$	$h(p_4)=40$
Op ₄ : $P_2 \Rightarrow p_{15} \& p_{10}$	$K(op_4)=5$	$h(p_{10})=30$
Op ₅ : $P_3 \Rightarrow p_6 \& p_7$	$K(op_5)=20$	$h(p_8)=22$
Op ₆ : $P_3 \Rightarrow p_8 \& p_9$	$K(op_6)=10$	$h(p_7)=30$ $h(p_9)=20$



AO* Search Algorithm

Solution: Search Tree Cont.

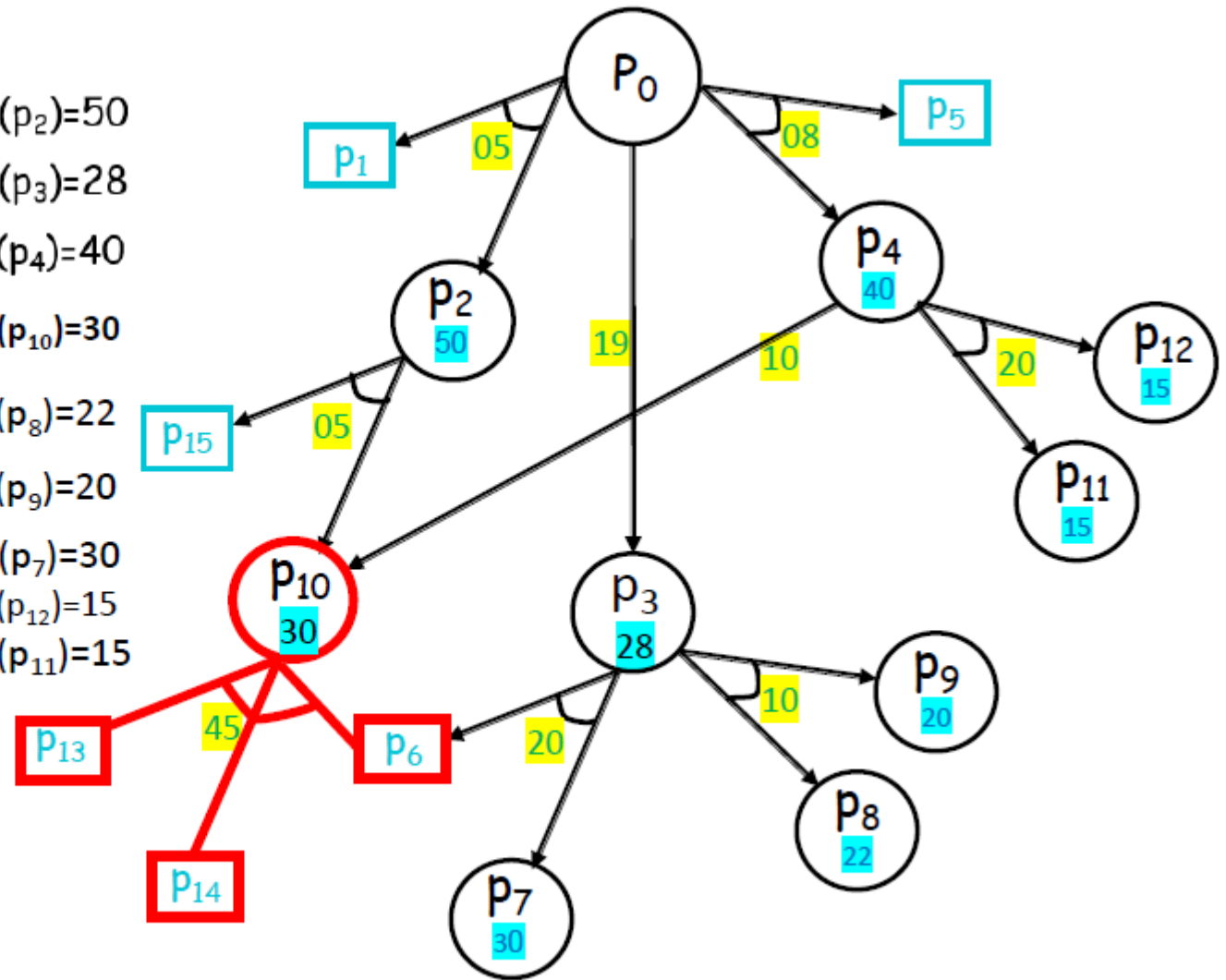
Op ₁ : $P_0 \Rightarrow p_1 \& p_2$	$K(op_1)=5$	$h(p_2)=50$
Op ₂ : $P_0 \Rightarrow p_3$	$k(op_2)=19$	$h(p_3)=28$
Op ₃ : $P_0 \Rightarrow p_4 \& p_5$	$K(op_3)=8$	$h(p_4)=40$
Op ₄ : $P_2 \Rightarrow p_{15} \& p_{10}$	$K(op_4)=5$	$h(p_{10})=30$
Op ₅ : $P_3 \Rightarrow p_6 \& p_7$	$K(op_5)=20$	$h(p_8)=22$
Op ₆ : $P_3 \Rightarrow p_8 \& p_9$	$K(op_6)=10$	$h(p_9)=20$ $h(p_7)=30$
Op ₇ : $P_4 \Rightarrow p_{10}$	$K(op_7)=10$	
Op ₈ : $P_4 \Rightarrow p_{11} \& p_{12}$	$K(op_8)=20$	$h(p_{11})=15$ $h(p_{12})=15$



AO* Search Algorithm

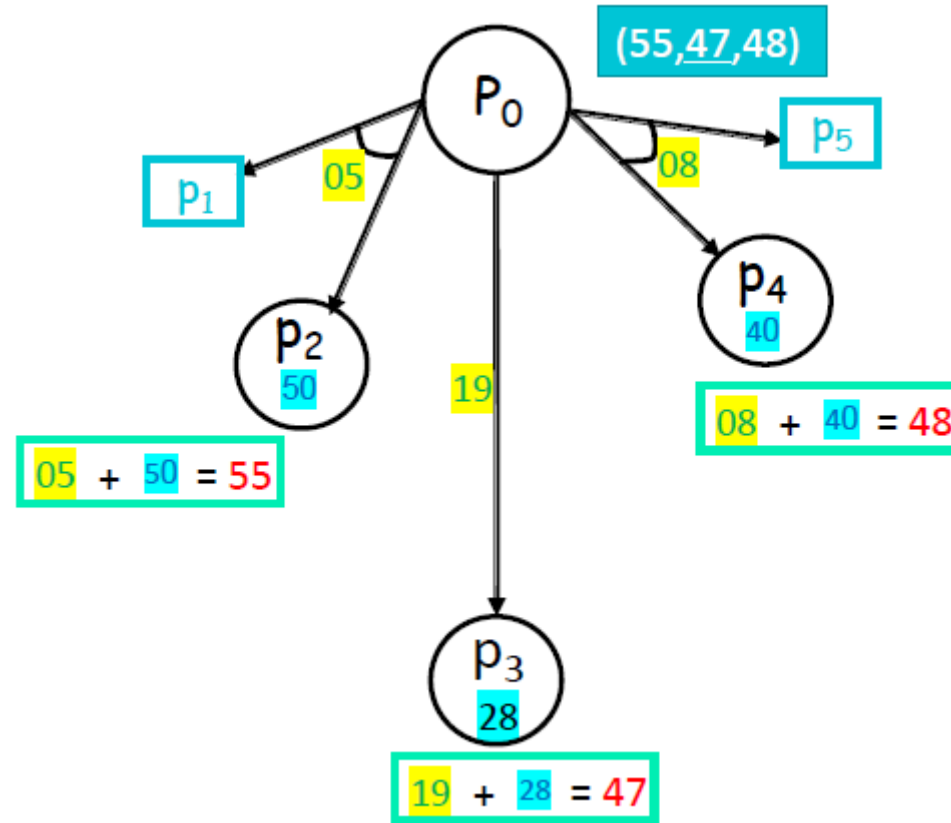
Solution: Search Tree Cont.

Op ₁ : $P_0 \Rightarrow p_1 \& p_2$	$K(op_1)=5$	$h(p_2)=50$
Op ₂ : $P_0 \Rightarrow p_3$	$k(op_2)=19$	$h(p_3)=28$
Op ₃ : $P_0 \Rightarrow p_4 \& p_5$	$K(op_3)=8$	$h(p_4)=40$
Op ₄ : $P_2 \Rightarrow p_{15} \& p_{10}$	$K(op_4)=5$	$h(p_{10})=30$
Op ₅ : $P_3 \Rightarrow p_6 \& p_7$	$K(op_5)=20$	$h(p_8)=22$
Op ₆ : $P_3 \Rightarrow p_8 \& p_9$	$K(op_6)=10$	$h(p_9)=20$
Op ₇ : $P_4 \Rightarrow p_{10}$	$K(op_7)=10$	$h(p_7)=30$
Op ₈ : $P_4 \Rightarrow p_{11} \& p_{12}$	$K(op_8)=20$	$h(p_{12})=15$
Op ₉ : $P_{10} \Rightarrow p_{13} \& p_{14} \& p_6$	$k(op_9)=45$	$h(p_{11})=15$



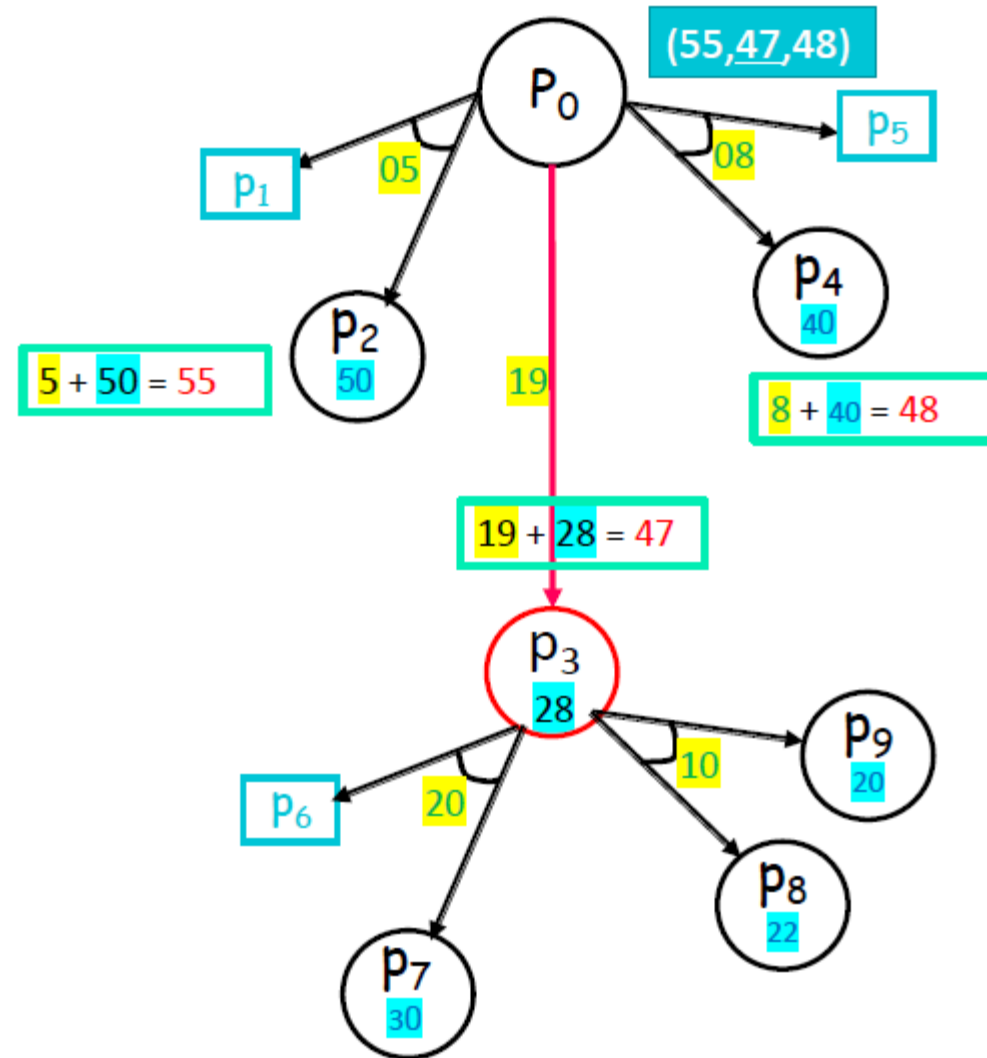
AO* Search Algorithm

Solution: Solve problems



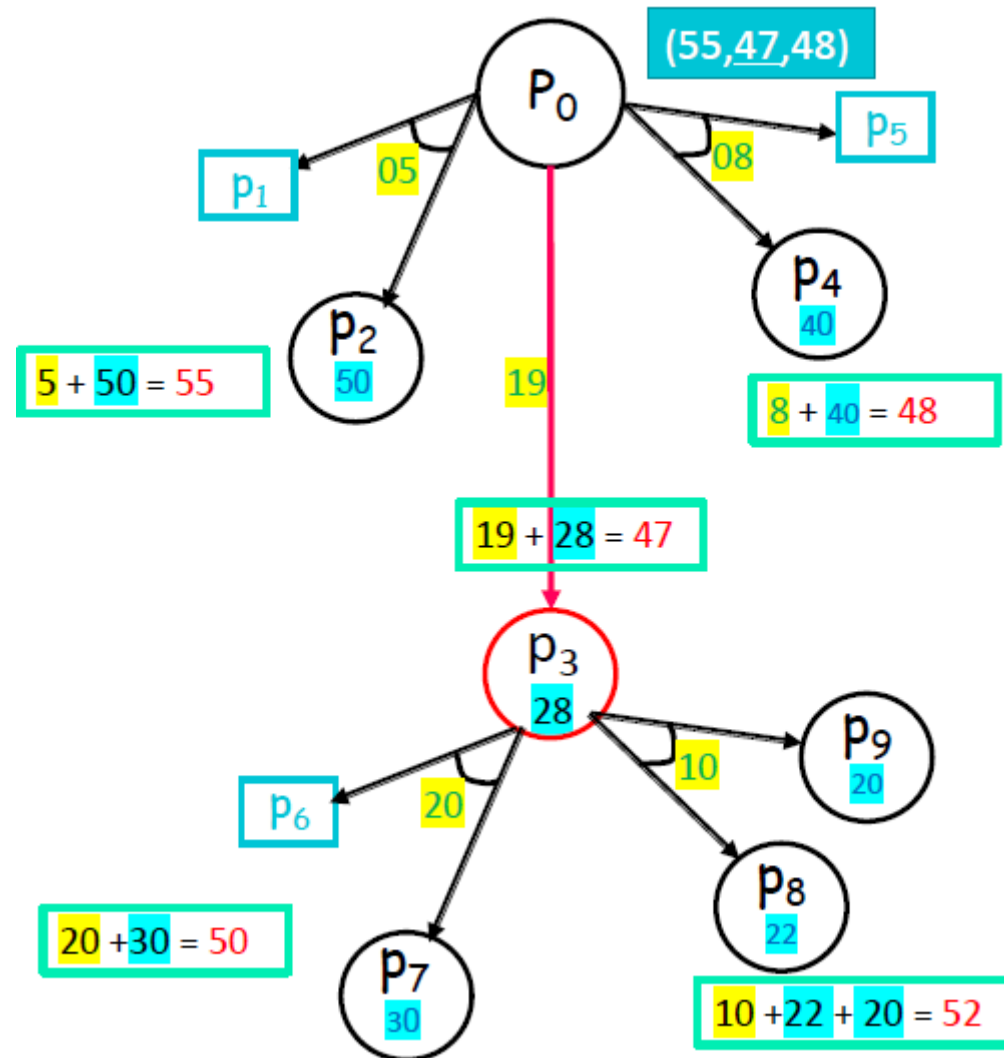
AO* Search Algorithm

Solution: Solve problems Cont.



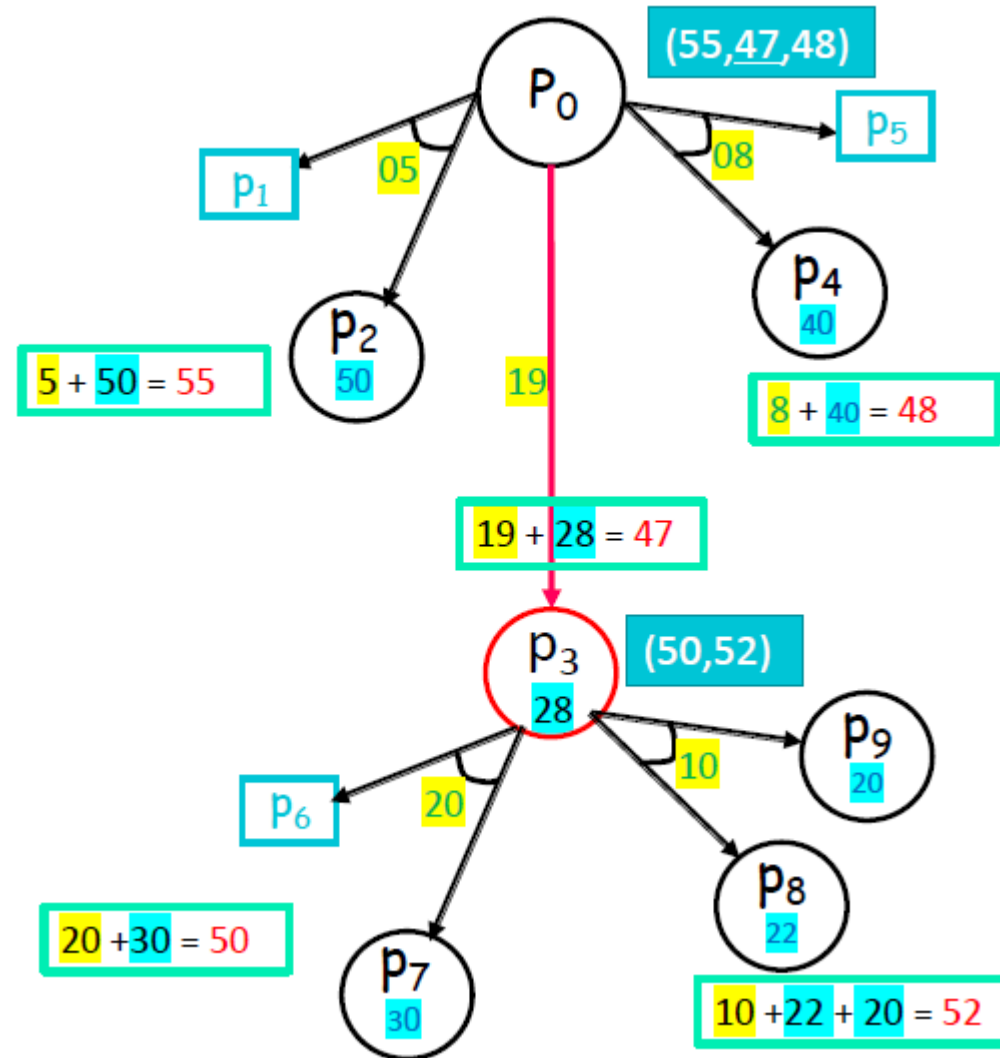
AO* Search Algorithm

Solution: Solve problems Cont.



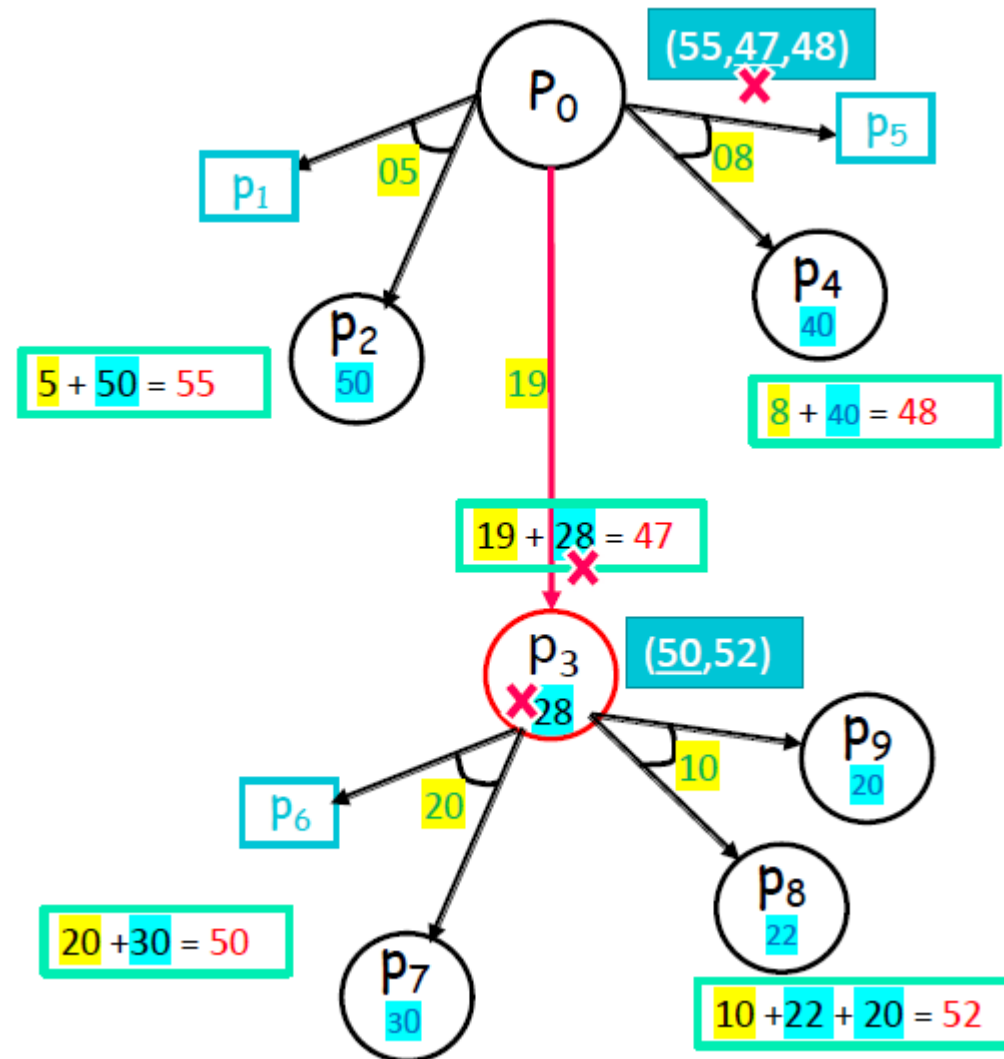
AO* Search Algorithm

Solution: Solve problems Cont.



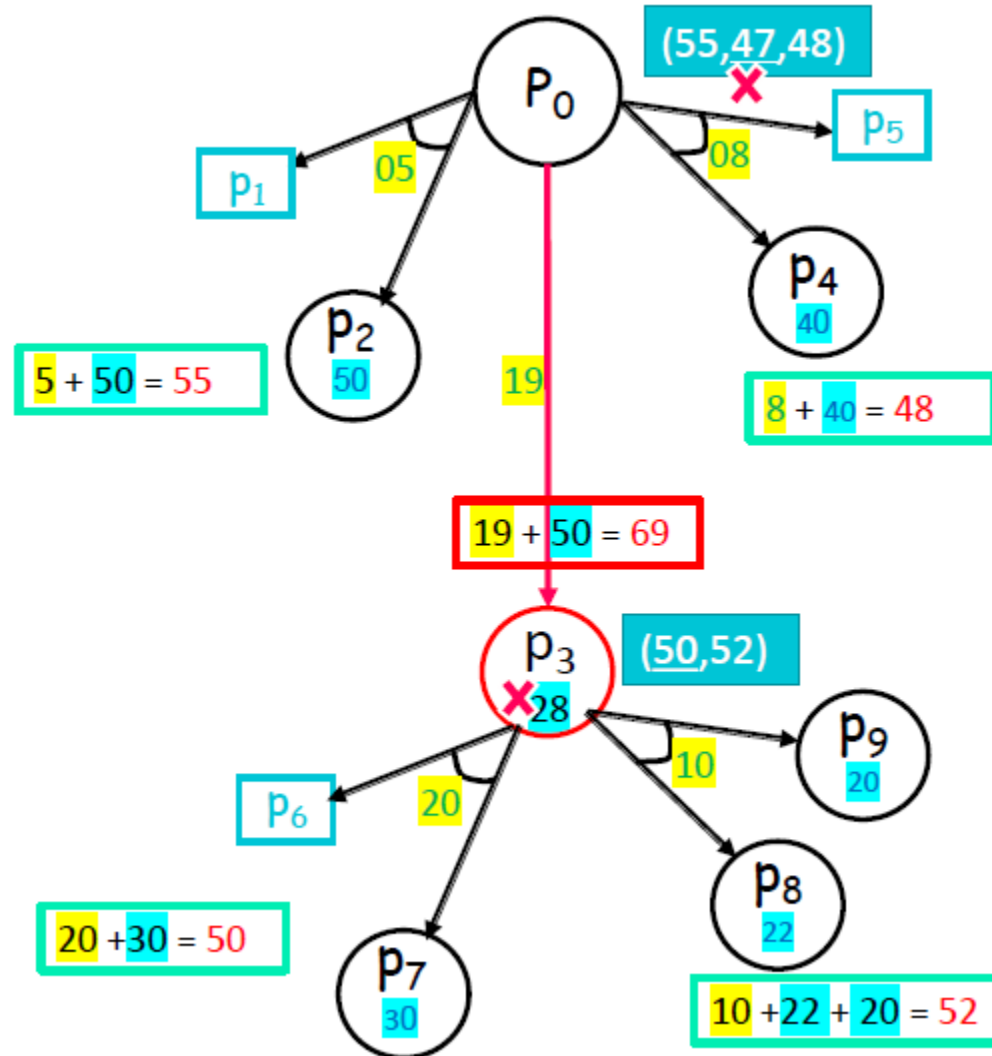
AO* Search Algorithm

Solution: Solve problems Cont.



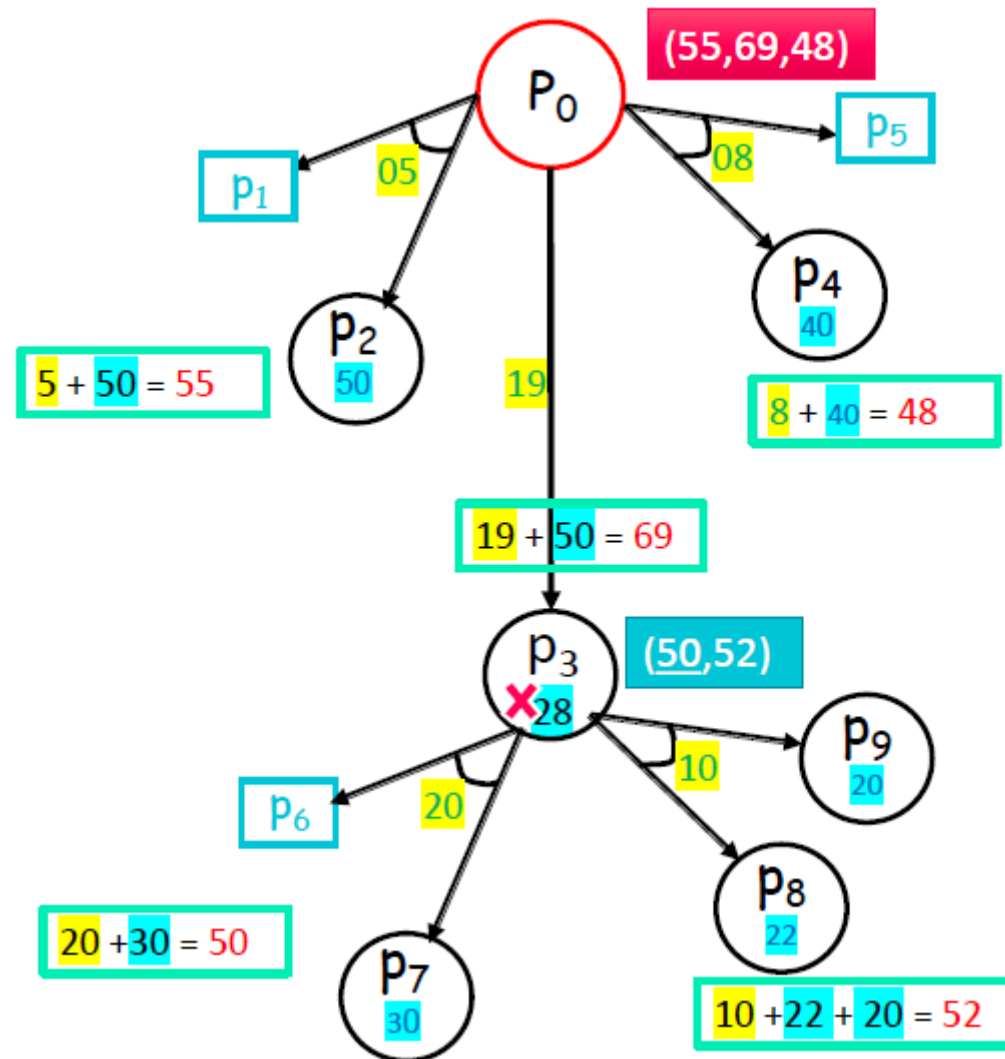
AO* Search Algorithm

Solution: Solve problems Cont.



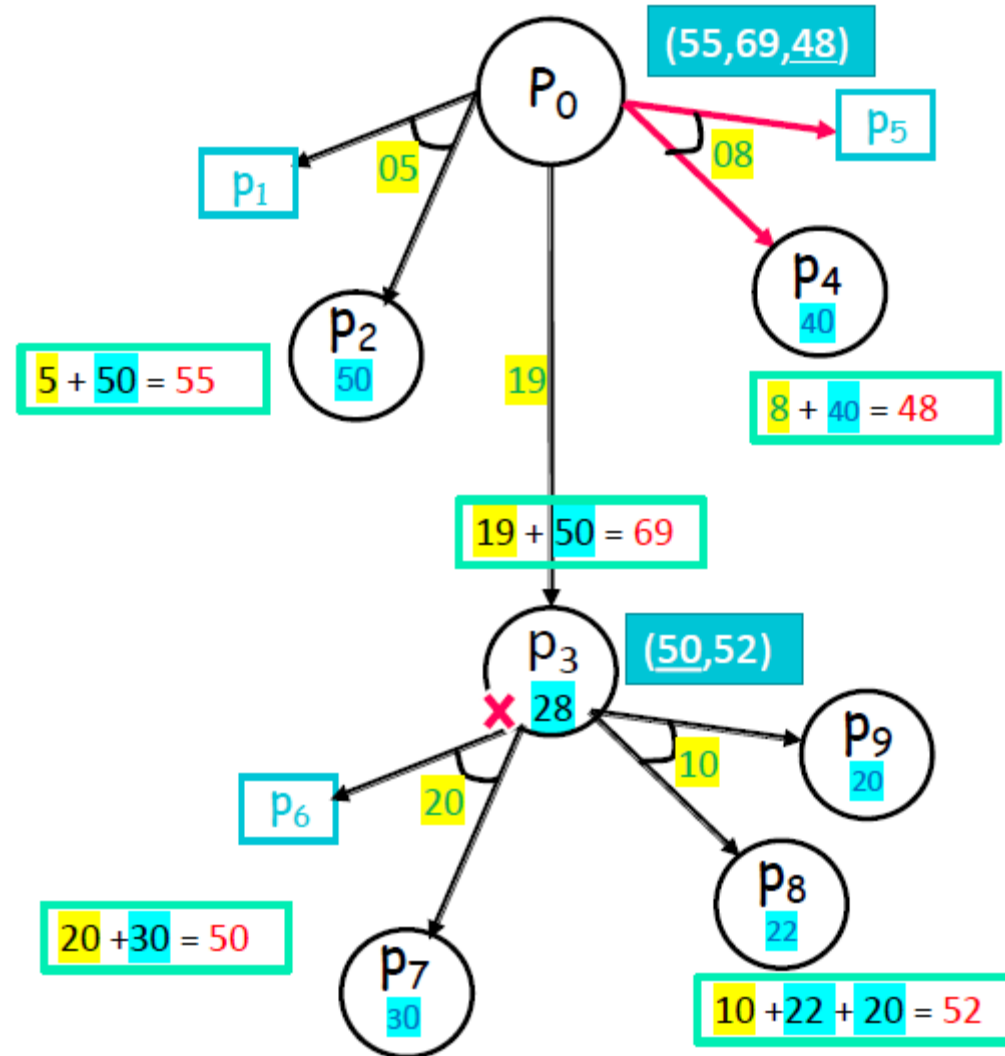
AO* Search Algorithm

Solution: Solve problems Cont.



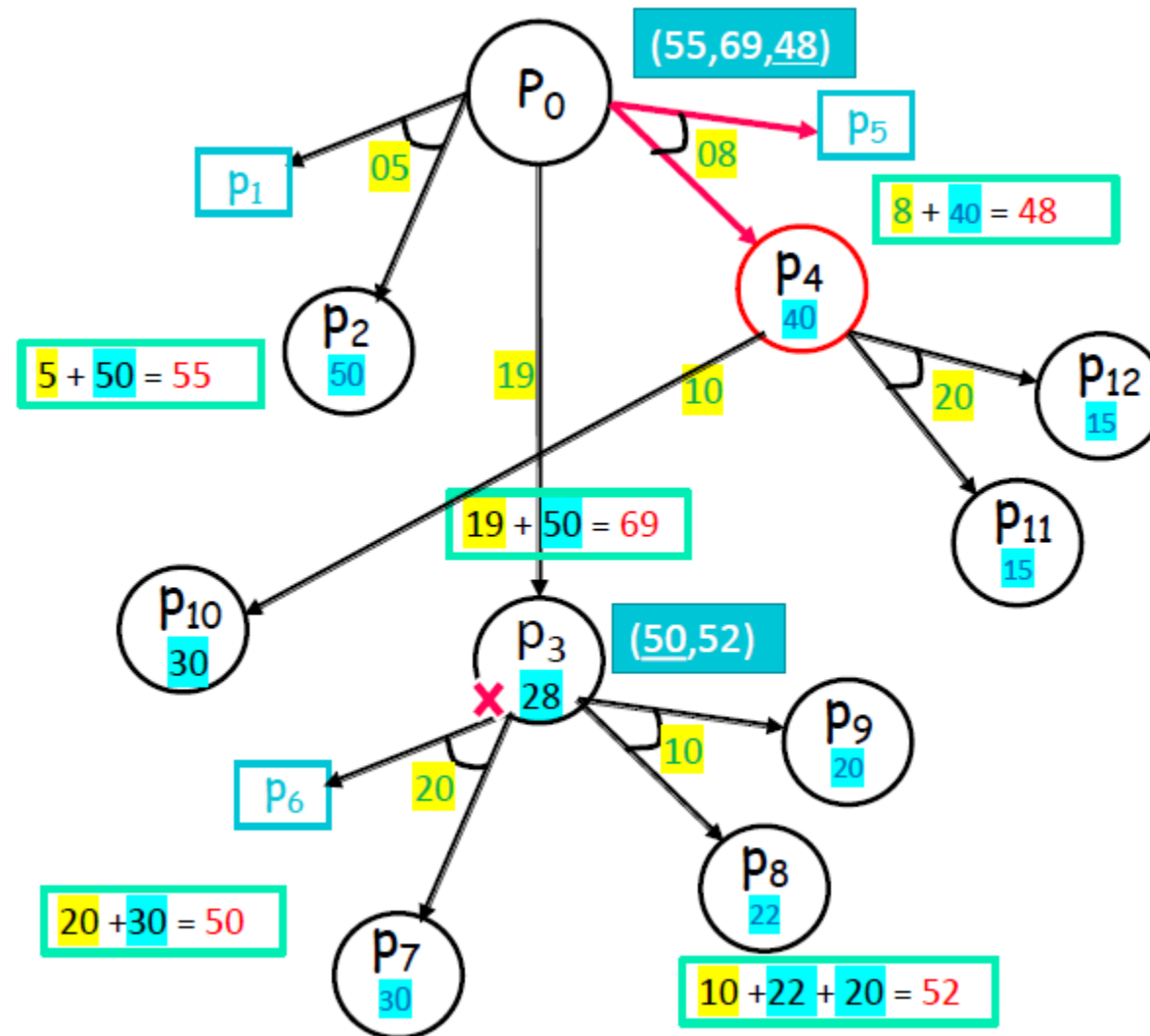
AO* Search Algorithm

Solution: Solve problems Cont.



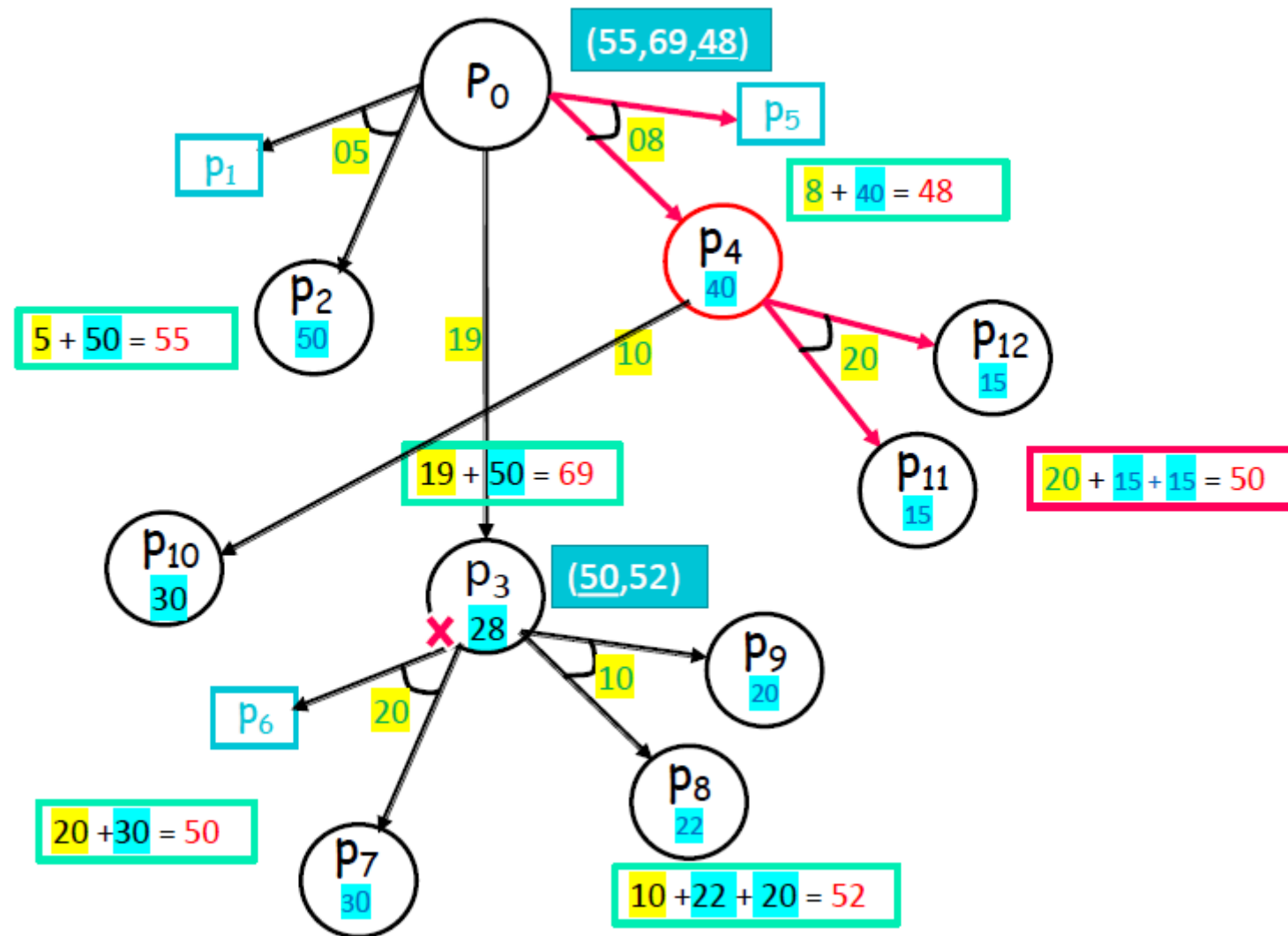
AO* Search Algorithm

Solution: Solve problems Cont.



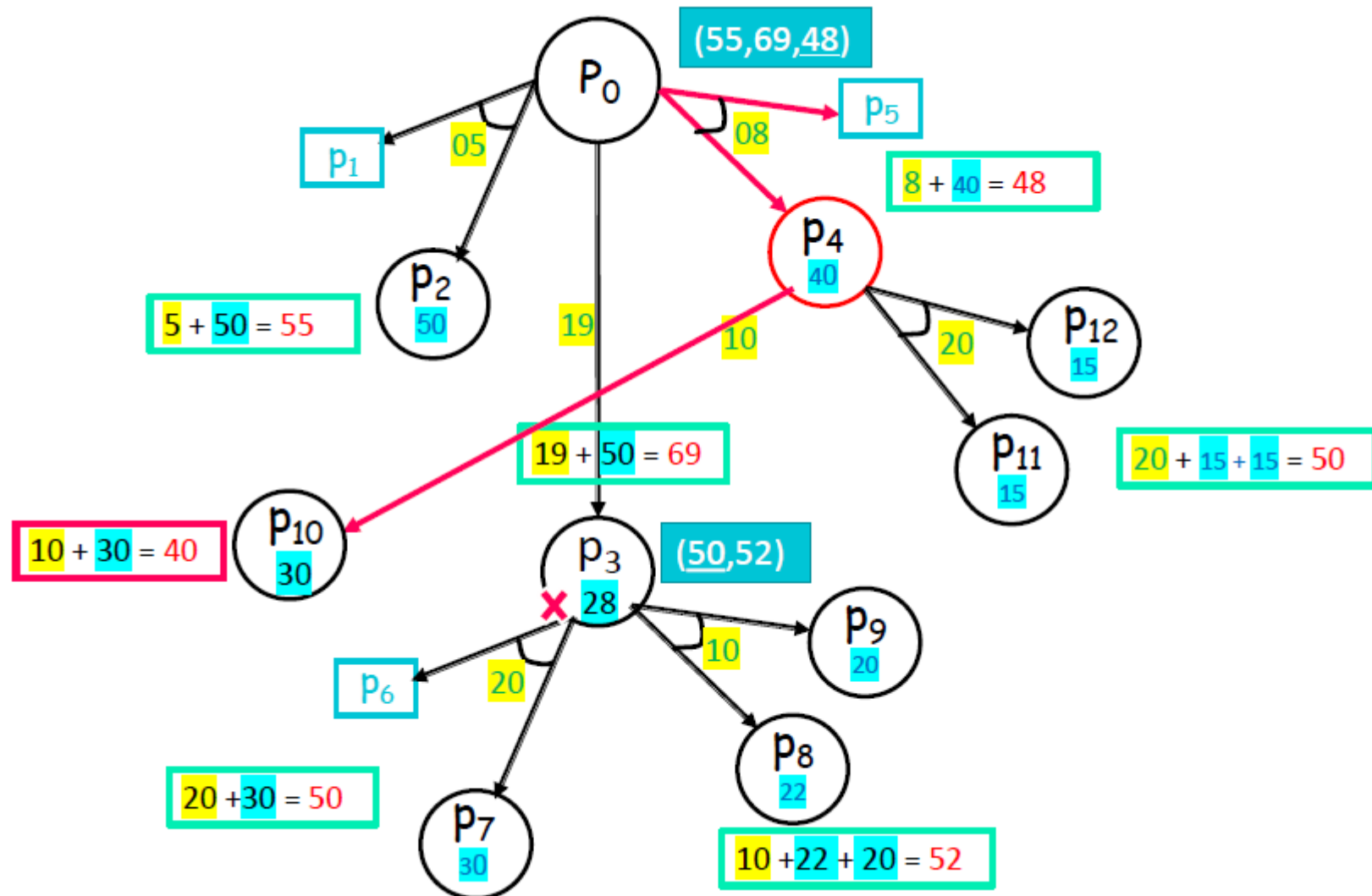
AO* Search Algorithm

Solution: Solve problems Cont.



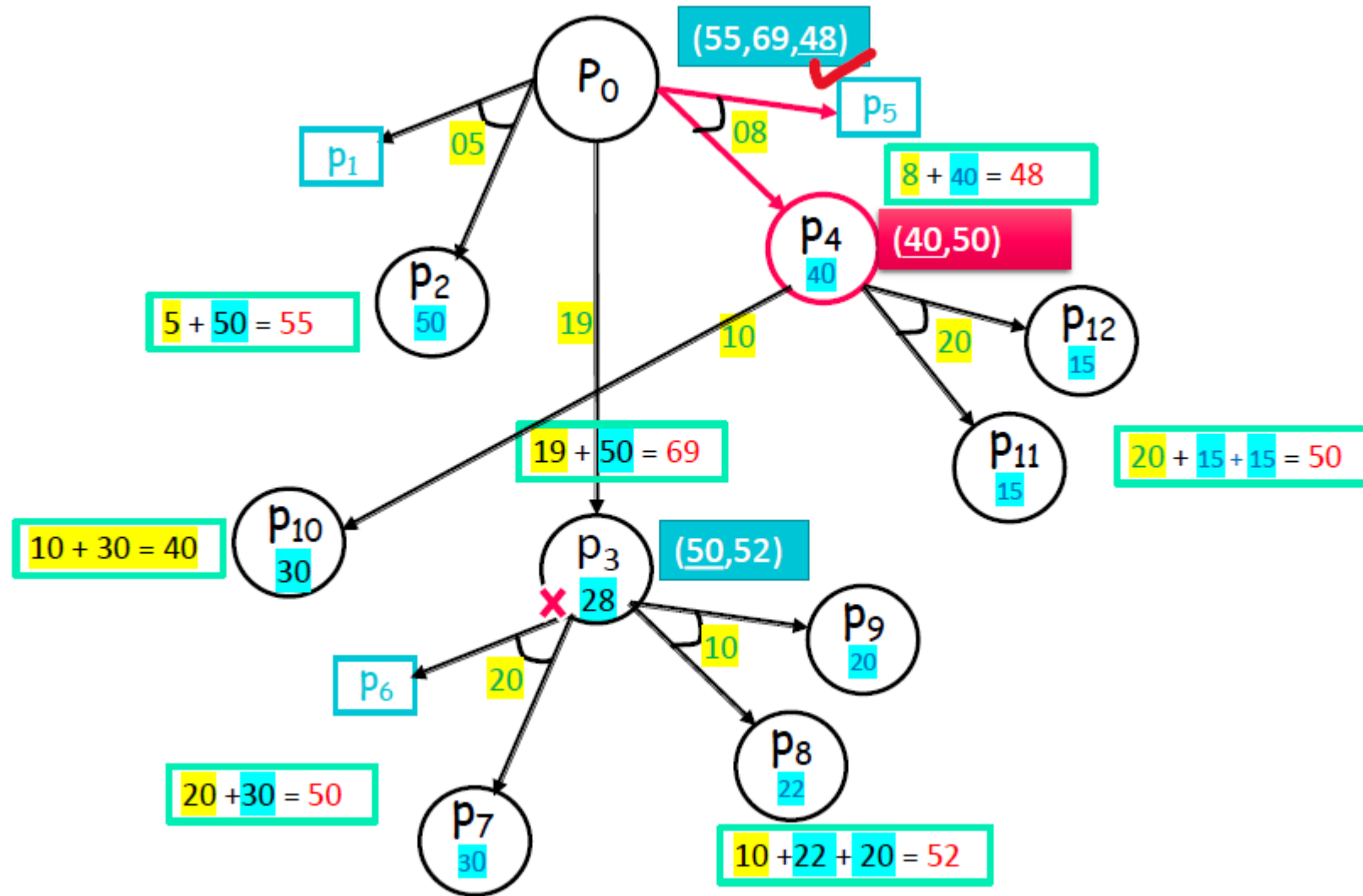
AO* Search Algorithm

Solution: Solve problems Cont.



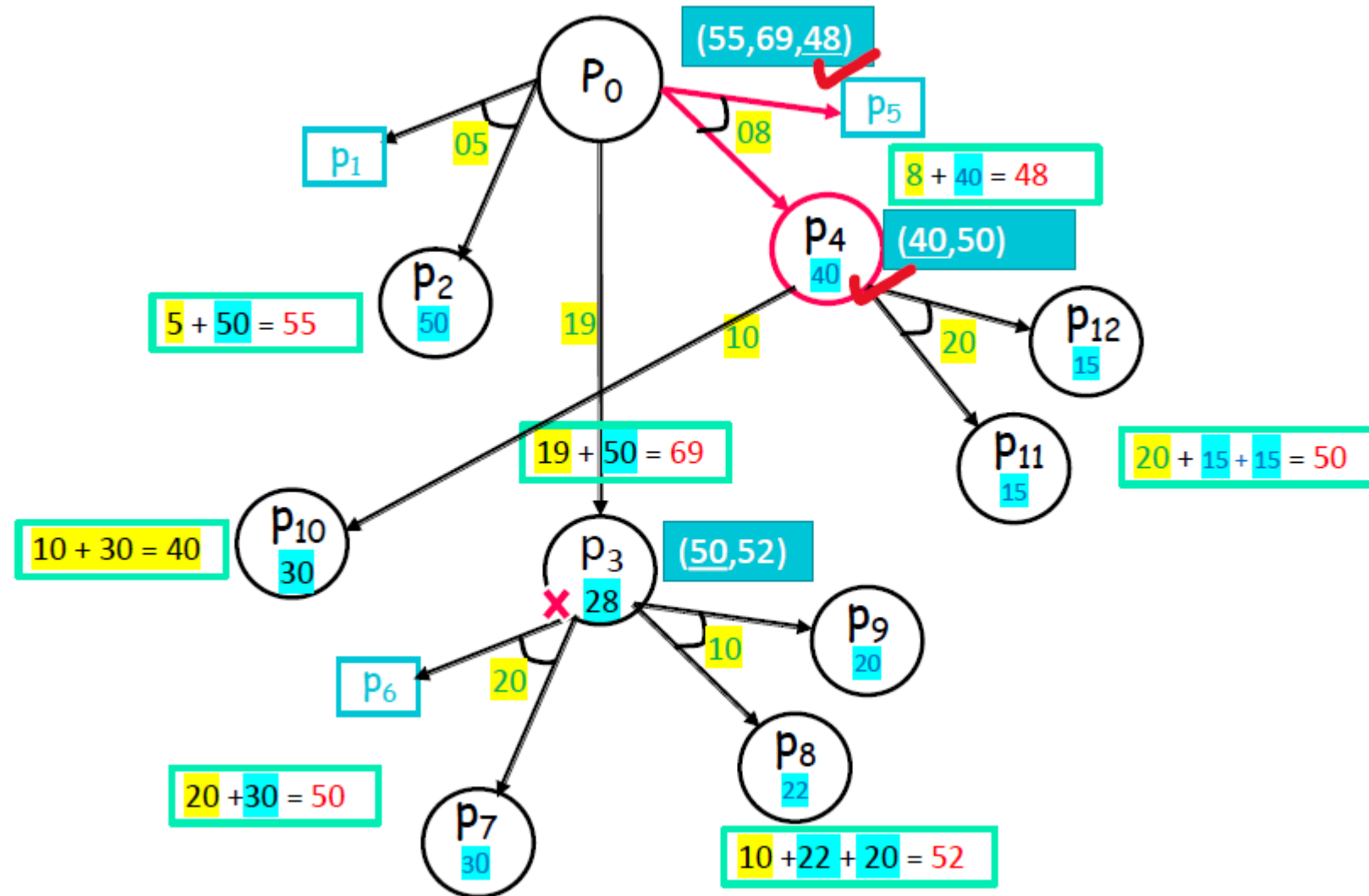
AO* Search Algorithm

Solution: Solve problems Cont.



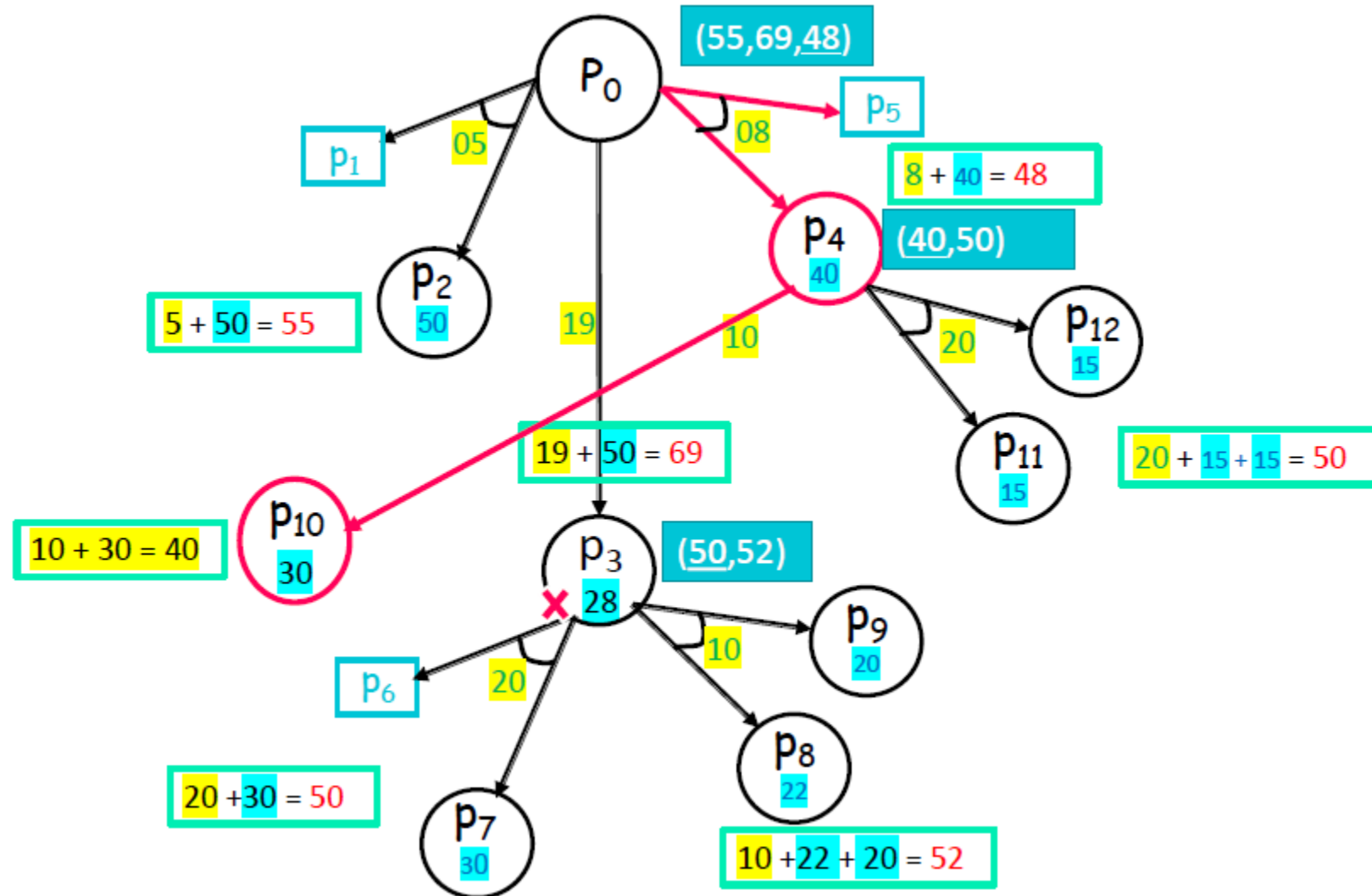
AO* Search Algorithm

Solution: Solve problems Cont.



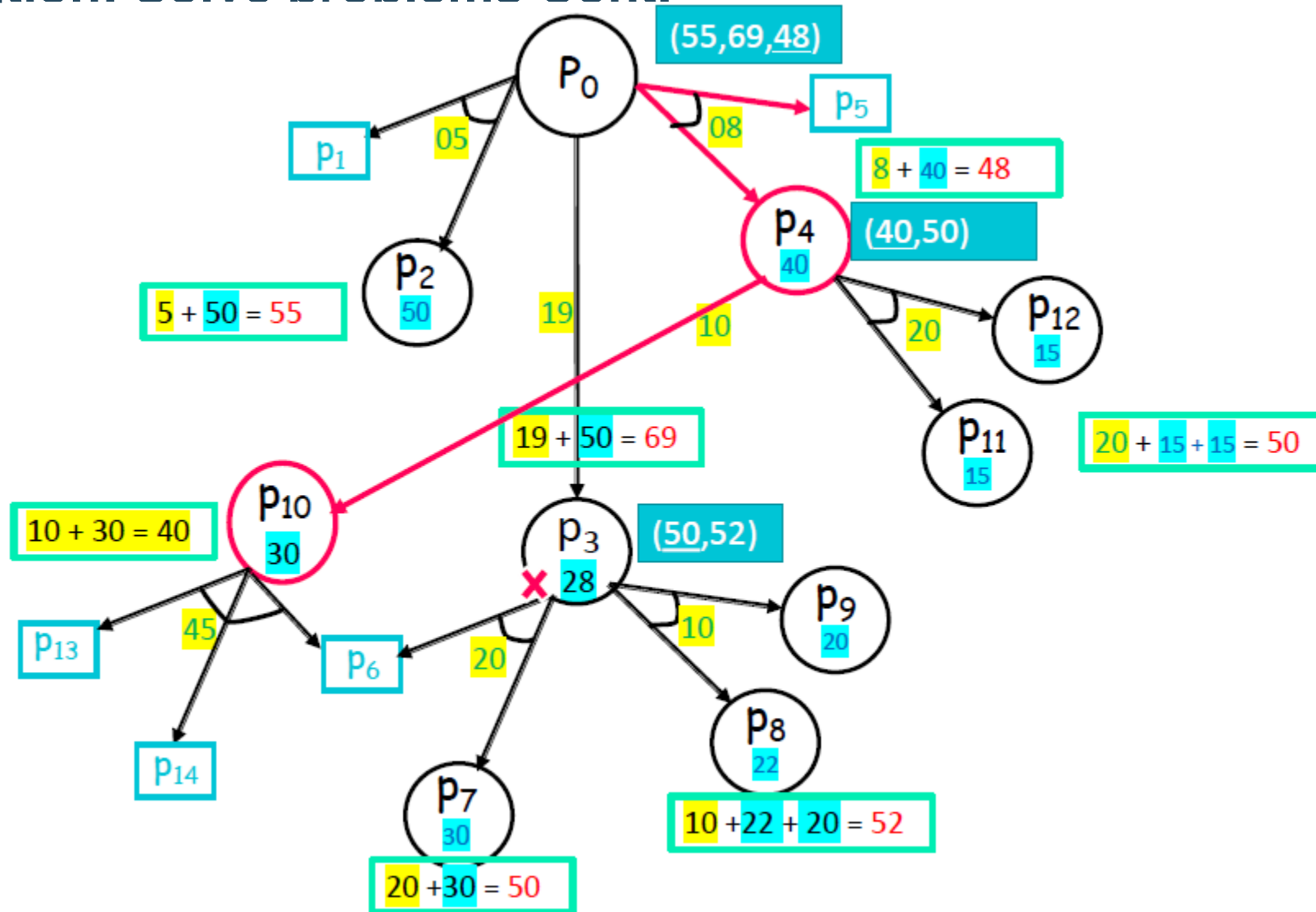
AO* Search Algorithm

Solution: Solve problems Cont.



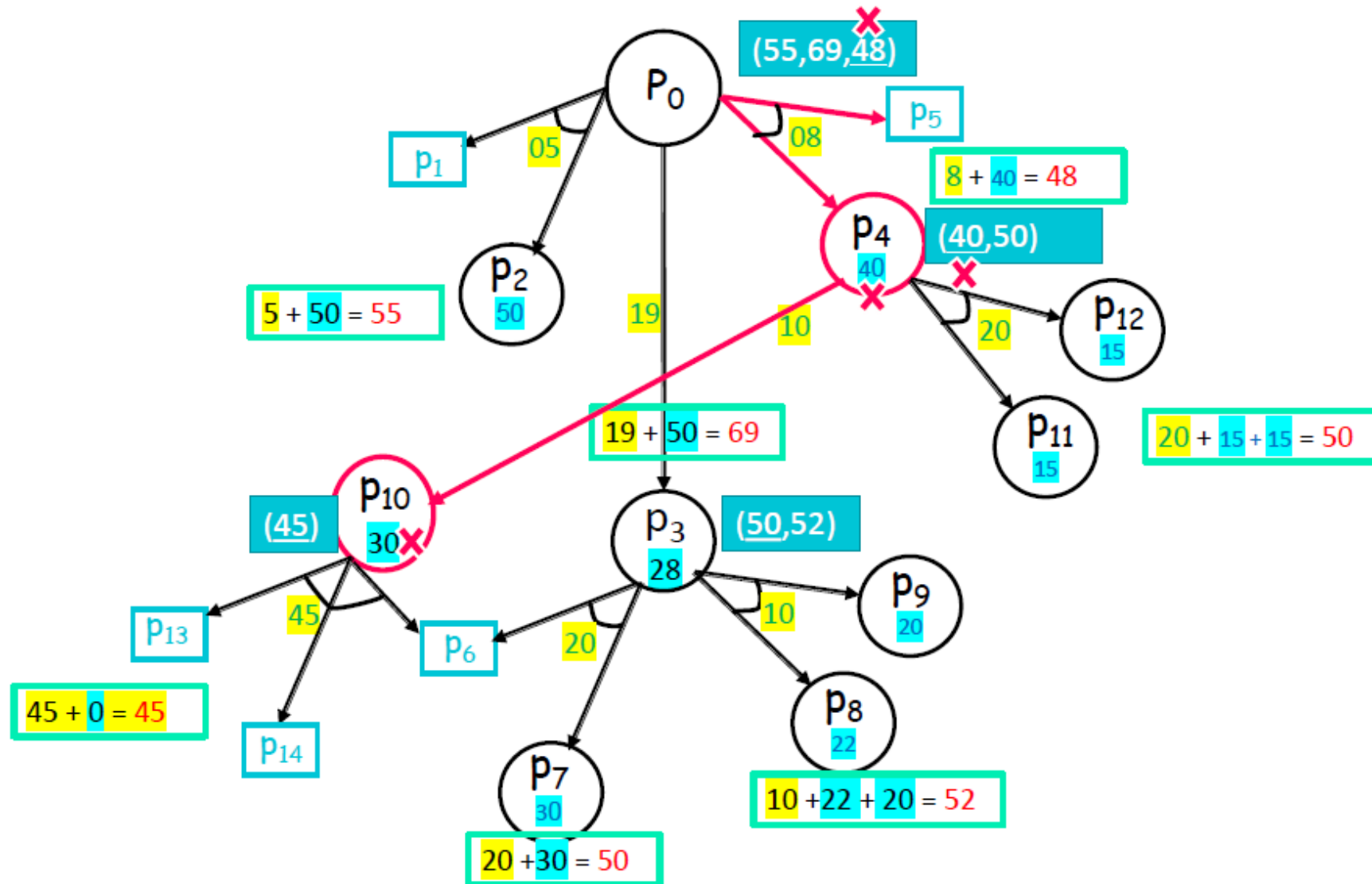
AO* Search Algorithm

Solution: Solve problems Cont.



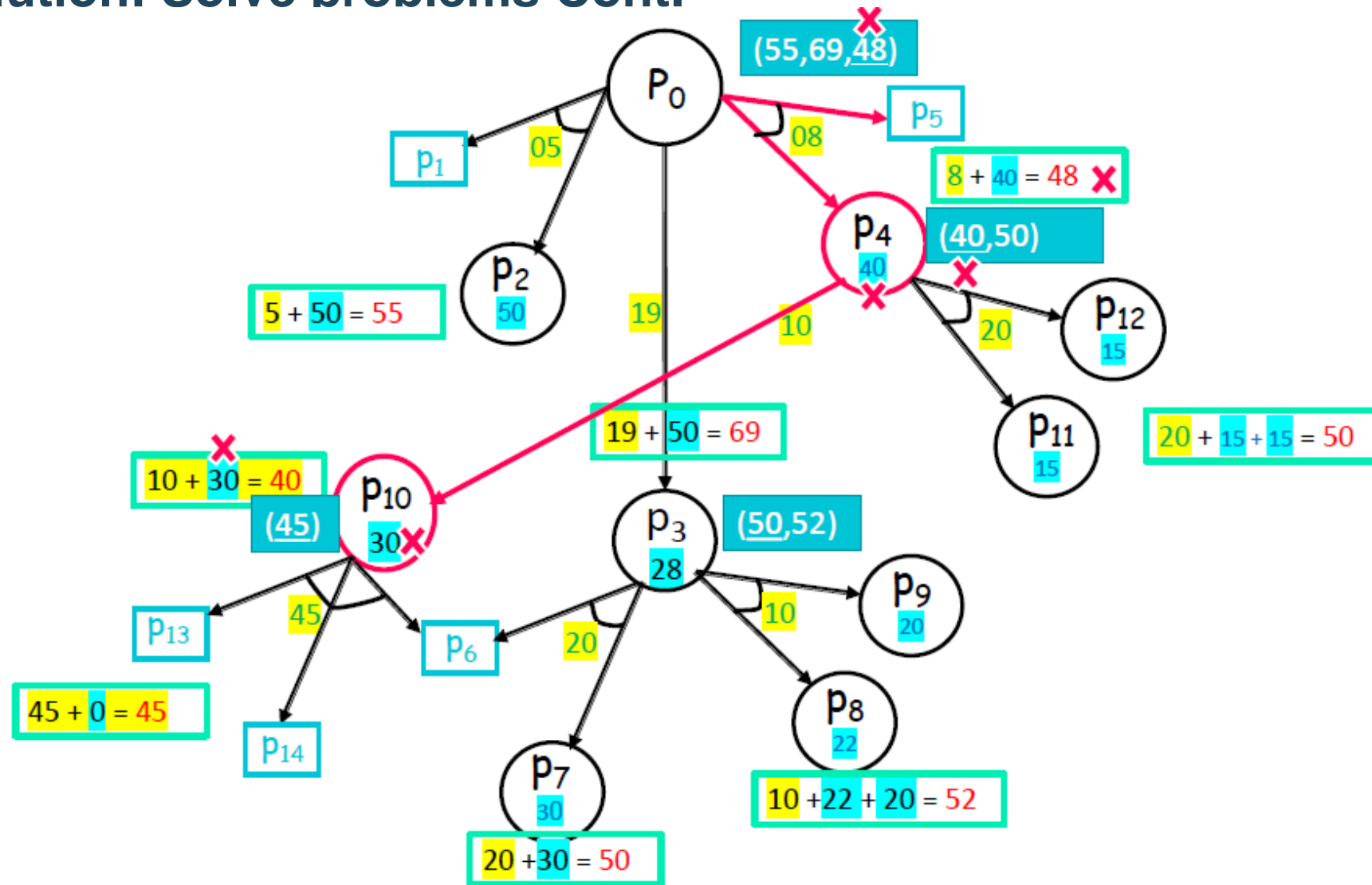
AO* Search Algorithm

Solution: Solve problems Cont.



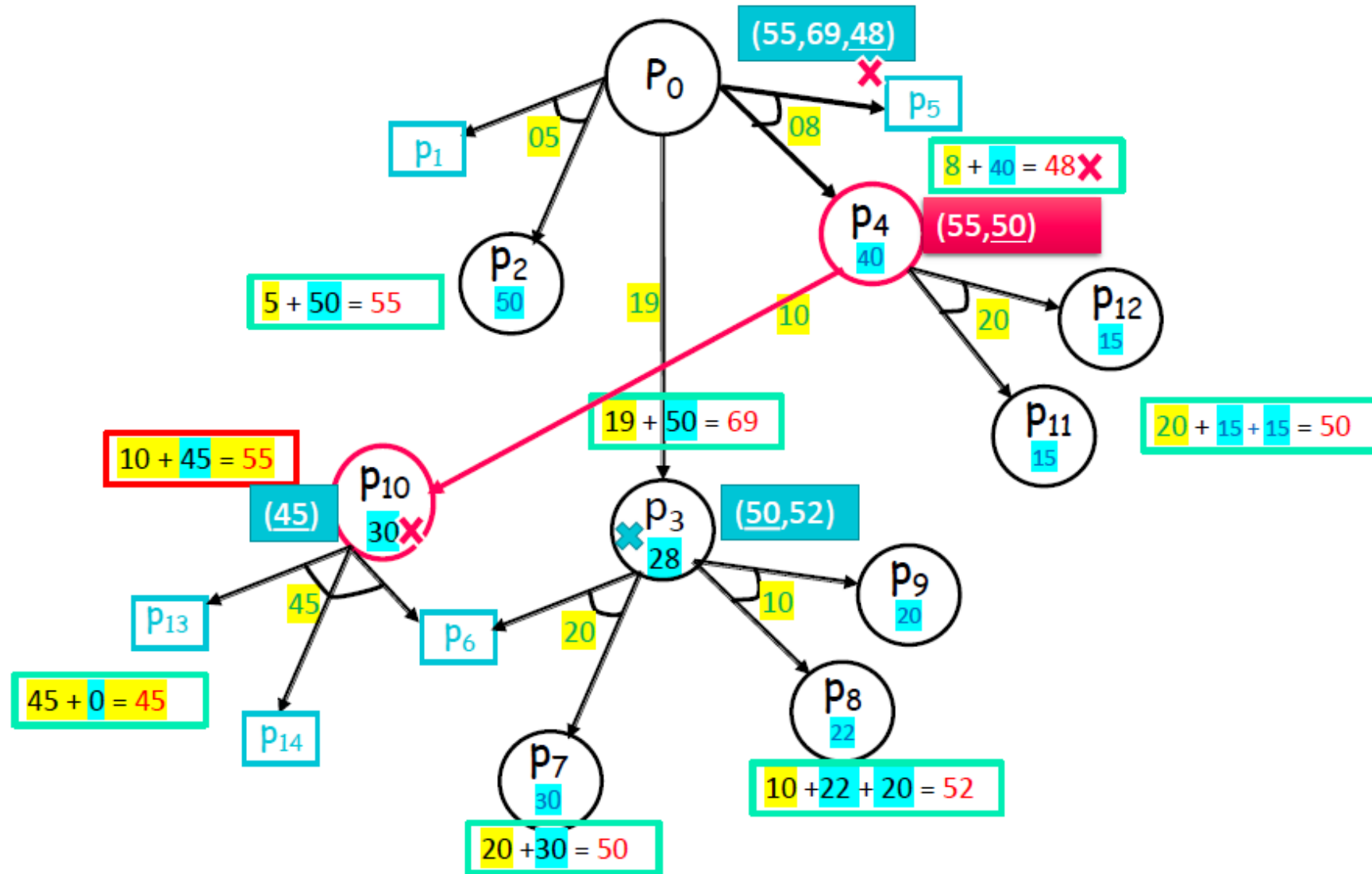
AO* Search Algorithm

Solution: Solve problems Cont.



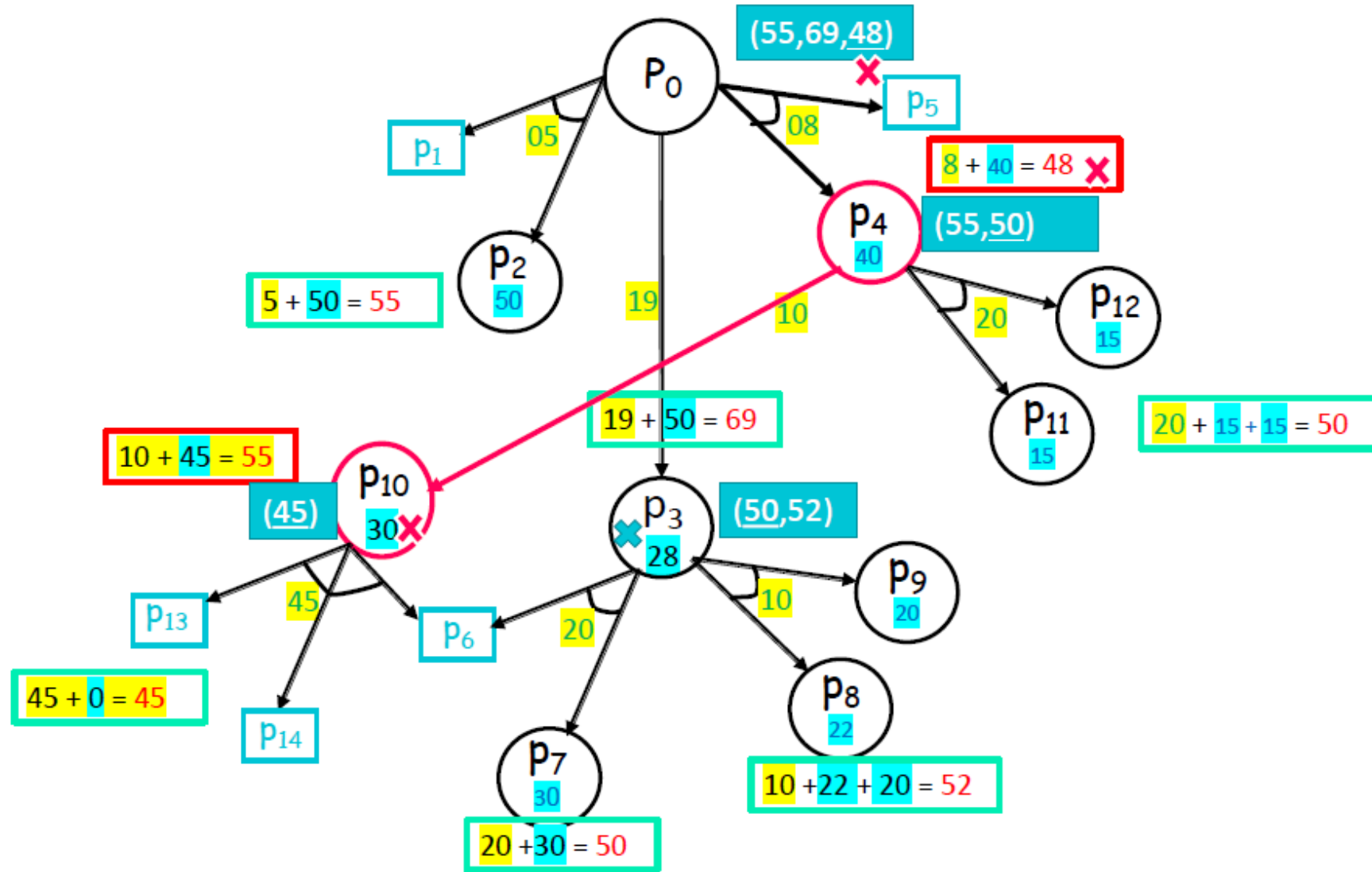
AO* Search Algorithm

Solution: Solve problems Cont.



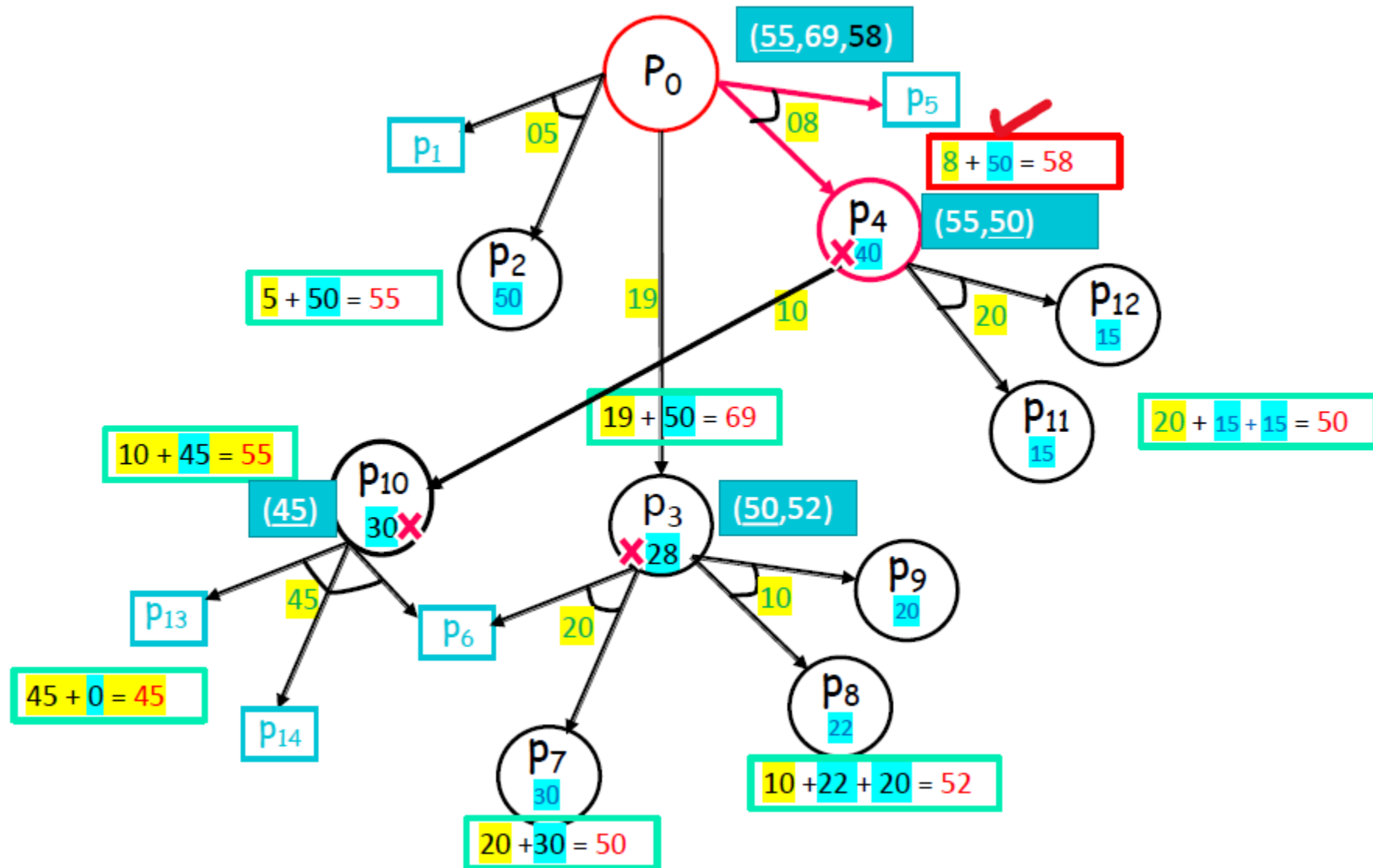
AO* Search Algorithm

Solution: Solve problems Cont.



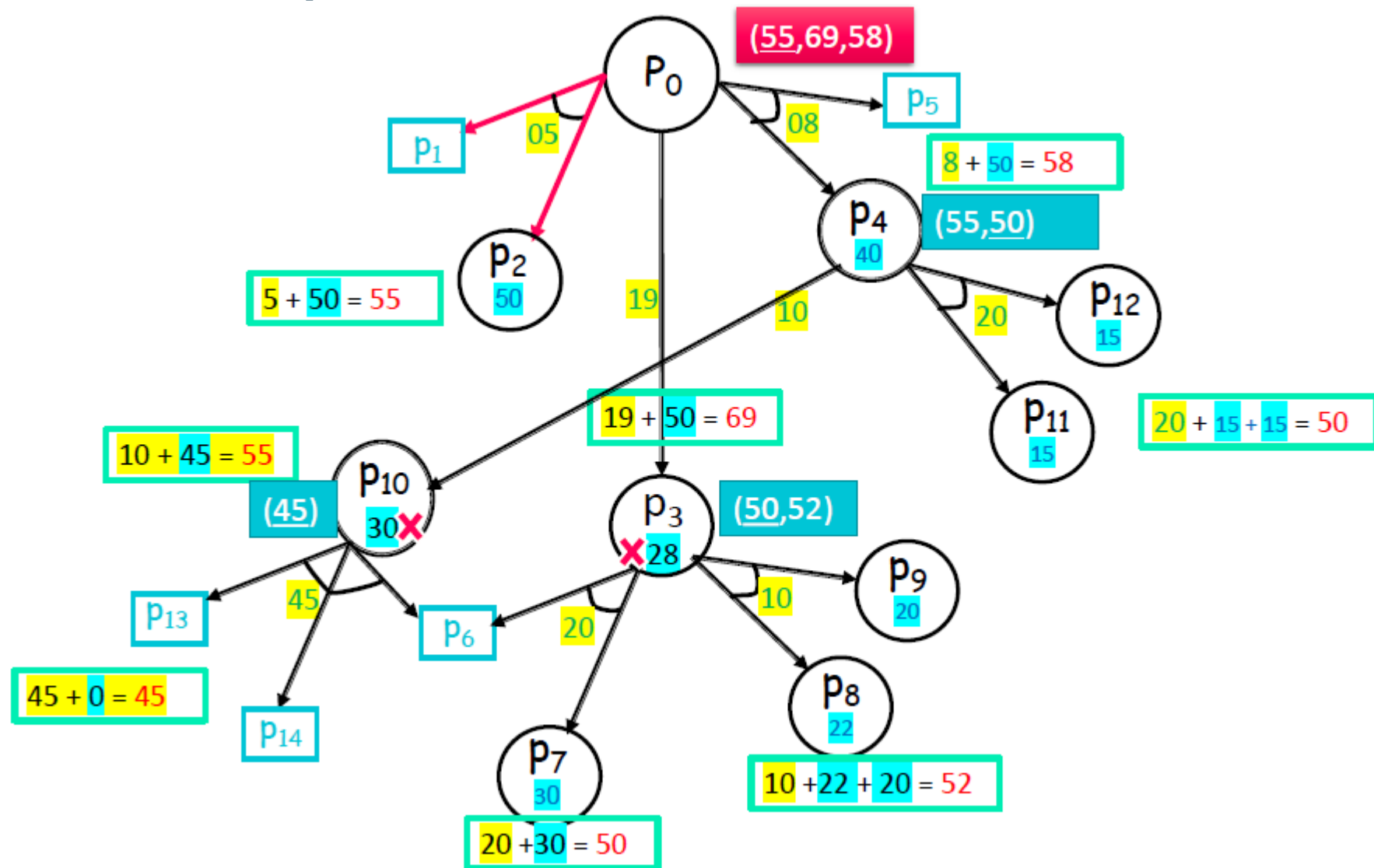
AO* Search Algorithm

Solution: Solve problems Cont.



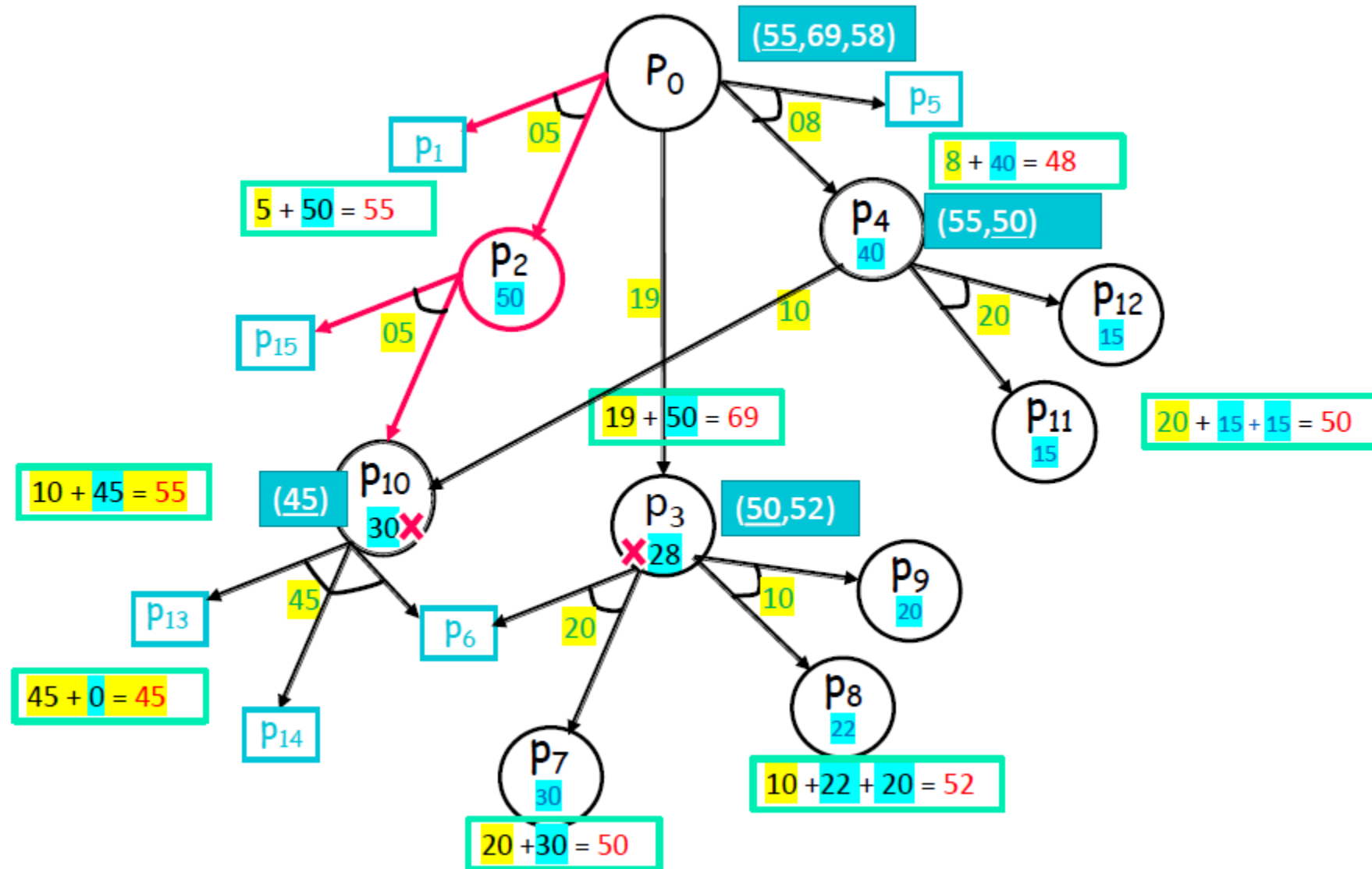
AO* Search Algorithm

Solution: Solve problems Cont.



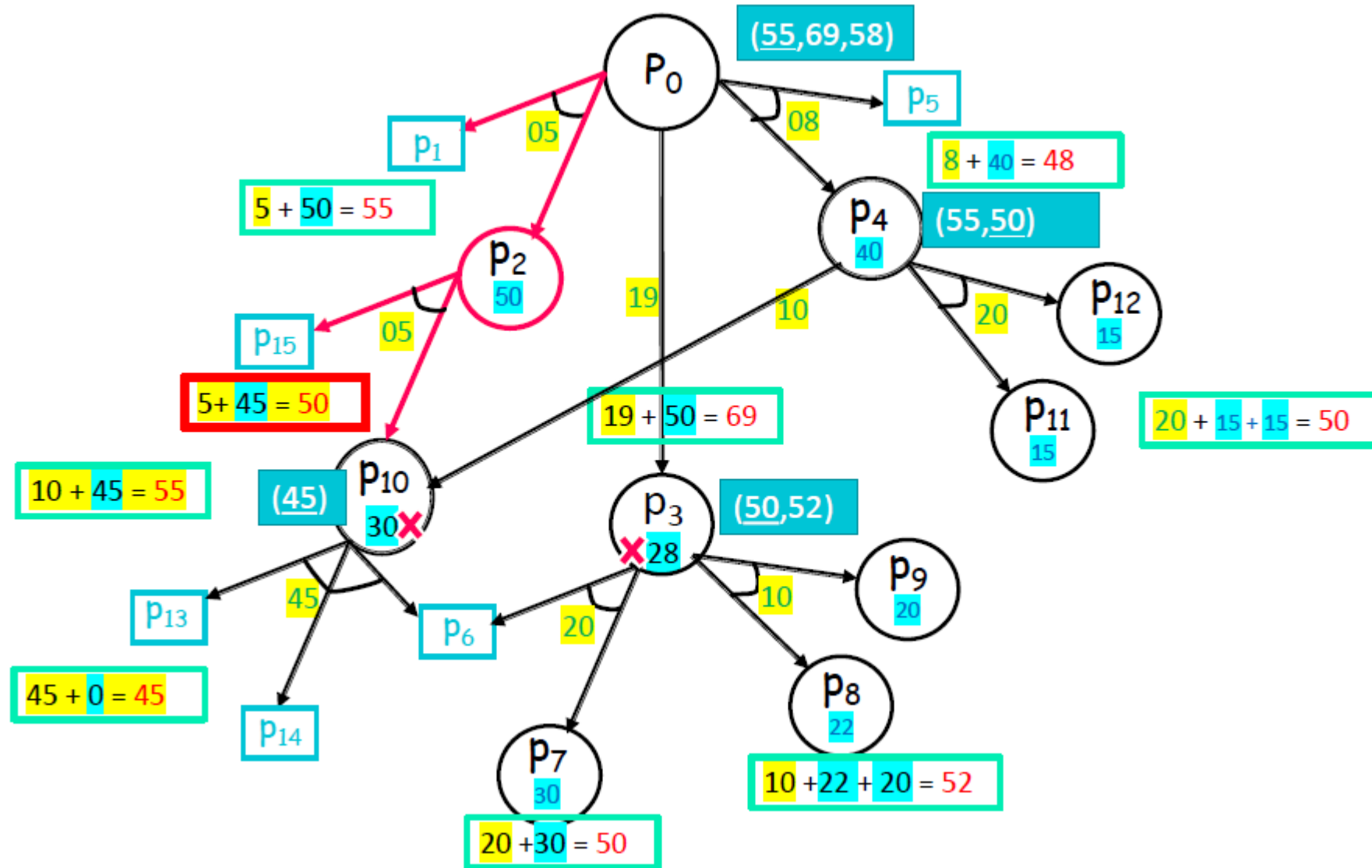
AO* Search Algorithm

Solution: Solve problems Cont.



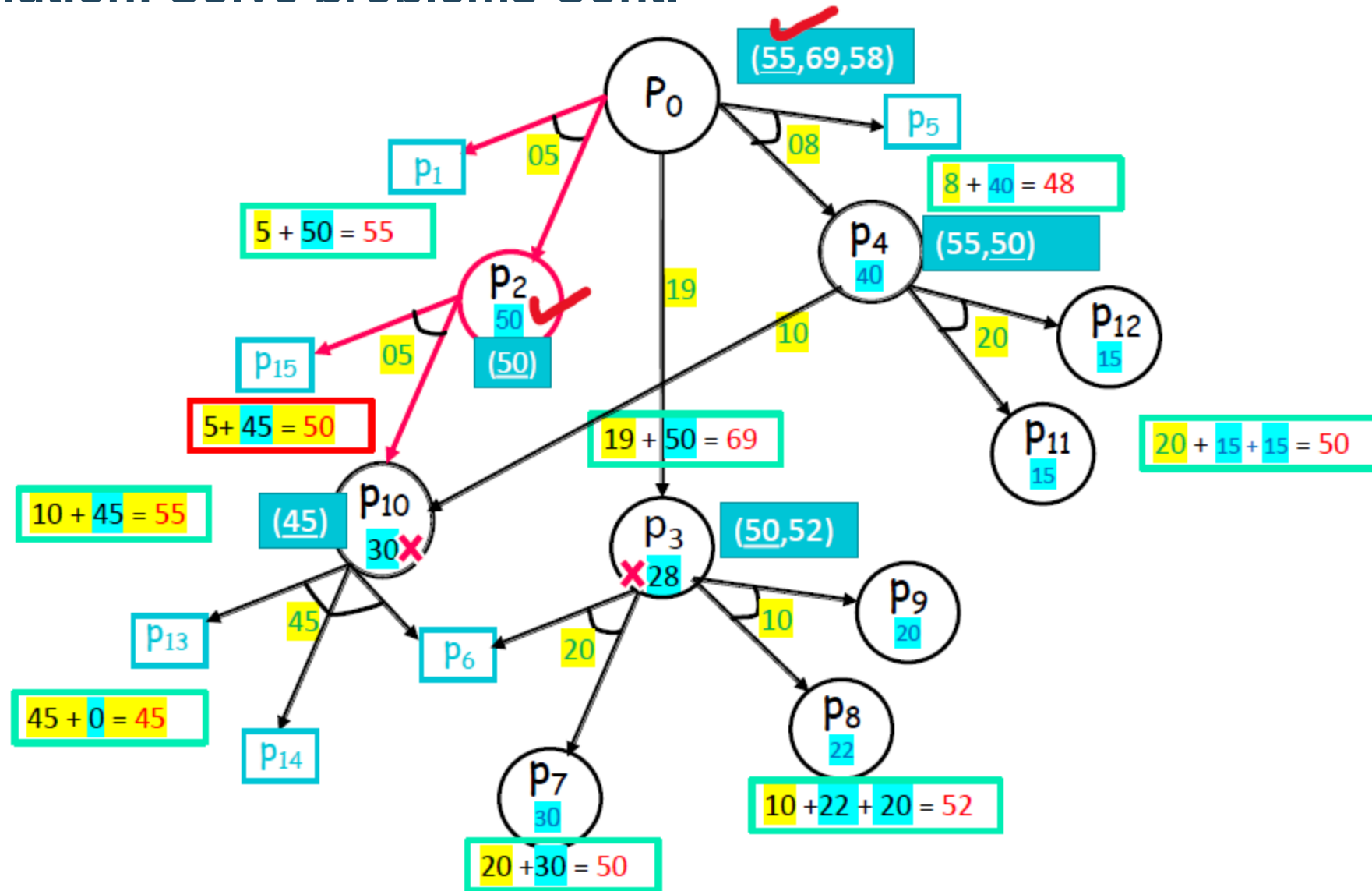
AO* Search Algorithm

Solution: Solve problems Cont.



AO* Search Algorithm

Solution: Solve problems Cont.



Solution: Solve problems Cont.





Any Questions