



Minimax

Softwareentwicklungspraktikum

PST



- Automatisches Planen
- Minimax
- Alpha-Beta-Pruning
- Tiefenbeschränkung & Heuristik

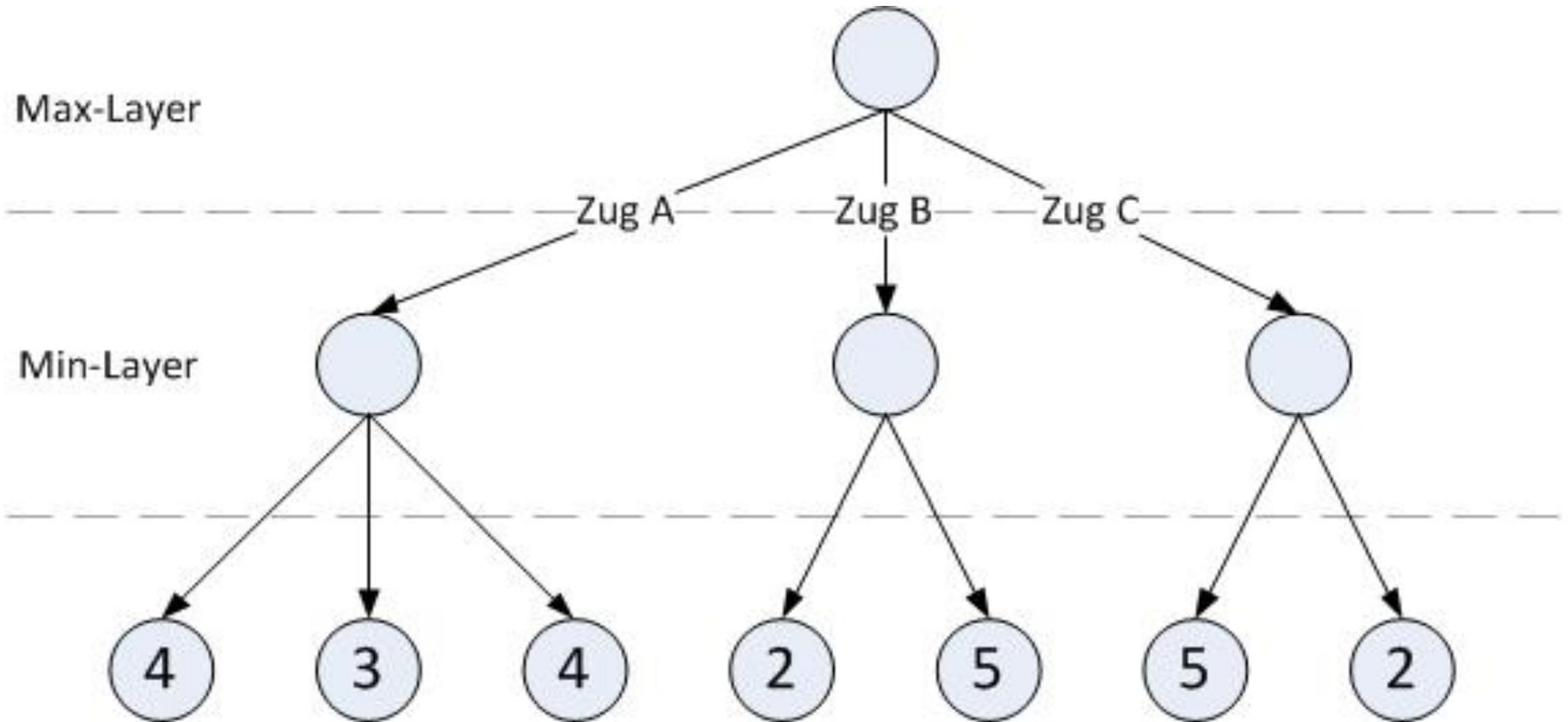


- Teilgebiet der KI
- Suche im Zustandsraum
 - Bewertung von Zuständen
 - Planung aufgrund von Bewertungen
 - z.B. Minimax
- Relationen und Abhängigkeiten bei Zuständen
 - Erlauben Optimierung der Suche
 - z.B. Alpha-Beta-Pruning

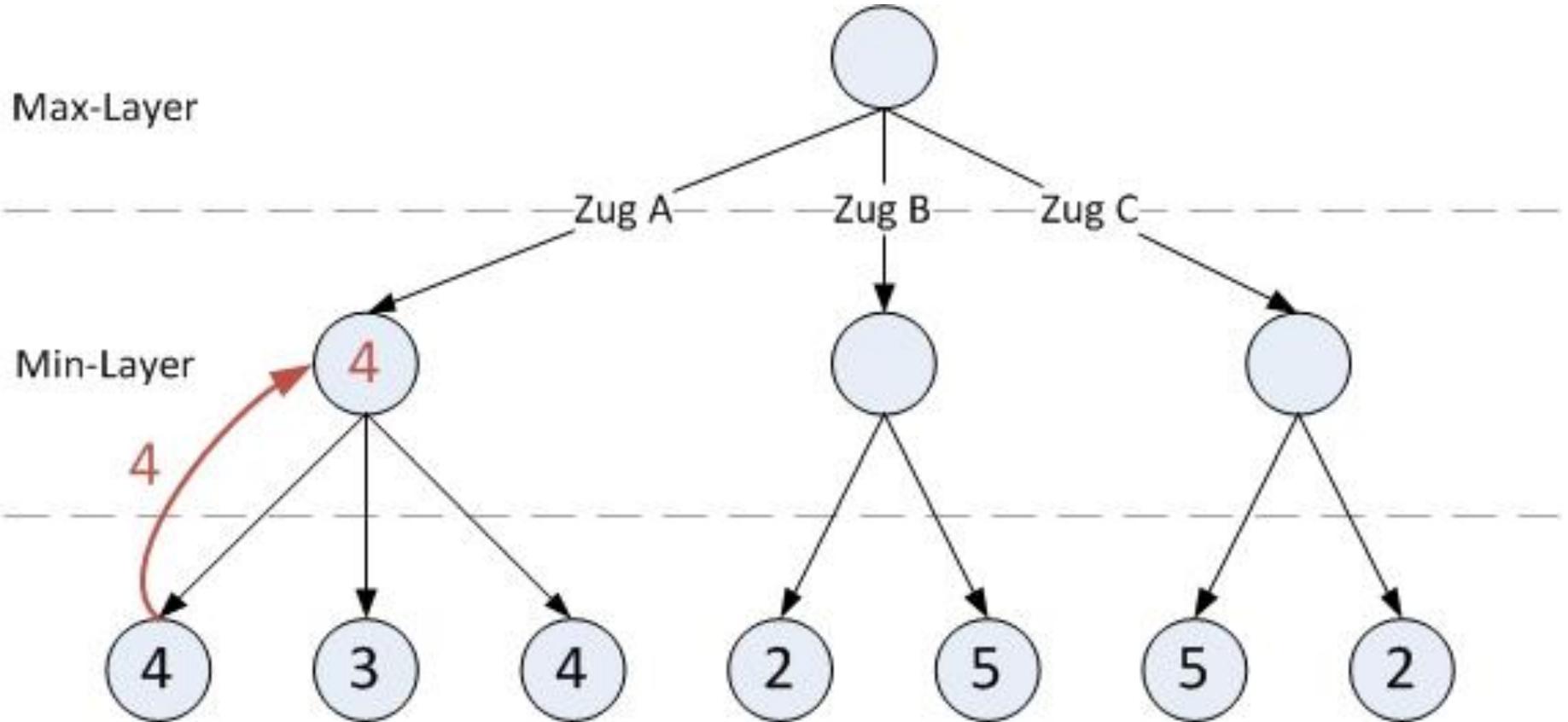


- Algorithmus zur Strategiesuche
- Liefert optimale Strategie bei Nullsummenspielen
 - „Wenn ich gewinne, verlierst du“
 - Der Gegner spielt optimal
- Arbeitet auf Spielbäumen
- Nicht effizient, aber wirksam
 - Kombinatorische Explosion

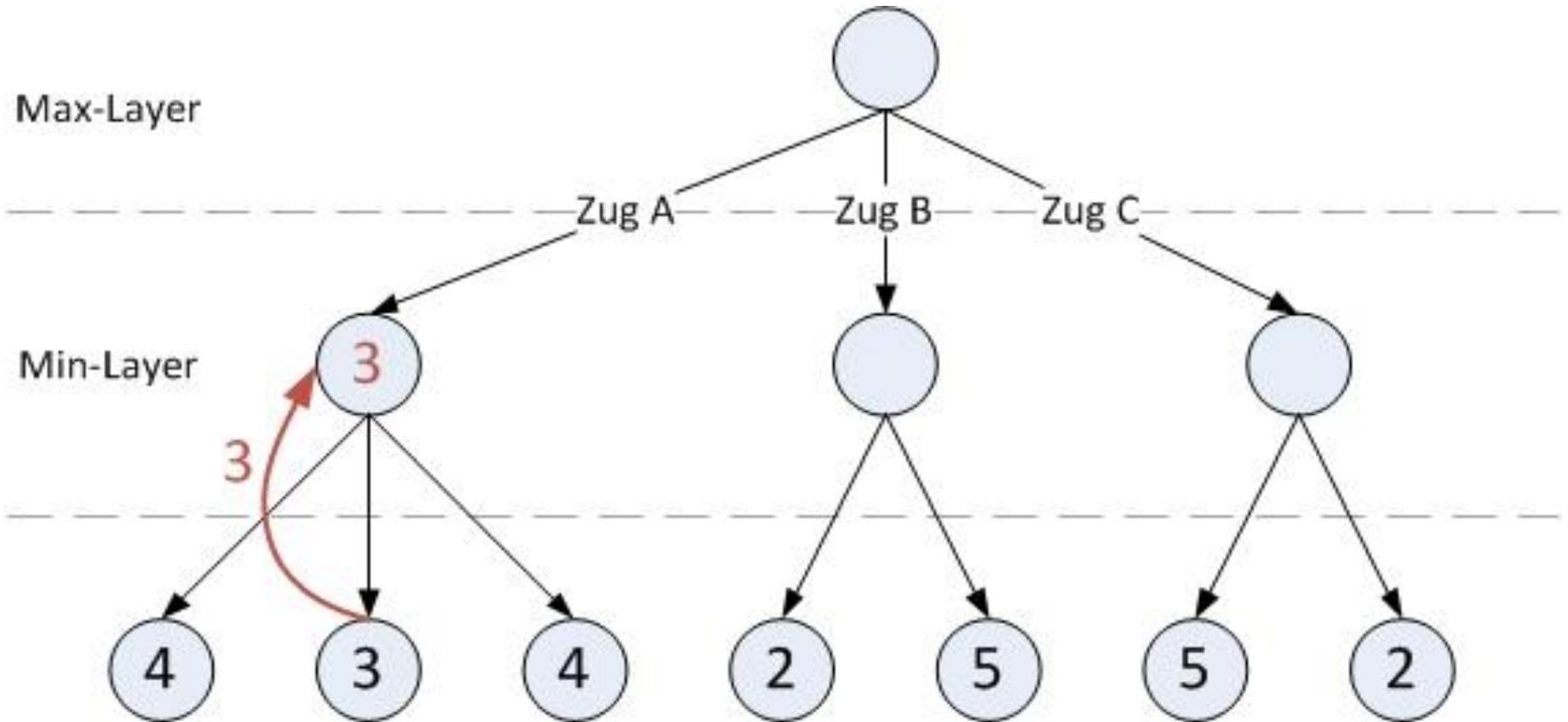
Minimax



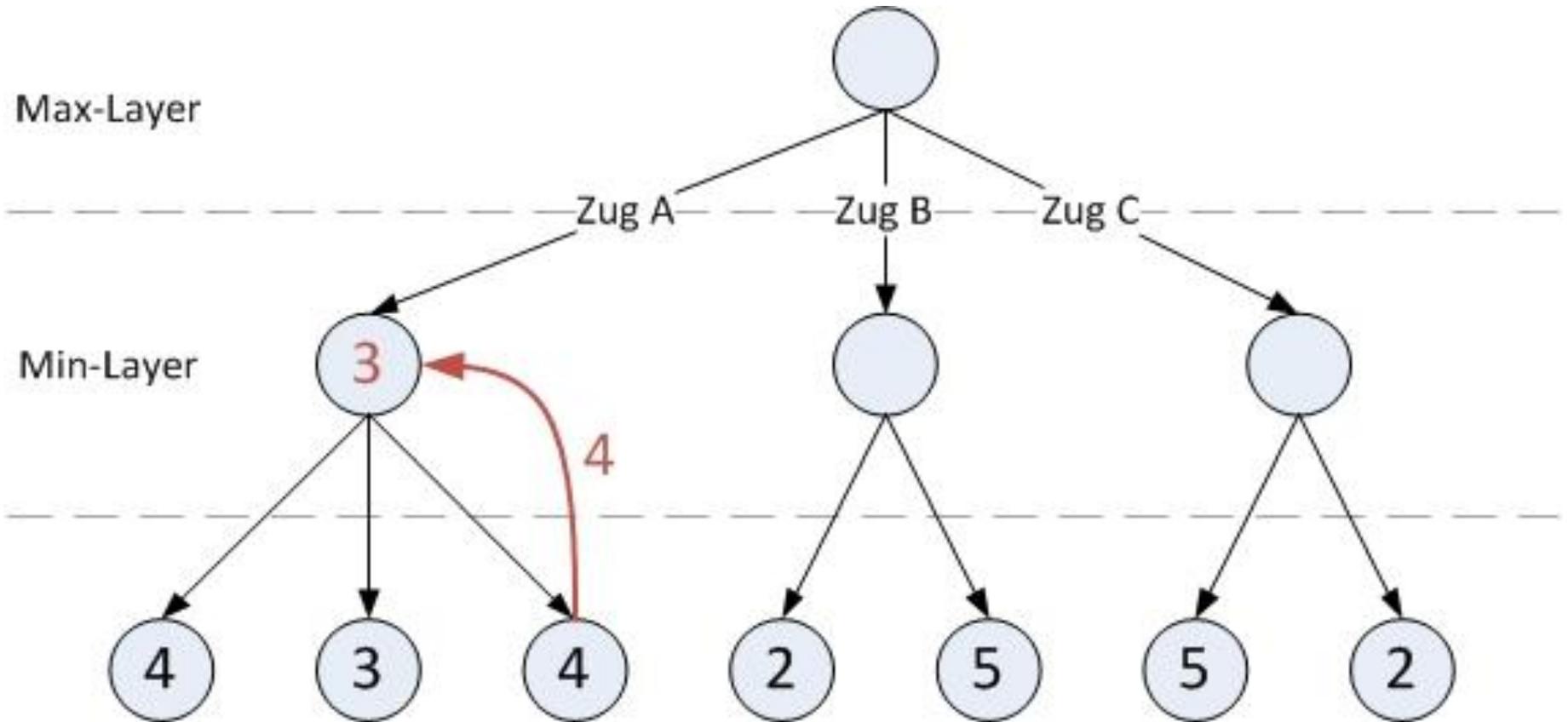
Minimax



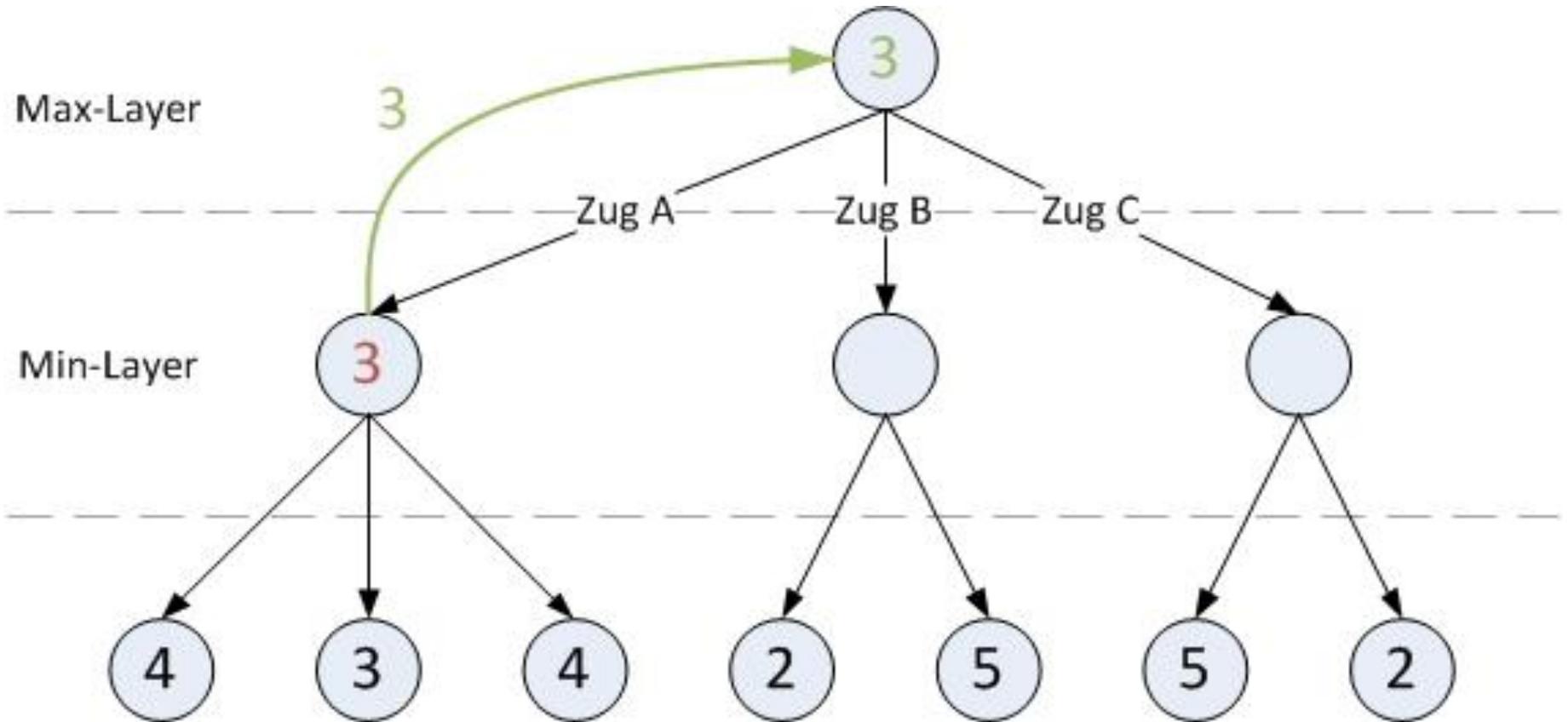
Minimax



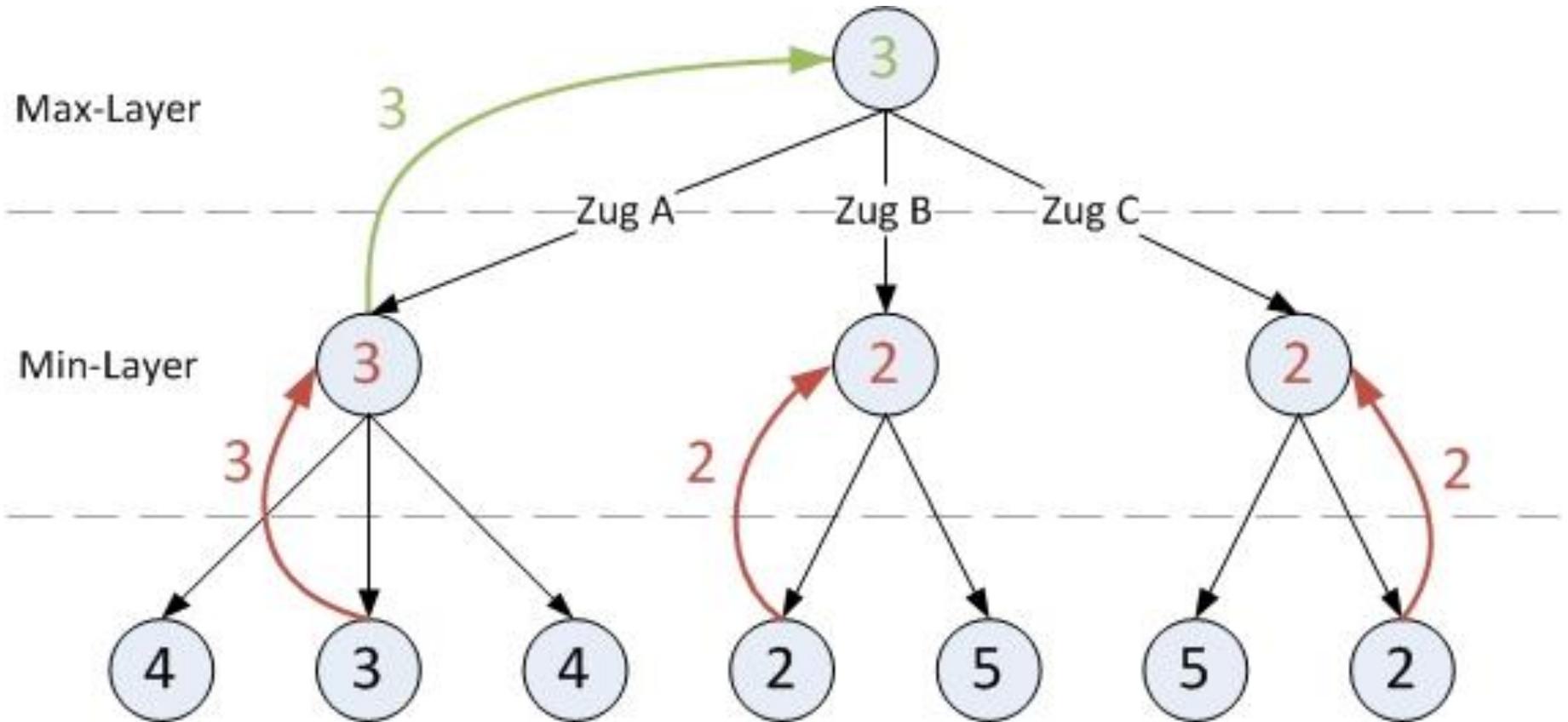
Minimax



Minimax



Minimax





```
function minimax(Node node)

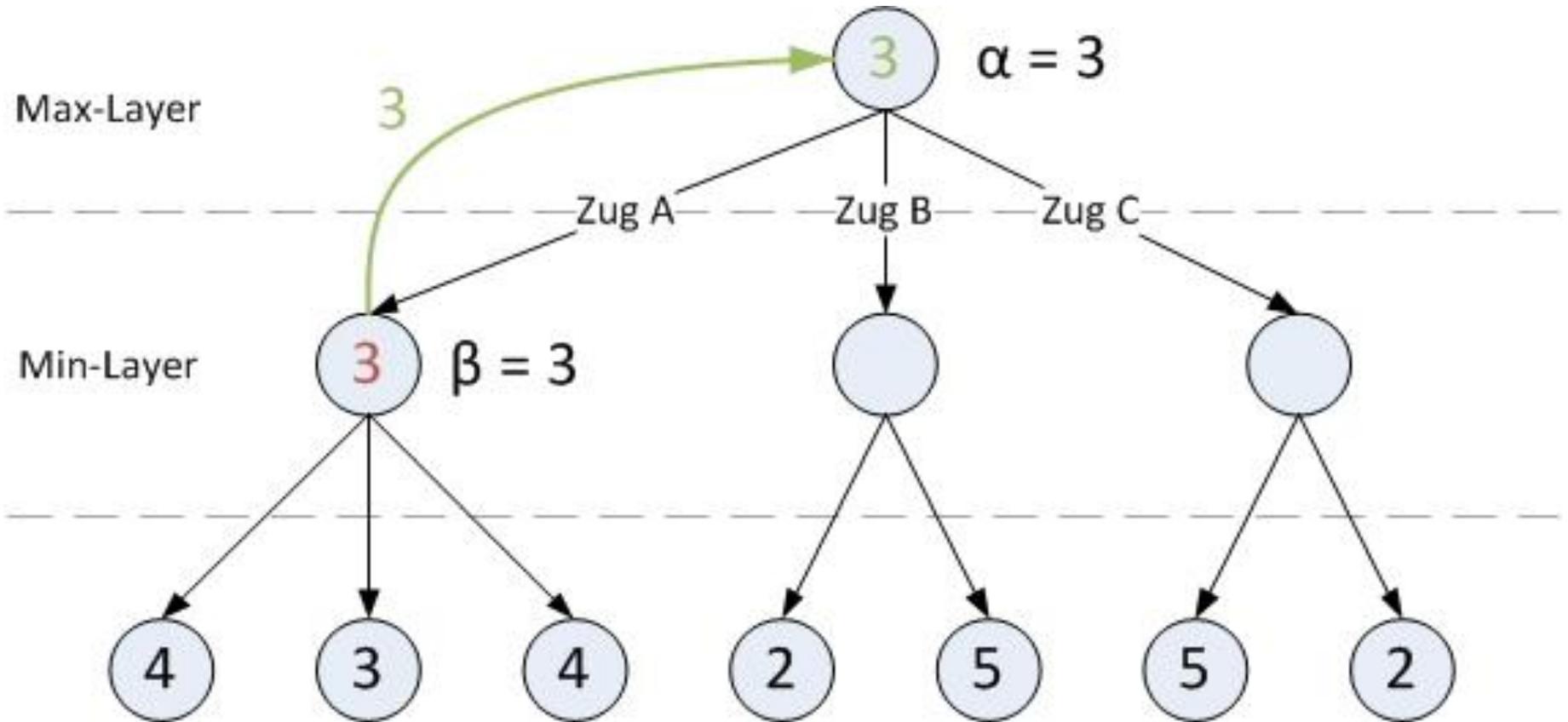
    if (isLeaf(node))
        return utility(node)

    if (isMax(node))
        for each Child c in node
            value = max(value, minimax(c))
        return value
    else
        for each Child c in node
            value = min(value, minimax(c))
        return value
```

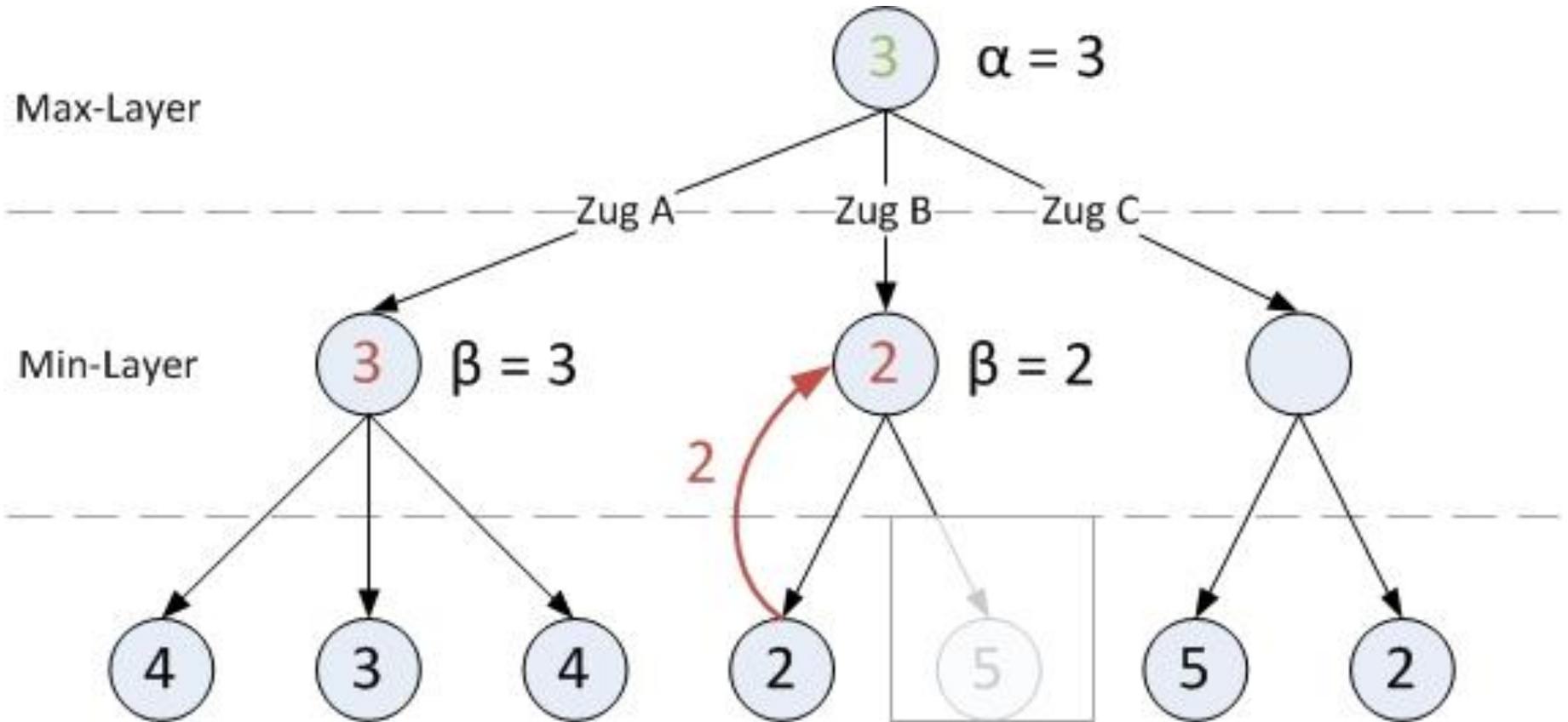
Alpha-Beta-Pruning

- Werte für Teilbäume speichern
 - Gespeicherte Werte vergleichen
 - Suche entsprechend beschränken
- ⇒ Beschränkung des Suchraums

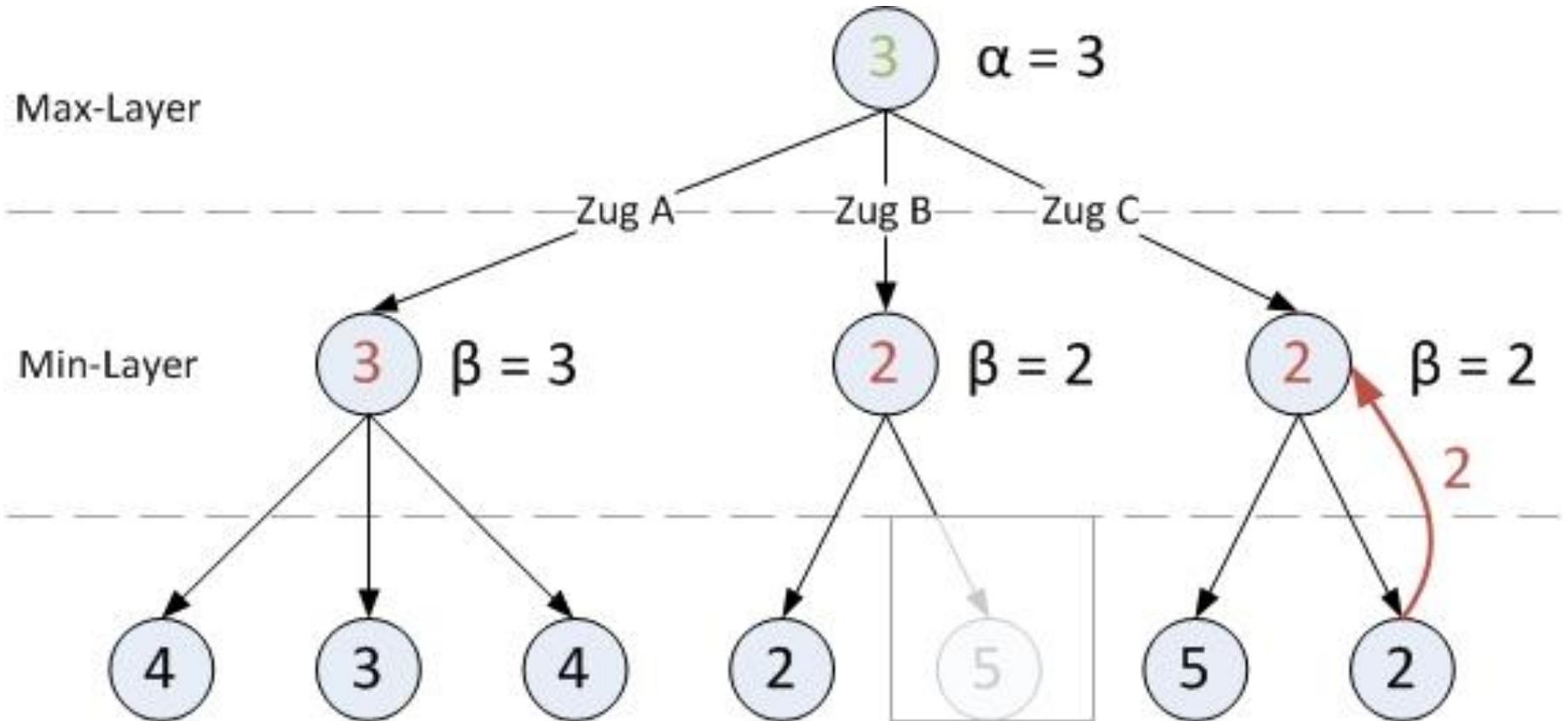
Alpha-Beta-Pruning



Alpha-Beta-Pruning



Alpha-Beta-Pruning



Pseudocode



```
function alphaBeta(Node node, int alpha, int beta)

    if (isLeaf(node))
        return utility(node)

    if (isMax(node))
        for each Child c in node
            value = max(value, alphaBeta(c, alpha, beta))

            if (value > beta)
                end for each loop
            else if (value > alpha)
                alpha = value

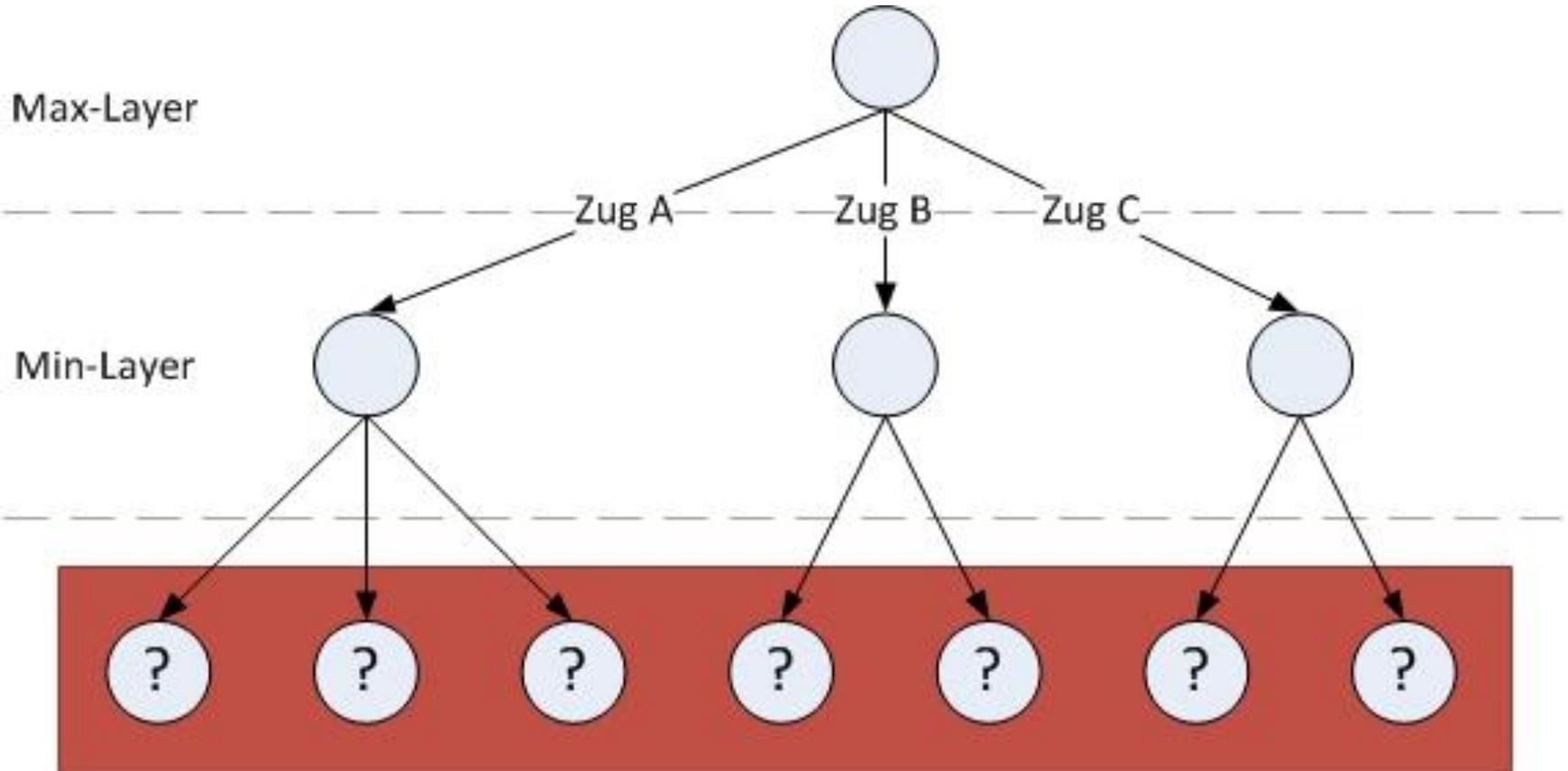
        return value
    else
        for each Child c in node
            value = min(value, alphaBeta(c, alpha, beta))

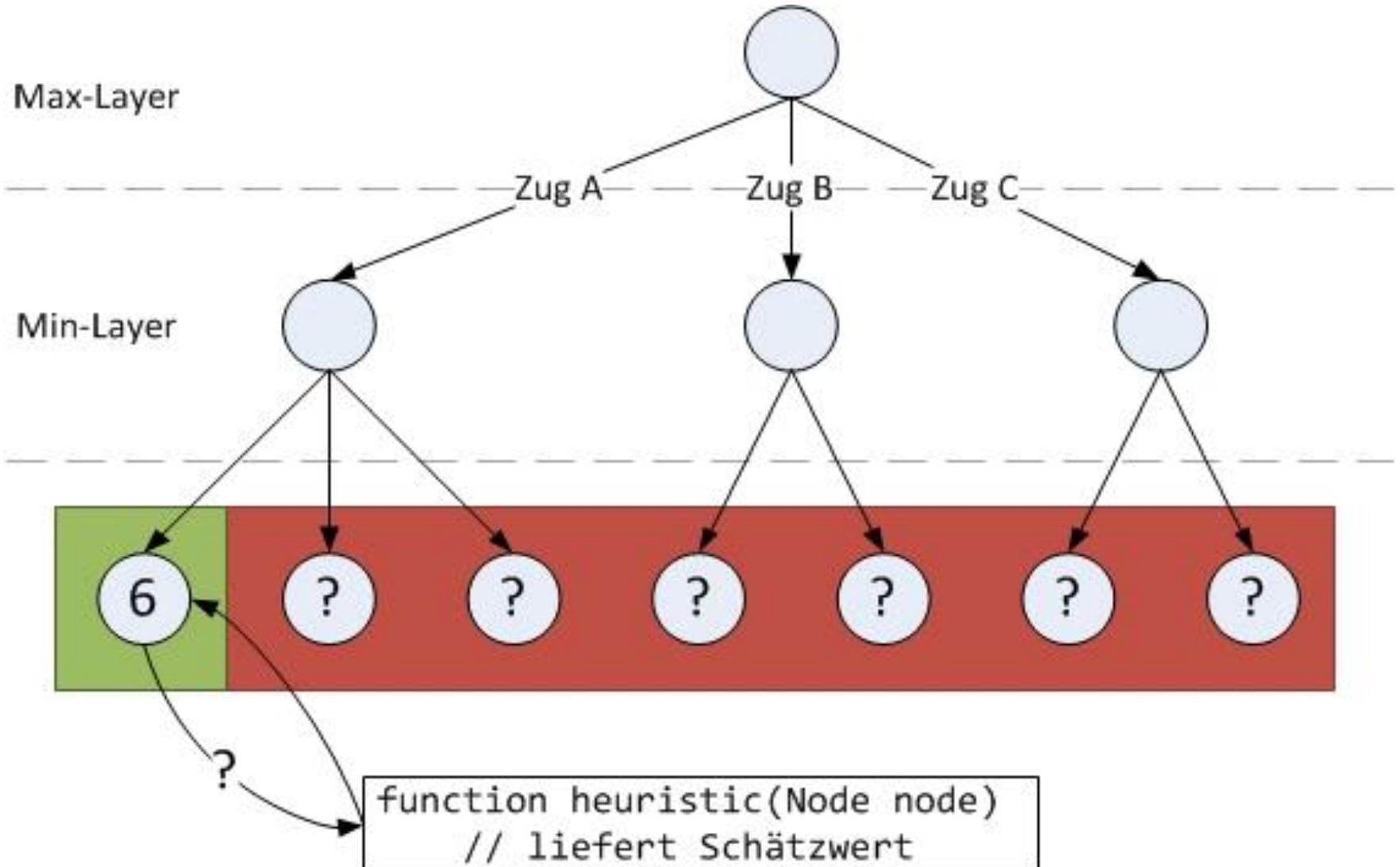
            if (value < alpha)
                end for each loop
            else if (value < beta)
                beta = value

        return value
```



- Für die meisten Probleme wird der Suchraum zu groß
- Beschränkung der Suchtiefe
 - ⇒ Blätter des Suchbaums haben keine Werte
- Anwendung von Heuristik-Funktionen
 - Verwendung von domänenspezifischer Information
 - Schätzung der Zustandswerte
 - Bildet Zustand auf geschätzten Wert ab







```
function minimaxHeuristic(Node node)

    if (isLeaf(node))
        return utility(node)

    if (maxDepthReached(node))
        return heuristic(node)

    if (isMax(node))
        for each Child c in node
            value = max(value, minimaxHeuristic(c))
        return value
    else
        for each Child c in node
            value = min(value, minimaxHeuristic(c))
        return value
```