UNIVERSITÀ DELLA CALABRIA

Coalitional Game Theory: Computational Issues

Sebastiano A. Piccolo

Roadmap

1. Computational limitations of coalitional game theory
2. Strategies to tackle computational limitations
3. Airport Game
4. Montecarlo Approximation
5. Compact representations

Limitations of naive approaches

- Naïve characteristic function approaches use $\mathcal{O}(2^N)$ memory space

Limitations of naive approaches

- Naïve characteristic function approaches use $\mathcal{O}(2^N)$ memory space

```
v = {
    frozenset(['A']): 0,
    frozenset(['B']): 0,
    frozenset(['C']): 0,
    frozenset(['A', 'B']): 750,
    frozenset(['A', 'C']): 750,
    frozenset(['B', 'C']): 0,
    frozenset(['A', 'B', 'C']): 1000
}
```

Limitations of naive approaches

- Naïve characteristic function approaches use $\mathcal{O}(2^N)$ memory space
- Algorithms exhibit $\mathcal{O}(2^N)$ computational complexity

Limitations of naive approaches

- Naïve characteristic function approaches use $\mathcal{O}(2^N)$ memory space
- Algorithms exhibit $\mathcal{O}(2^N)$ computational complexity

```python
# This function checks if an outcome for a given game is stable
def is_stable(outcome, characteristic_function):
    return all(
        [
            sum([outcome[player] for player in coalition]) >=
                characteristic_function[coalition]
            for coalition in characteristic_function
        ]
    )
```

Limitations of naive approaches

- Naïve characteristic function approaches use $\mathcal{O}(2^N)$ memory space
- Algorithms exhibit $\mathcal{O}(2^N)$ computational complexity

```
def shapley_value(player, characteristic_function):
    player = set([player])
    N = len(max(characteristic_function, key = len))
    shapley_val = 0
    for coalition in characteristic_function:
        S = len(coalition)
        marginal_contribution = characteristic_function[coalition] - \
            characteristic_function[coalition - player]
        if marginal_contribution:
            shapley_val += ((factorial(N - S) * factorial(S - 1)) / \
                factorial(N)) * marginal_contribution
    return round(shapley_val, 10)
```

Limitations of naive approaches

- Naïve characteristic function approaches use $\mathcal{O}(2^N)$ memory space
- Algorithms exhibit $\mathcal{O}(2^N)$ computational complexity

Can we do better?

Strategies to attack computational issues

- Focusing on specific types of games, which we can solve analytically
  - Low memory footprint
  - Polynomial algorithms

Strategies to attack computational issues

- Focusing on specific types of games, which we can solve analytically
  - Low memory footprint
  - Polynomial algorithms
- Approximation algorithms (e.g. Montecarlo approximation)
  - Polynomial time algorithms
  - Approximation error is small in practice

Strategies to attack computational issues

- Focusing on specific types of games, which we can solve analytically
  - Low memory footprint
  - Polynomial algorithms
- Approximation algorithms (e.g. Montecarlo approximation)
  - Polynomial time algorithms
  - Approximation error is small in practice
- Compact representations for the characteristic function
  - Low memory footprint
  - High expressivity (can represent many/all games)
  - Polynomial algorithms

Airport game

There are N airlines. Each airline needs a runway of a certain length for their planes. Since airlines can share a runway, they can join forces to build one runway which is big enough for everyone, and split the cost. How should they split the cost? (Hint: Shapley value!)

Airport game

We have a set $N = \{1,2,\dots,n\}$ of players, each associated to a cost c_i such that $c_1 < c_2 < \cdots < \mathrm{c_n}$. The characteristic function is:

$$v(S) = \max_{\mathrm{i} \in S} c_i \quad \forall S \subseteq N$$

The Shapley value for the player i, in this game is given by:

$$\phi_i = \sum_{j=1}^{i} \frac{c_j - c_{j-1}}{n - j + 1} \quad \forall i \in N; \quad c_0 = 0$$

Airport game (example)

4 players; costs = [8, 11, 13, 18]

Player	Adding 1	Adding 2	Adding 3	Adding 4	Shapley value
Marginal cost					
Cost to P1					
Cost to P2					
Cost to P3					
Cost to P4					

Airport game (example)

4 players; costs = [8, 11, 13, 18]

Player	Adding 1	Adding 2	Adding 3	Adding 4	Shapley value
Marginal cost	8	3	2	5	
Cost to P1					
Cost to P2					
Cost to P3					
Cost to P4					

Airport game (example)

4 players; costs = [8, 11, 13, 18]

Player	Adding 1	Adding 2	Adding 3	Adding 4	Shapley value
Marginal cost	**8**	**3**	**2**	**5**	
Cost to P1	2				
Cost to P2	2				
Cost to P3	2				
Cost to P4	2				

Airport game (example)

4 players; costs = [8, 11, 13, 18]

Player	Adding 1	Adding 2	Adding 3	Adding 4	Shapley value
Marginal cost	**8**	**3**	**2**	**5**	
Cost to P1	2				
Cost to P2	2	1			
Cost to P3	2	1			
Cost to P4	2	1			

Airport game (example)

4 players; costs = [8, 11, 13, 18]

Player	Adding 1	Adding 2	Adding 3	Adding 4	Shapley value
Marginal cost	**8**	**3**	**2**	**5**	
Cost to P1	2				
Cost to P2	2	1			
Cost to P3	2	1	1		
Cost to P4	2	1	1		

Airport game (example)

4 players; costs = [8, 11, 13, 18]

Player	Adding 1	Adding 2	Adding 3	Adding 4	Shapley value
Marginal cost	**8**	**3**	**2**	**5**	
Cost to P1	2				
Cost to P2	2	1			
Cost to P3	2	1	1		
Cost to P4	2	1	1	5	

Airport game (example)

4 players; costs = [8, 11, 13, 18]

Player	Adding 1	Adding 2	Adding 3	Adding 4	Shapley value
Marginal cost	**8**	**3**	**2**	**5**	
Cost to P1	2				**2**
Cost to P2	2	1			**3**
Cost to P3	2	1	1		**4**
Cost to P4	2	1	1	5	**9**

Montecarlo approximation

Suppose we have a probability distribution $P(X)$ and we want to compute $P(x)$.

Montecarlo approximation

Suppose we have a probability distribution $P(X)$ and we want to compute $P(x)$.

IDEA → We can approximate $P(x)$ using sample frequencies.

Montecarlo approximation

Suppose we have a probability distribution $P(X)$ and we want to compute $P(x)$.

IDEA → We can approximate $P(x)$ using sample frequencies.

IDEA → Generate a sample D of size M from $P(X)$ and compute $P(x)$ as:

$$P_D(X = x) = \frac{M_{X=x}}{M}$$

Shapley value: exact formula

The shapley value for a player i is the average marginal contribution of the player i over all possible coalitions.

$$\phi(i,v) = \frac{1}{|N|!} \sum_{\pi \in \Pi_N} v(B(\pi,i) \cup \{i\}) - v(B(\pi,i))$$

Where:

Π_N is the set of all possible permutations of N

$B(\pi,i)$ is the set of predecessors on i in the permutation π

Shapley value: Montecarlo approximation

$$\tilde{\phi}(i,v) = \frac{1}{m} \sum_{\pi \in \mathcal{P}} v(B(\pi,i) \cup \{i\}) - v(B(\pi,i))$$

Where:

$\mathcal{P} \subset \Pi_N$ is a subset of all possible permutations of N

$B(\pi,i)$ is the set of predecessors on i in the permutation π

Shapley value: Montecarlo approximation

Input: v → characteristic function; m → number of samples

def MC_Shapley(v, m):

  $\tilde{\phi}_i = 0 \ \forall i \in N$

  for $k = 1 \dots m$:

    $\pi_k =$ random permutation of N

    for $i = 1 \dots n$:

      $sv = v(B(\pi, i) \cup \{i\}) - v(B(\pi, i))$

      $\tilde{\phi}_i \mathrel{+}= sv$

  for $k = 1 \dots n$:

    $\tilde{\phi}_i = \frac{\tilde{\phi}_i}{m}$

  return $\tilde{\phi}_1, \tilde{\phi}_2, \dots, \tilde{\phi}_n$

Compact representations

Compact representations aim at reducing the memory footprint of the characteristic function, typically using network structures.

Compact representations

Compact representations aim at reducing the memory footprint of the characteristic function, typically using network structures.

The value of a coalition will no longer be accessed in $\mathcal{O}(1)$ as it happens with the naive representation, but will be obtained in polynomial time.

Compact representations

Compact representations aim at reducing the memory footprint of the characteristic function, typically using network structures.

The value of a coalition will no longer be accessed in $\mathcal{O}(1)$ as it happens with the naive representation, but will be obtained in polynomial time.

Compact representations, by leveraging the additive property of the Shapley value, let us compute the Shapley value in polynomial time.

Induced subgraph games (ISG)

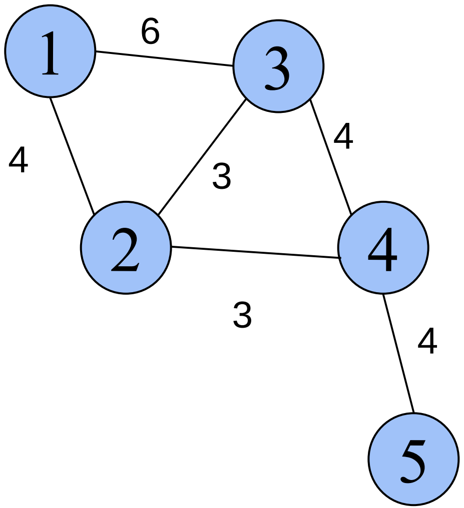
Players are nodes in a graph. Edges are coalitions of two players. Weights on edges are the value of the coalition. ISGs can represent the following characteristic function:

$$v(C) = \sum_{i,j \subseteq C} w_{ij}$$

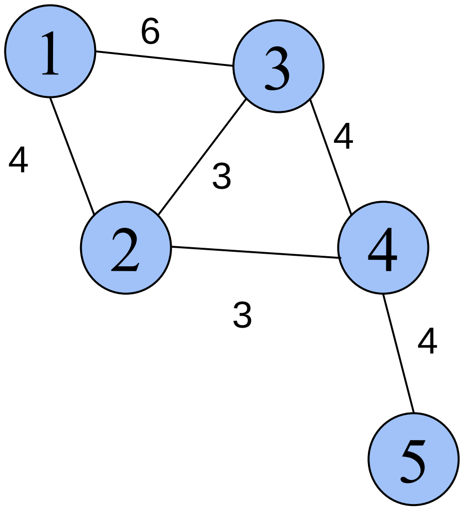
We can compute the Shapley value for player i as follows:

$$\phi_i = w_{ii} + \frac{1}{2} \sum_{j \in \Gamma(i)} w_{ij}$$

Induced subgraph games (ISG)

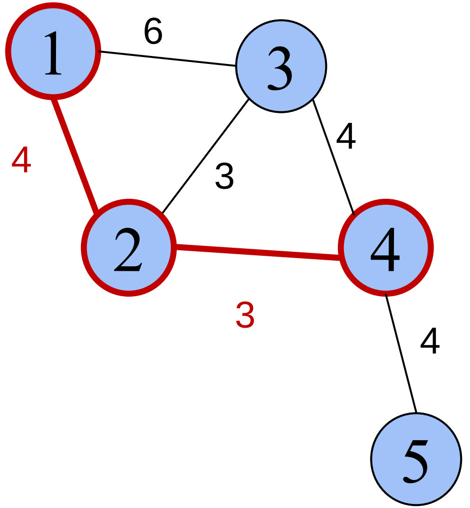


Induced subgraph games (ISG)



Obtaining the value of a coalition

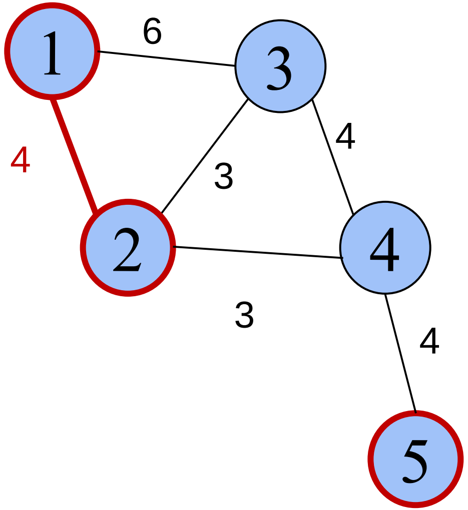
Induced subgraph games (ISG)



Obtaining the value of a coalition

$v(1,2,4) = 4 + 3 = 7$

Induced subgraph games (ISG)

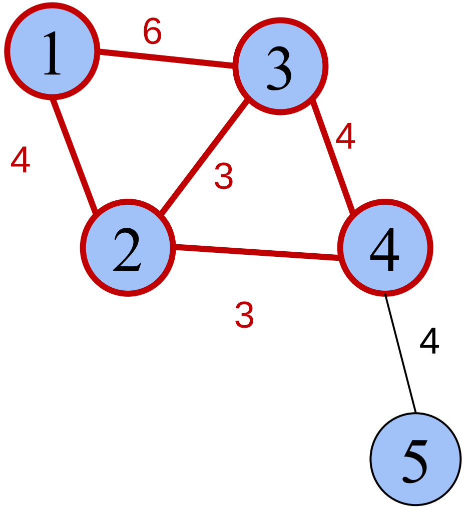


Obtaining the value of a coalition

$v(1,2,4) = 4 + 3 = 7$

$v(1,2,5) = 4$

Induced subgraph games (ISG)



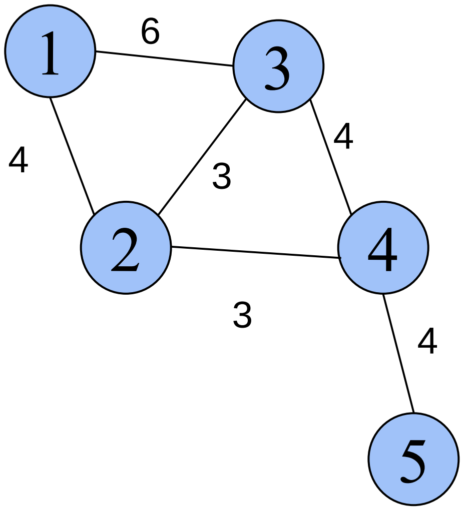
Obtaining the value of a coalition

$v(1,2,4) = 4 + 3 = 7$

$v(1,2,5) = 4$

$v(1,2,3,4) = 4 + 6 + 3 + 4 + 3 = 20$

Induced subgraph games (ISG)



Obtaining the value of a coalition

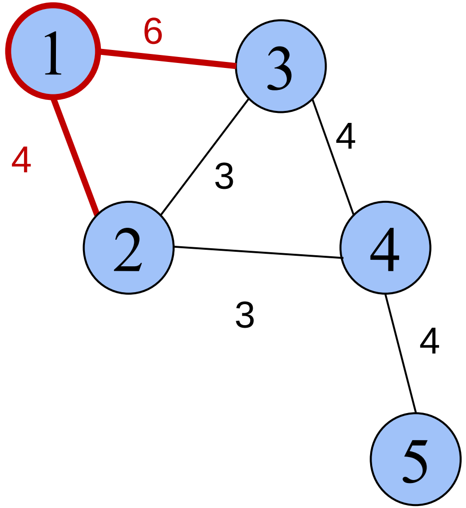
$v(1,2,4) = 4 + 3 = 7$

$v(1,2,5) = 4$

$v(1,2,3,4) = 4 + 6 + 3 + 4 + 3 = 20$

Shapley value

Induced subgraph games (ISG)



Obtaining the value of a coalition

$v(1,2,4) = 4 + 3 = 7$

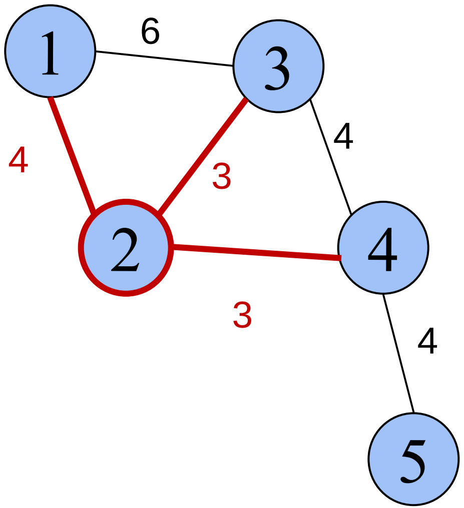
$v(1,2,5) = 4$

$v(1,2,3,4) = 4 + 6 + 3 + 4 + 3 = 20$

Shapley value

$$\phi_1 = \frac{1}{2}(6 + 4) = 5$$

Induced subgraph games (ISG)



Obtaining the value of a coalition

$v(1,2,4) = 4 + 3 = 7$

$v(1,2,5) = 4$

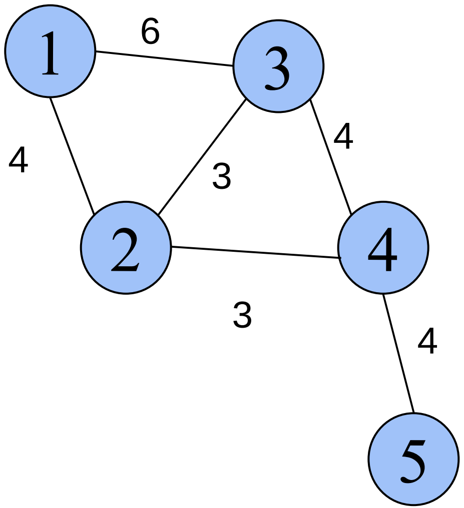
$v(1,2,3,4) = 4 + 6 + 3 + 4 + 3 = 20$

Shapley value

$\phi_1 = \frac{1}{2}(6+4) = 5$

$\phi_2 = \frac{1}{2}(4+3+3) = 5$

Induced subgraph games (ISG)



Obtaining the value of a coalition

$v(1,2,4) = 4 + 3 = 7$

$v(1,2,5) = 4$

$v(1,2,3,4) = 4 + 6 + 3 + 4 + 3 = 20$

Shapley value

$$\phi_1 = \frac{1}{2}(6 + 4) = 5$$

$$\phi_2 = \frac{1}{2}(4 + 3 + 3) = 5$$

$$\phi_3 = \frac{1}{2}(6 + 3 + 4) = 6.5$$

$$\phi_4 = \frac{1}{2}(4 + 3 + 4) = 5.5$$

$$\phi_5 = \frac{1}{2}4 = 2$$

Marginal Contribution Nets (MC-Nets)

IDEA → represent the characteristic function as a set of rules in the form

pattern ⟶ value

The pattern is a boolean formula over N

The value associated to a pattern is its marginal contribution

If pattern is in the form $\{a \wedge b \wedge \cdots \wedge c\}$ and the associated value can beeither negative or positive, we can represent *any* game.

Marginal Contribution Nets (MC-Nets)

Take as an example the following game:

$$\{a \wedge b\} \longrightarrow 5$$
$$\{b\} \longrightarrow 2$$

It represents the following characteristic function:

$$v(\emptyset) = 0; v(\{a\}) = 0; v(\{b\}) = 2; v(\{a, b\}) = 5 + 2 = 7$$

Marginal Contribution Nets (MC-Nets)

$$\{a \wedge b\} \longrightarrow 5$$
$$\{b\} \longrightarrow 2$$

Obtaining the value of a coalition

$\{a\}$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition
$\{a\}$

$\{a, b\} \nsubseteq \{a\}$ The rule does not apply.

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition
$\{a\}$

$\{b\} \nsubseteq \{a\}$ The rule does not apply.

Marginal Contribution Nets (MC-Nets)

$\{a \land b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$

$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$

$\{b\} =$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition
$\{a\} = 0$
$\{b\} =$

$\{a, b\} \nsubseteq \{b\}$ The rule does not apply.

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$

$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$

$\{b\} =$

$\{b\} \subseteq \{b\}$ The rule does apply!

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} =$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition
$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} = 5$

$\{a, b\} \subseteq \{a, b\}$ The rule does apply!

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$

$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$

$\{b\} = 2$

$\{a, b\} = 5 + 2$

$\{b\} \subseteq \{a, b\}$ The rule does apply!

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Shapley values are computed as:

$$\phi_i = \sum_{\varphi \longrightarrow x \in rs_i} \frac{x}{|\varphi|}$$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} = 7$

Shapley value

$\phi_a =$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} = 7$

Shapley values are computed as:

$$\phi_i = \sum_{\varphi \longrightarrow x \in rs_i} \frac{x}{|\varphi|}$$

Shapley value

$$\phi_a = \frac{5}{2}$$

$$\{a, b\} \supseteq \{a\}$$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} = 7$

Shapley values are computed as:

$$\phi_i = \sum_{\varphi \longrightarrow x \in rs_i} \frac{x}{|\varphi|}$$

Shapley value

$\phi_a = \frac{5}{2} + 0$

$\{b\} \not\supseteq \{a\}$

Marginal Contribution Nets (MC-Nets)

$\{a \land b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} = 7$

Shapley values are computed as:

$$\phi_i = \sum_{\varphi \longrightarrow x \in rs_i} \frac{x}{|\varphi|}$$

Shapley value

$\phi_a = \frac{5}{2} = 2.5$
$\phi_b =$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} = 7$

Shapley values are computed as:

$$\phi_i = \sum_{\varphi \longrightarrow x \in rs_i} \frac{x}{|\varphi|}$$

Shapley value

$\phi_a = \frac{5}{2} = 2.5$

$\phi_b = \frac{5}{2}$

$\{a, b\} \supseteq \{b\}$

Marginal Contribution Nets (MC-Nets)

$\{a \wedge b\} \longrightarrow 5$
$\{b\} \longrightarrow 2$

Obtaining the value of a coalition

$\{a\} = 0$
$\{b\} = 2$
$\{a, b\} = 7$

Shapley values are computed as:

$$\phi_i = \sum_{\varphi \longrightarrow x \in rs_i} \frac{x}{|\varphi|}$$

Shapley value

$$\phi_a = \frac{5}{2} = 2.5$$

$$\phi_b = \frac{5}{2} + \frac{2}{1} = 4.5$$

$$\{b\} \supseteq \{b\}$$

Summary

Technique	Pros	Cons
Particular types of games (Airport game)	+ Compact representation + Fast Shapley value computation	- Limited expressivity
Montecarlo Approximation	+ Fast Shapley value computation + Convergence properties + Applicable to any game and any representation	- Not exact - No way to exactly determine m (in practice $m \in [1000, 10000]$ gives good results).
Induced subgraph games	+ Can represent many games + Fast Shapley value computation + Low memory footprint	- Not complete: cannot represent any game.
MC-Nets	+ Complete: can represent any game + Fast Shapley value computation + Typically low memeory footprint	- Worst-case memory space required is still $\mathcal{O}(2^N)$ - Offsets the computation of the marginal contributions: we need to pre-compute them