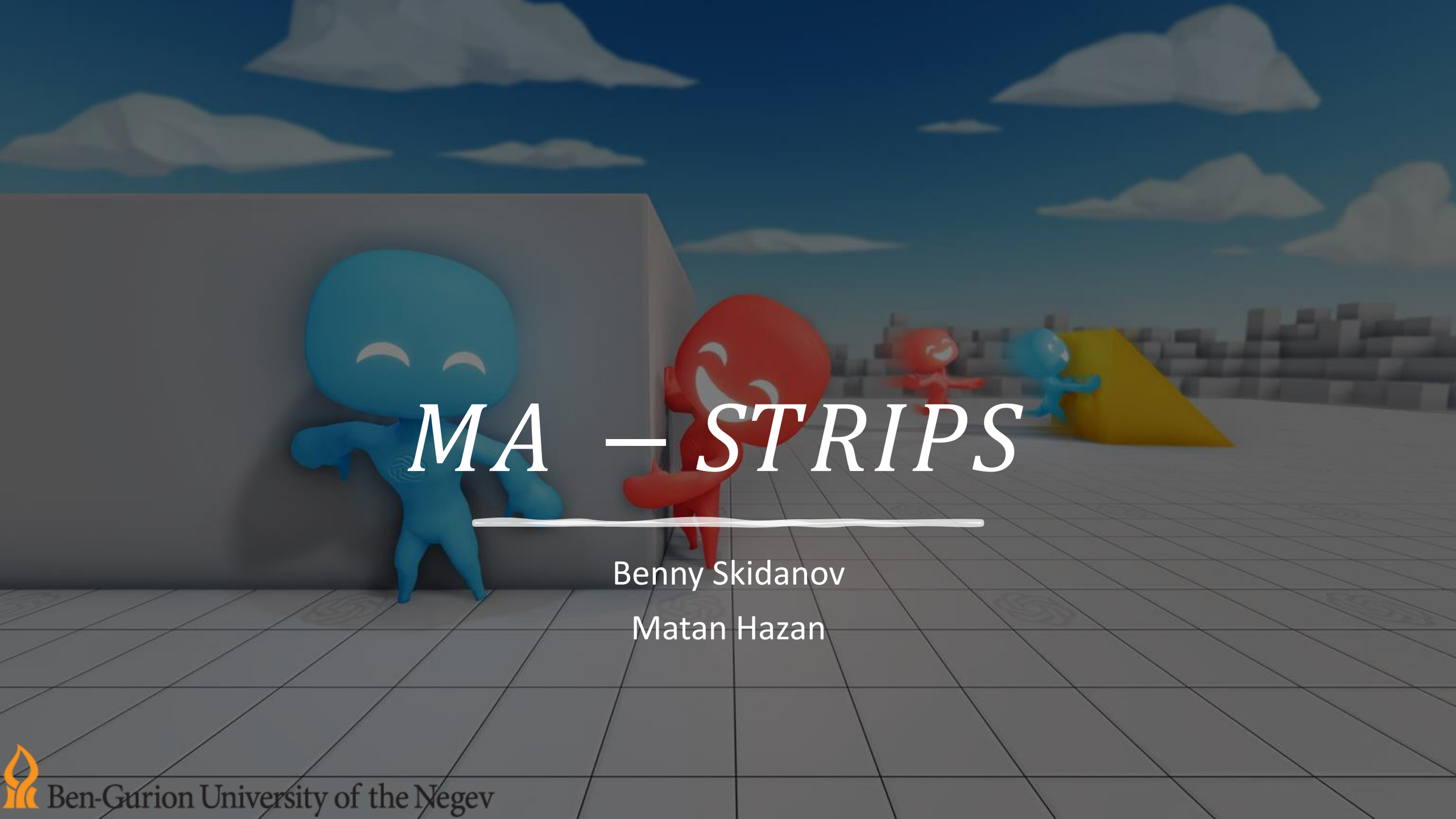


# MA – STRIPS

Benny Skidanov

Matan Hazan

Ben-Gurion University of the Negev

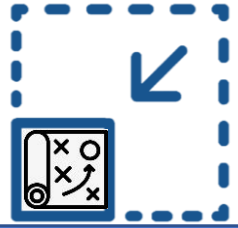
# Our Inspiration

# Our Project

**1**

**What to build**

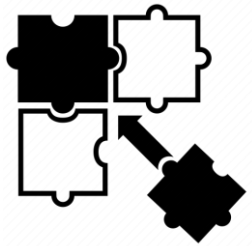
Represented by a $3D$ matrix and a matching Visualization



**2**

**Reduction**

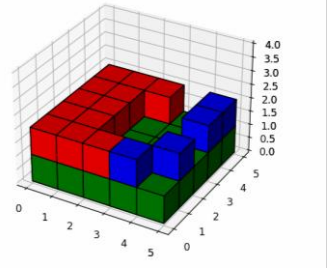
Convert to $MA-STRIPS$ problem



**3**

**Solve**

Send to $MA-STRIPS$ solver



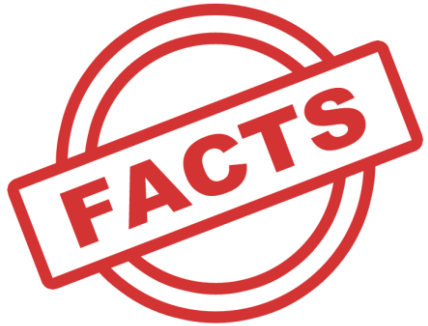
**4**

**Visualize**

Present the solution in convenient way using ***Python Plots***

# Modeling

```
(:predicates
    (atagent ?a - agent ?src - location)
    (atbrick ?dst - location)
    (freecell ?src - location)
    (freeagent ?a - agent)
    (adjacent_horizontal ?src - location ?dst - location)
    (adjacent_vertical_up ?loc - location ?loc - location)
)
```



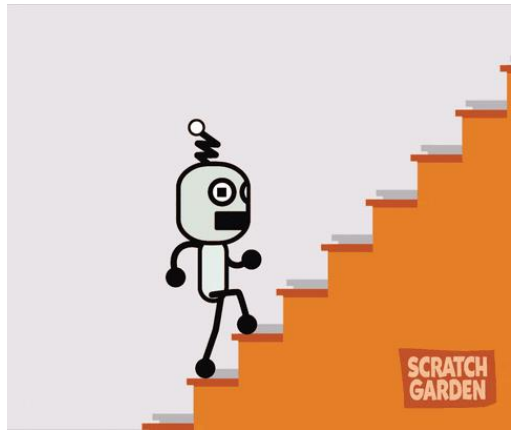
# Modeling

```
(:action agent_move
    :parameters (?a - agent ?src - location ?belowdst - location ?dst - location)
    :precondition (and
        (atagent ?a ?src)
        (freecell ?dst)
        (adjacent_horizontal ?src ?dst)
        (adjacent_vertical_up ?belowdst ?dst)
        (atbrick ?belowdst)
    )
    :effect (and
        (atagent ?a ?dst)
        (not (atagent ?a ?src))
        (not (freecell ?dst))
        (freecell ?src)
    )
)
```

# Modeling

```
(:action agent_climb
    :parameters (?a - agent ?src - location ?base - location ?dst - location)
    :precondition (and
        (atagent ?a ?src)
        (freecell ?dst)
        (atbrick ?base)
        (adjacent_horizontal ?src ?base)
        (adjacent_vertical_up ?base ?dst)
    )
    :effect (and
        (atagent ?a ?dst)
        (not (atagent ?a ?src))
        (not (freecell ?dst))
        (freecell ?src)
    )
)

(:action agent_slide
    :parameters (?a - agent ?src - location ?standingbase - location ?base - location ?dst - location)
    :precondition (and
        (atagent ?a ?src)
        (freecell ?dst)
        (atbrick ?base)
        (atbrick ?standingbase)
        (adjacent_horizontal ?standingbase ?dst)
        (adjacent_vertical_up ?base ?dst)
        (adjacent_vertical_up ?standingbase ?src)
    )
    :effect (and
        (atagent ?a ?dst)
        (not (atagent ?a ?src))
        (not (freecell ?dst))
        (freecell ?src)
    )
)
```

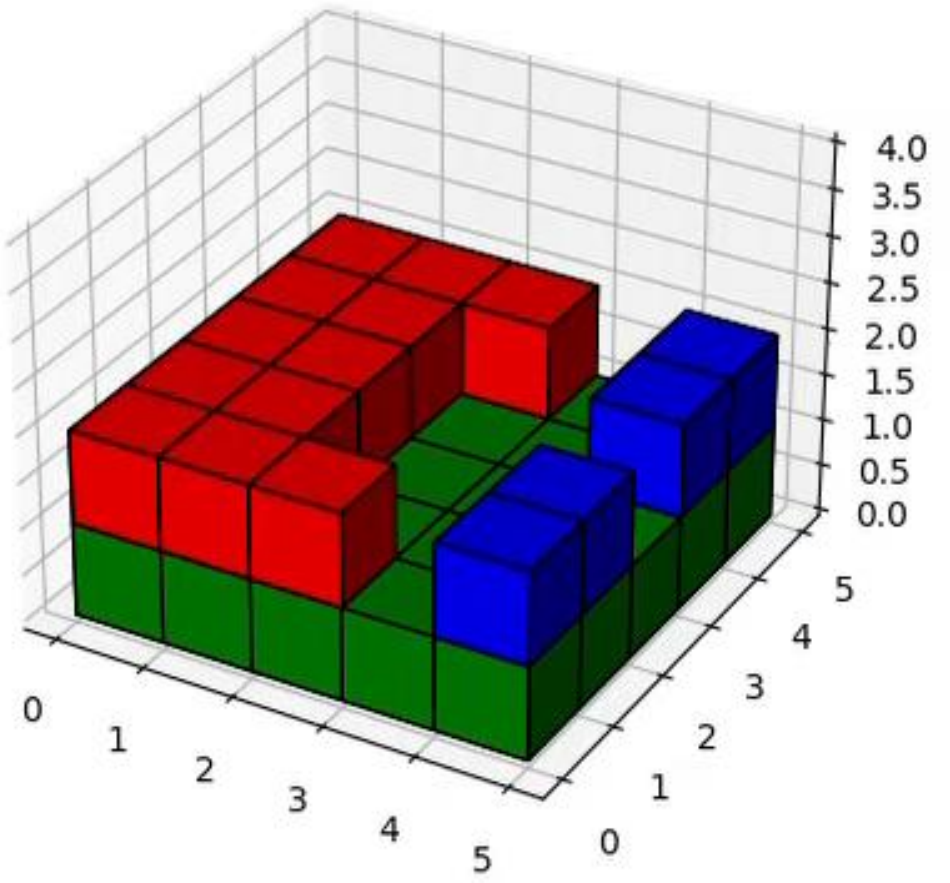


# Modeling

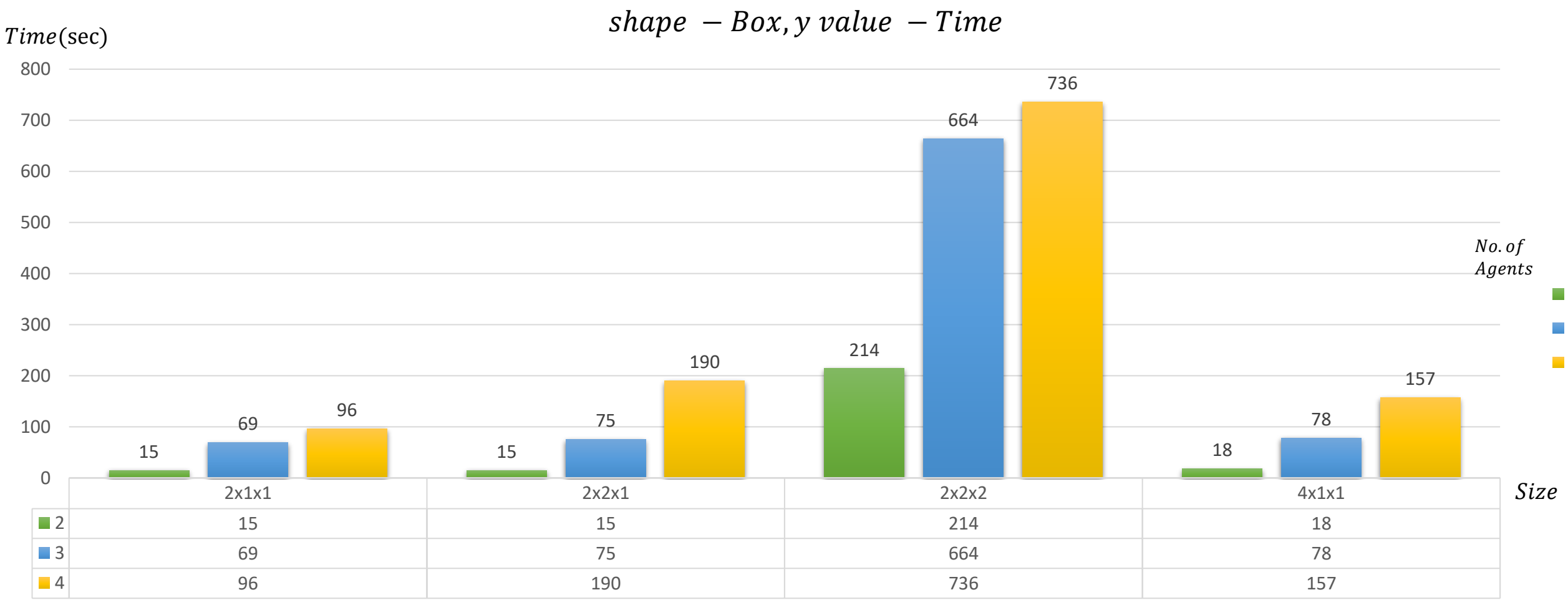
```
(:action pickup-brick
    :parameters (?a - agent ?src - location ?above - location ?dst - location)
    :precondition (and
        (atbrick ?dst)
        (adjacent_horizontal ?src ?dst)
        (adjacent_vertical_up ?dst ?above)
        (freecell ?above)
        (atagent ?a ?src)
        (freeagent ?a)
    )
    :effect (and
        (not (freeagent ?a))
        (not (atbrick ?dst))
        (freecell ?dst)
    )
)

(:action put-brick
    :parameters (?a - agent ?src - location ?below - location ?dst - location)
    :precondition (and
        (adjacent_horizontal ?src ?dst)
        (atagent ?a ?src)
        (adjacent_vertical_up ?below ?dst)
        (atbrick ?below)
        (not (freeagent ?a))
        (freecell ?dst)
    )
    :effect (and
        (freeagent ?a)
        (atbrick ?dst)
        (not (freecell ?dst))
    )
)
```

# Example

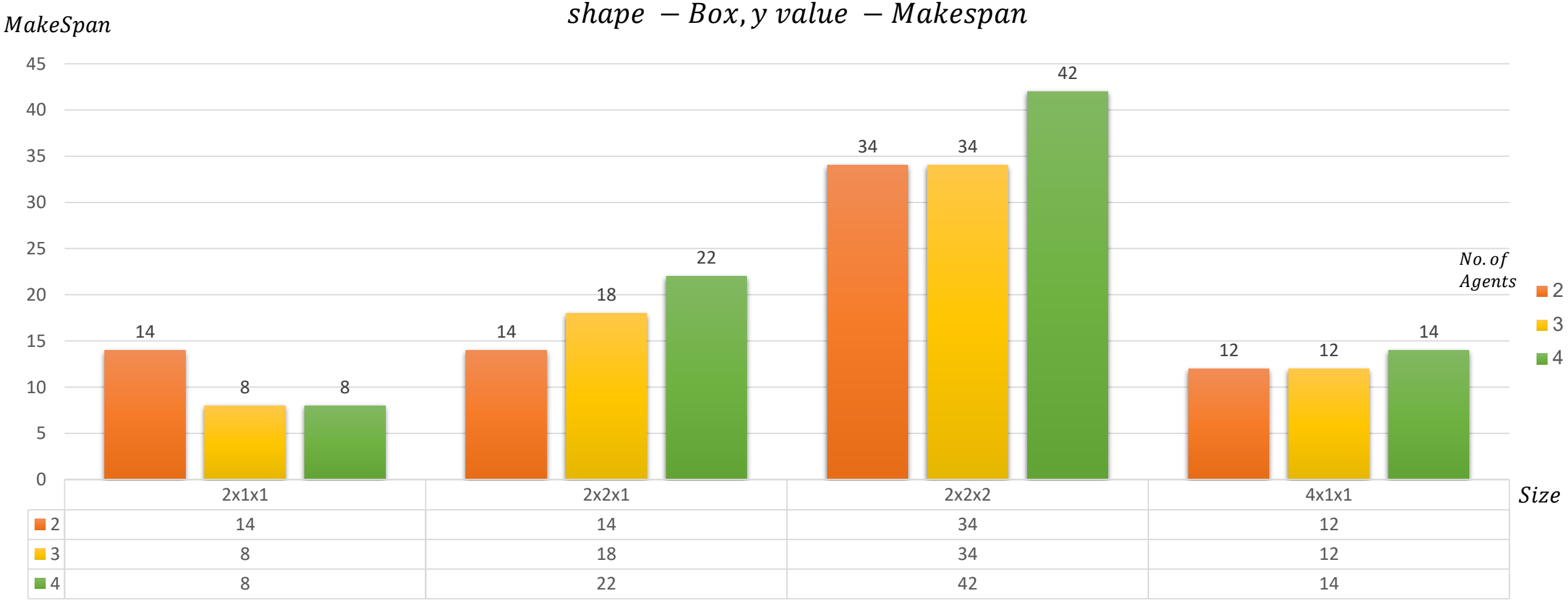


# Results




Ben-Gurion University of the Negev

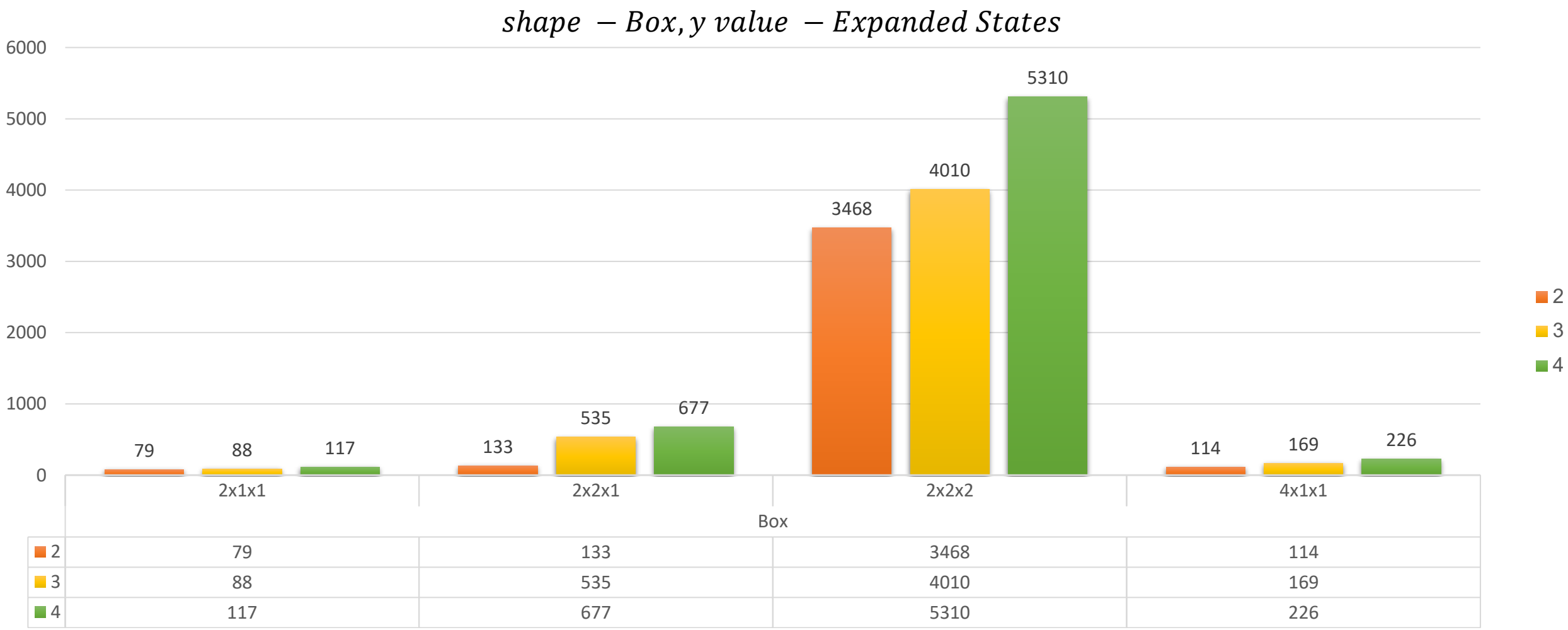
# Results




| | 2x1x1 | 2x2x1 | 2x2x2 | 4x1x1 |
|---|---|---|---|---|
| 2 | 14 | 14 | 34 | 12 |
| 3 | 8 | 18 | 34 | 12 |
| 4 | 8 | 22 | 42 | 14 |

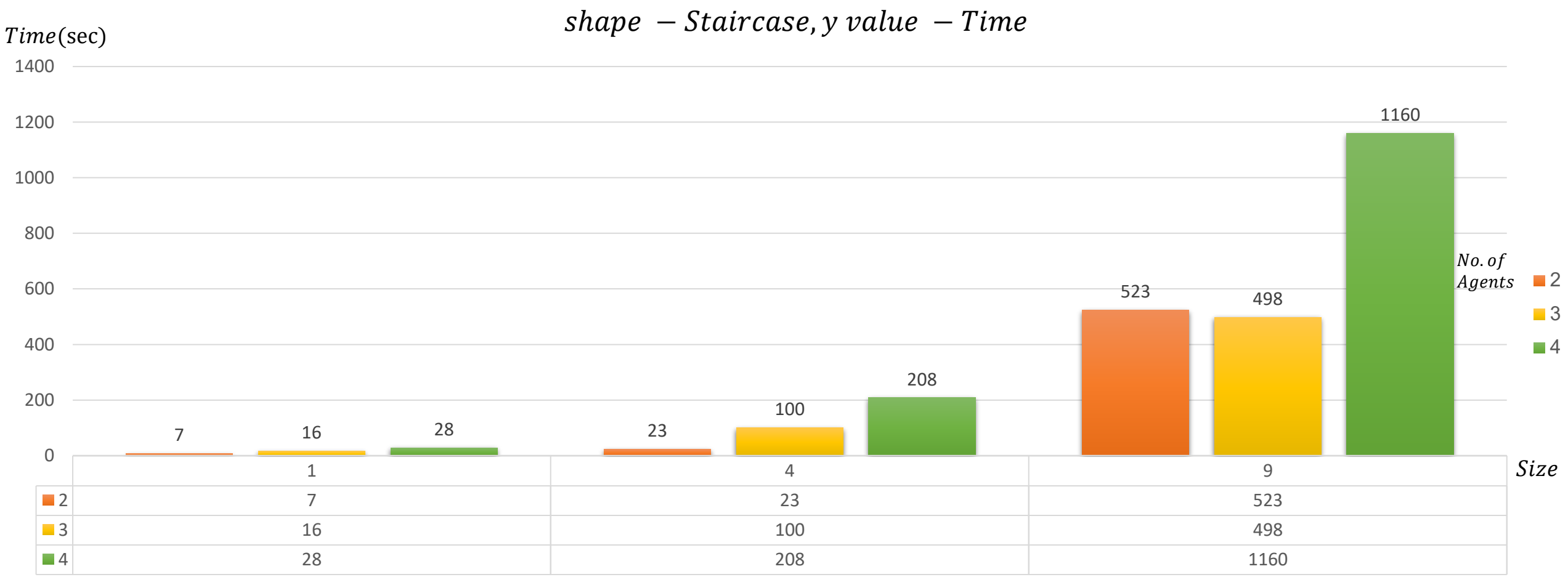
Ben-Gurion University of the Negev

# Results



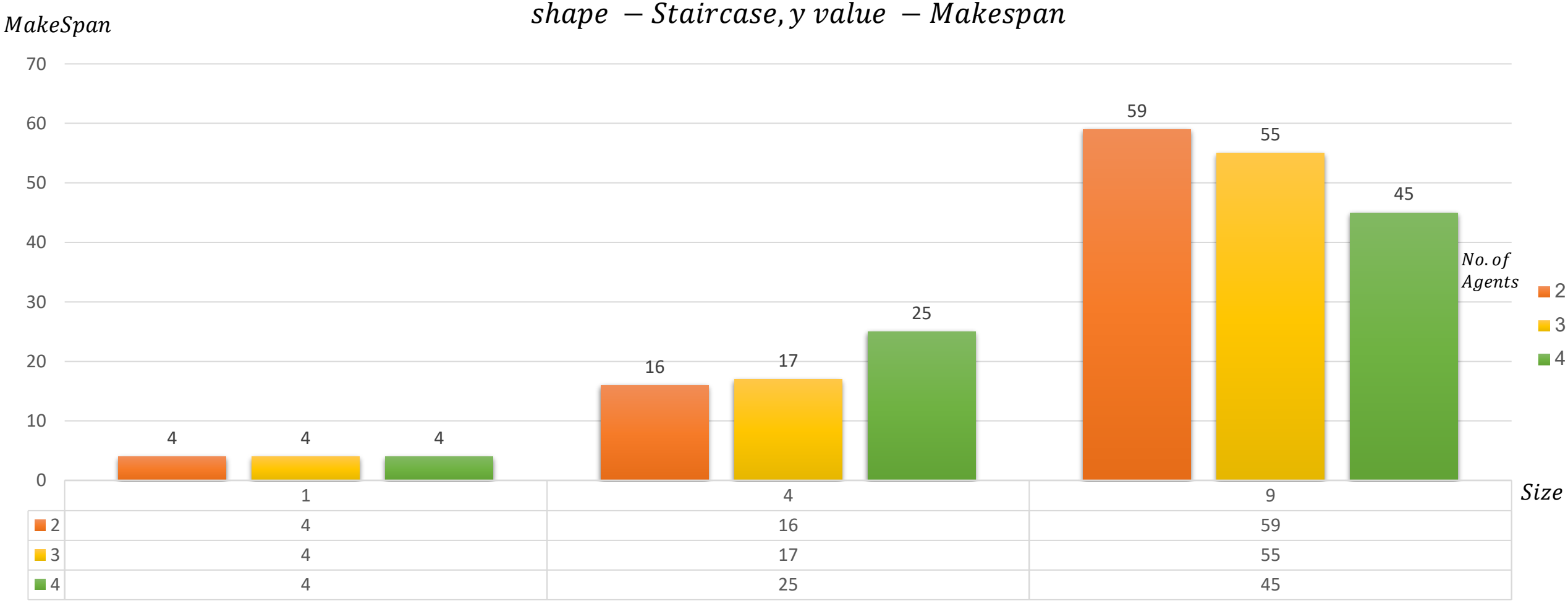

| | 2x1x1 | 2x2x1 | 2x2x2 | 4x1x1 |
|---|---|---|---|---|
| 2 | 79 | 133 | 3468 | 114 |
| 3 | 88 | 535 | 4010 | 169 |
| 4 | 117 | 677 | 5310 | 226 |

# Results




Ben-Gurion University of the Negev

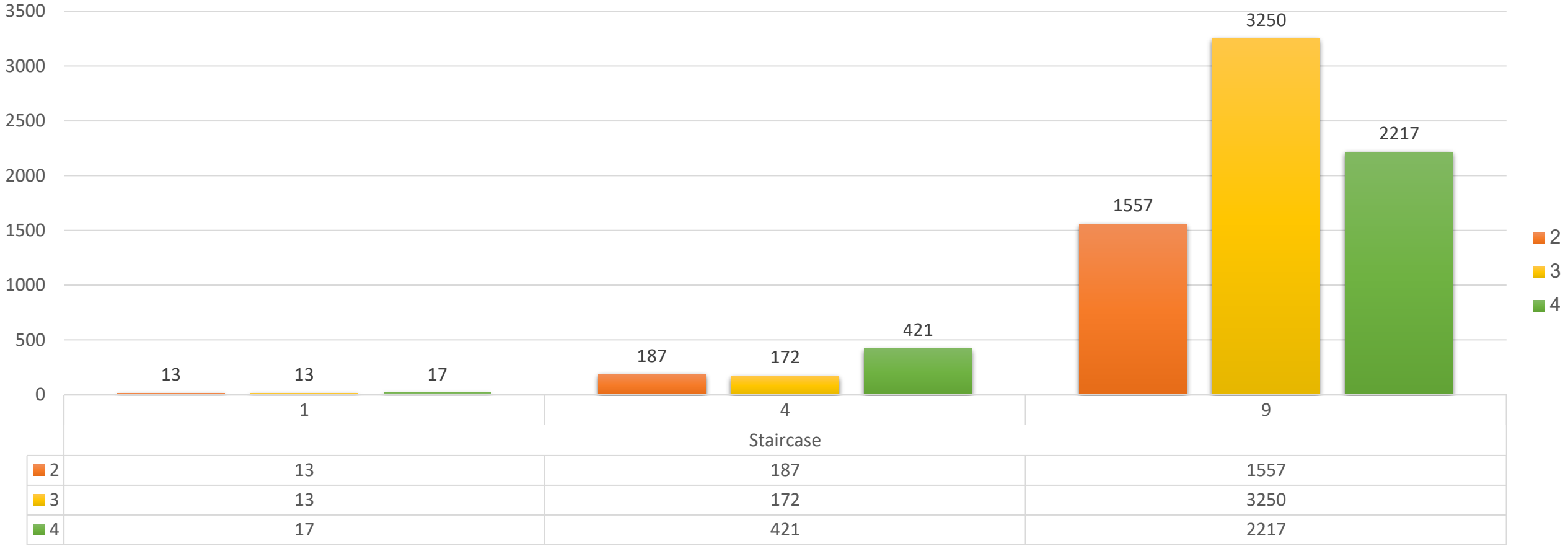
# Results




Ben-Gurion University of the Negev

# Results

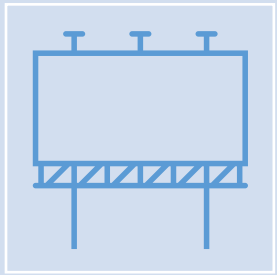
*shape – Staircase, y value – Expanded States*



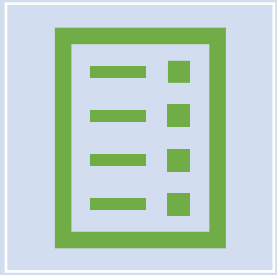

| | 1 | 4 | 9 |
|---|---|---|---|
| 2 | 13 | 187 | 1557 |
| 3 | 13 | 172 | 3250 |
| 4 | 17 | 421 | 2217 |

Ben-Gurion University of the Negev

# Future Work



*Use a better, stronger planner*



*Find a more optimal way to represnt a GRID*


THANK YOU
Any Questions?