Layout in Visual Modelling

Gitte Bluekens

S0110627

Inspired by D. Dubé, Graph Layout for domain-specific modeling (2006)

Table of contents

1. Introduction

- Layout algorithms necessary to make visually attractive models
- Implemented in AToMPM using transformations
- Language = PetriNets

2. Spring-Embedder algorithm

- Edges = springs
- Vertices = rings
- Pre-processing step recommended
  → improve convergence speed and quality
- Combination of repulsion, attraction and gravity

Repulsion algorithm

1. Calculate Manhattan and Euclidean distances
2. Calculate scalar force
3. Multiply force by 2D Manhattan distance vector

- avoid vertex overlaps
- generate large repulsive forces if overlap

Attraction algorithm

1. Calculate Manhattan and Euclidean distances
2. Calculate spring force
3. Multiply force by 2D Manhattan distance vector

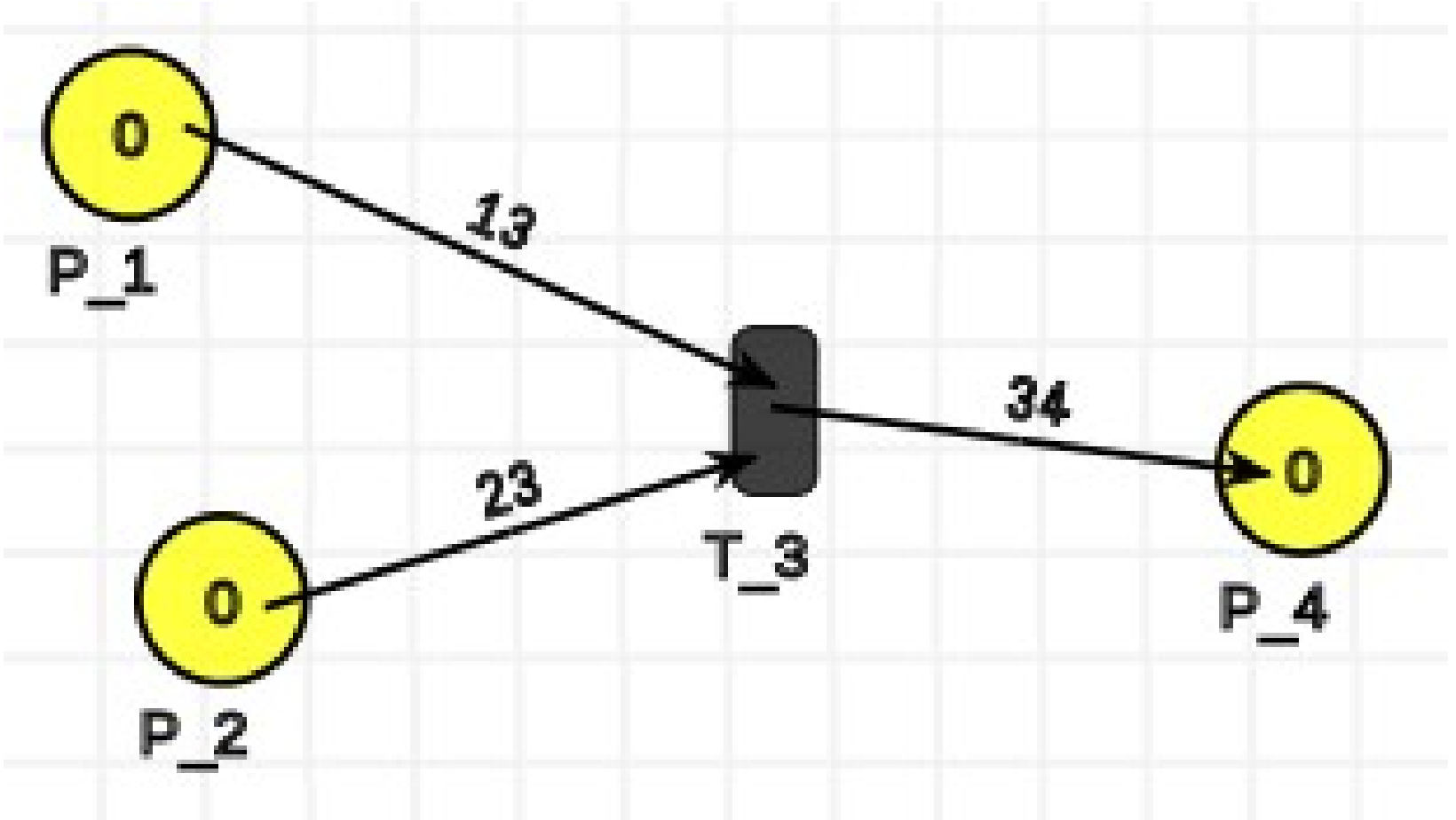
- Attract source- and targetvertex of edge

Gravity algorithm

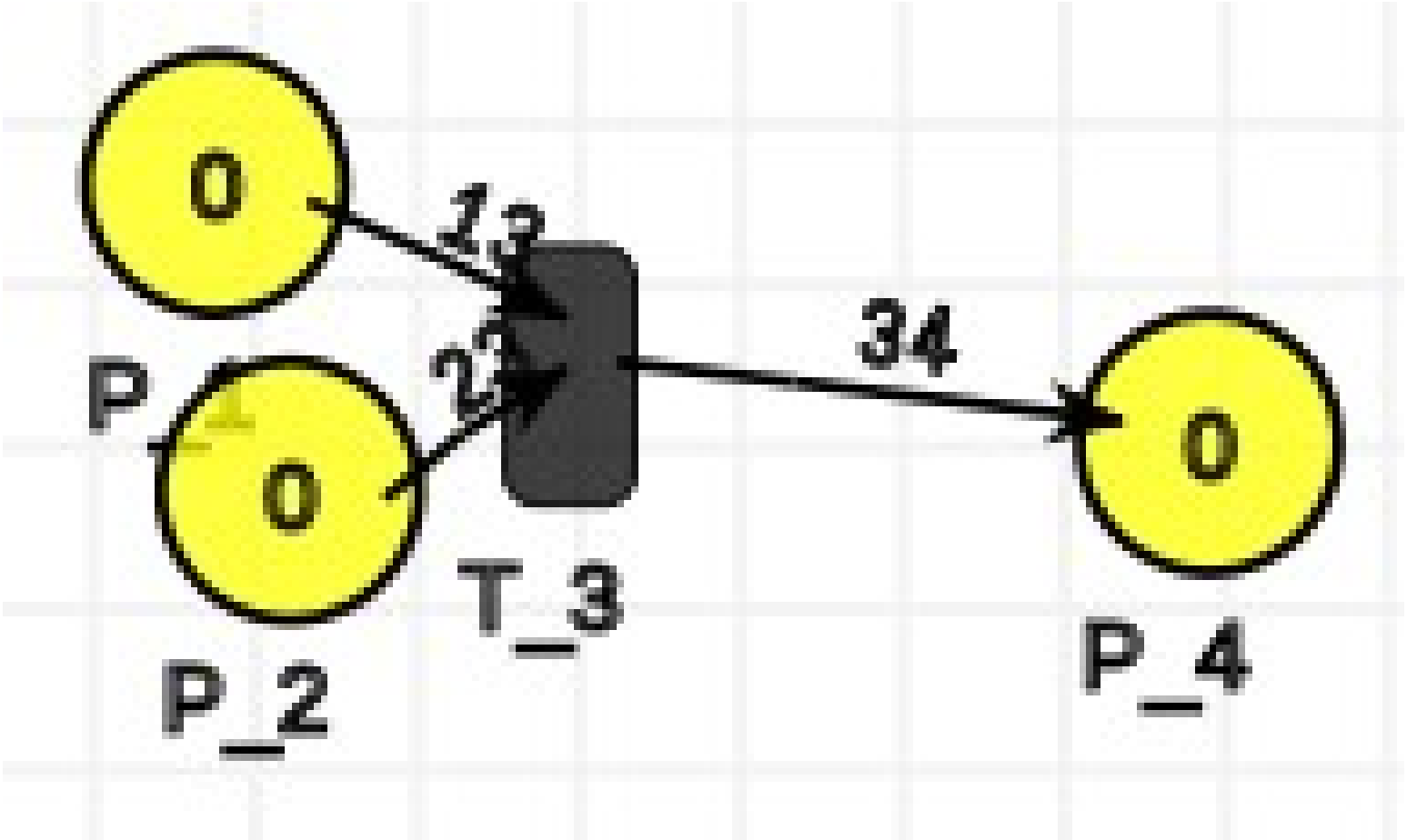
1. Impart on each vector a velocity towards the gravitational field source
2. Calculate force vector

- Increase area usage efficiently

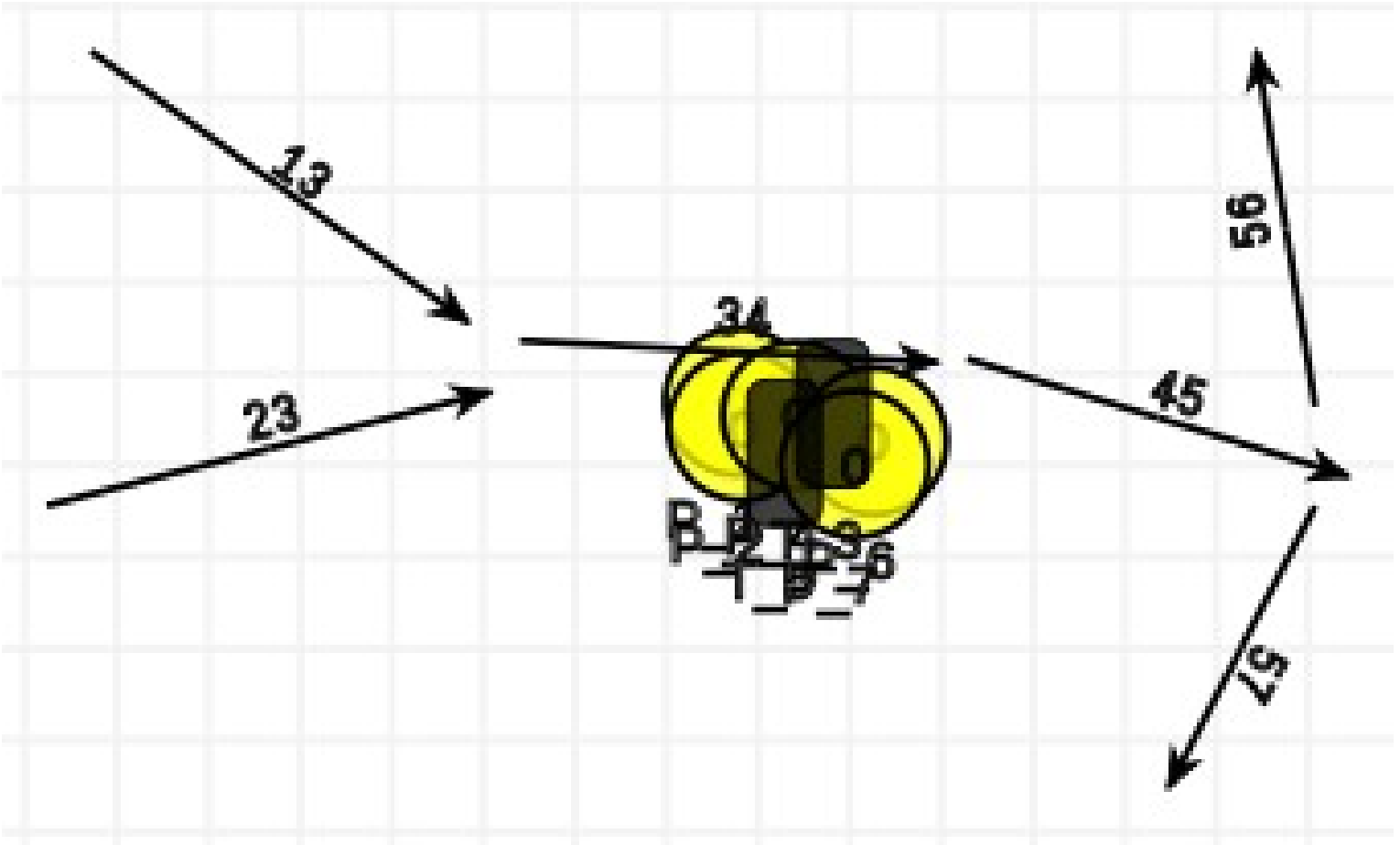
PetriNet before Spring-Embedder



PetriNet after Spring-Embedder



PetriNet after Spring-Embedder (2)

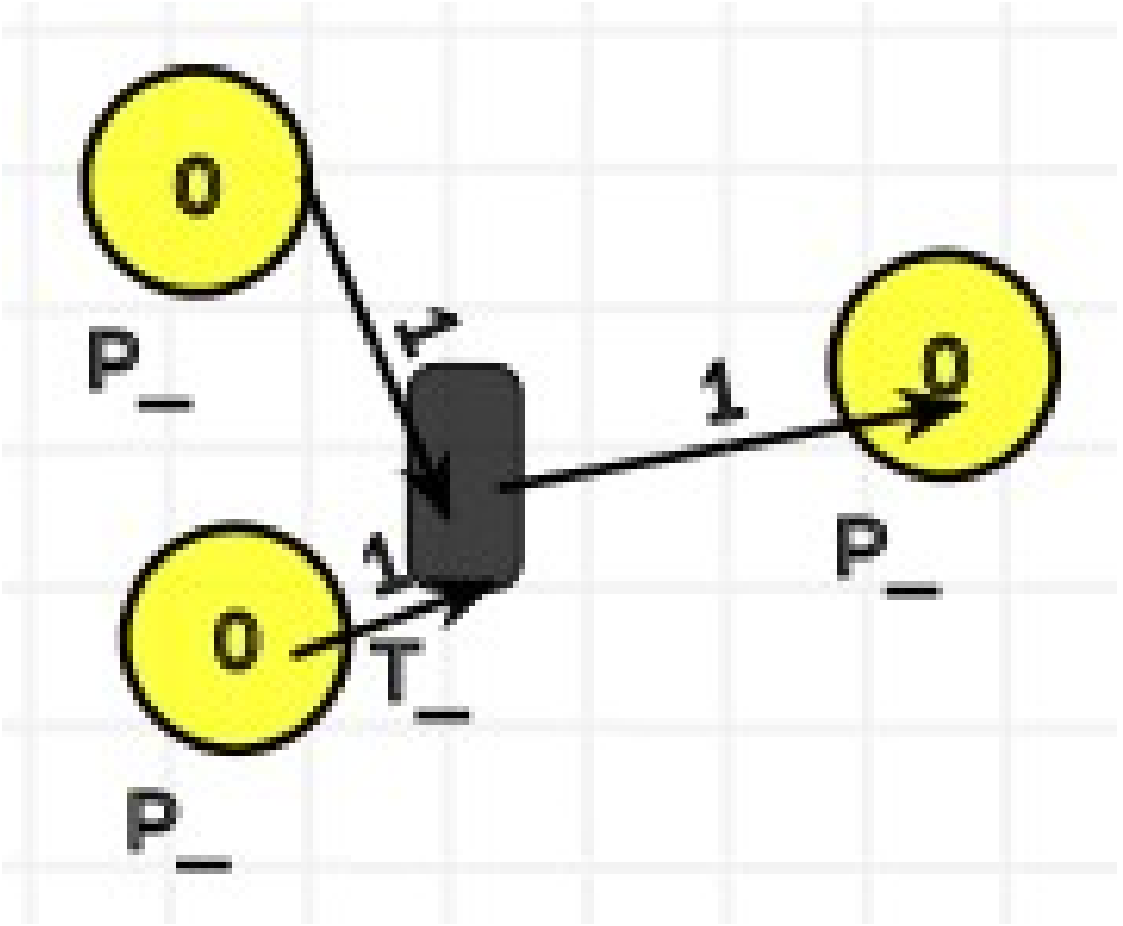


3. Force-transfer layout algorithm

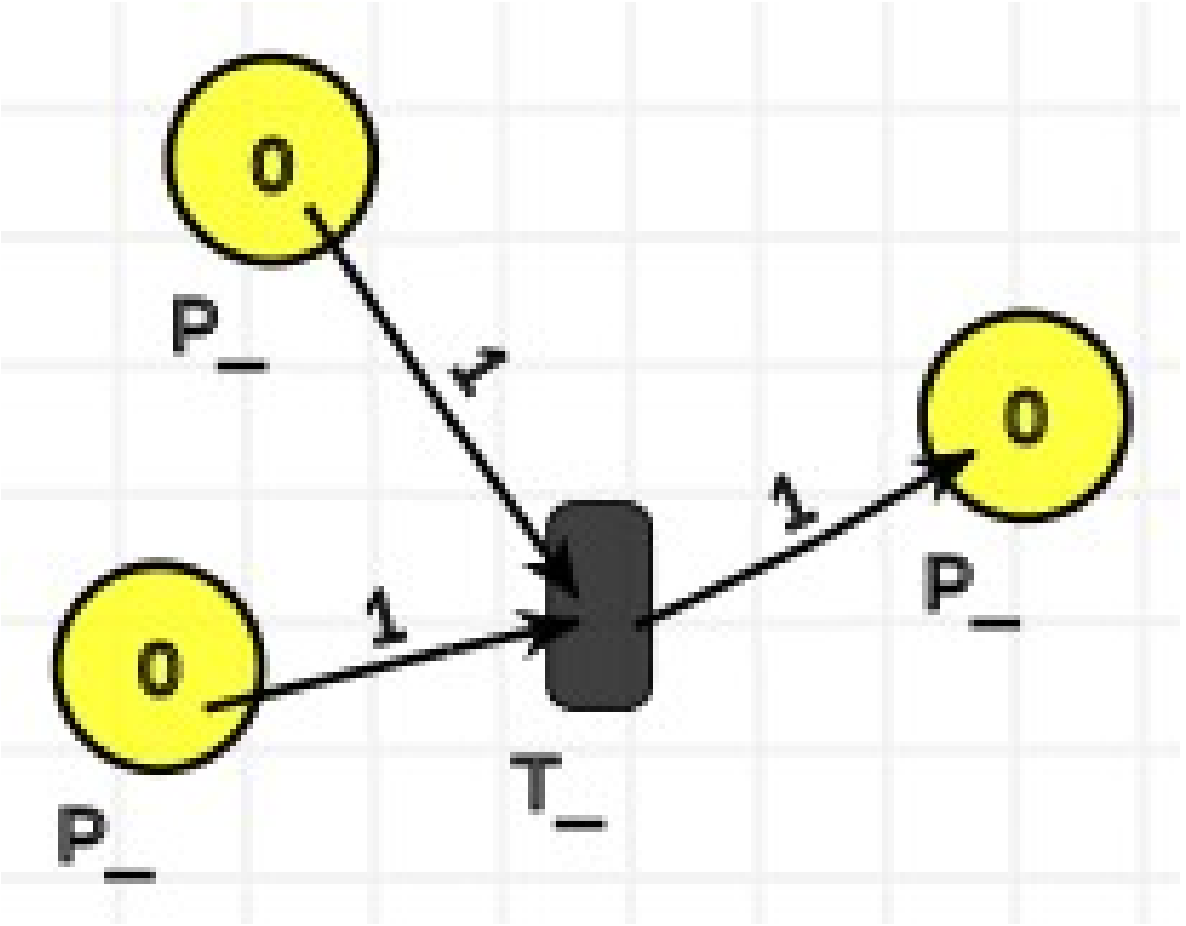
- Initialization phase
  - Set forces acting on each vertex to zero
  - Set position of vertex to its center coordinate
- Simulation phase
  - Each vertex exerts forces on overlapping neighboring vertices

    1. Calculate Manhattan and Euclidean distances
    2. Compute scalar force magnitude
- Termination
  - No more overlap
  - Fixed number of iterations

PetriNet before Force-transfer



PetriNet after Force-transfer

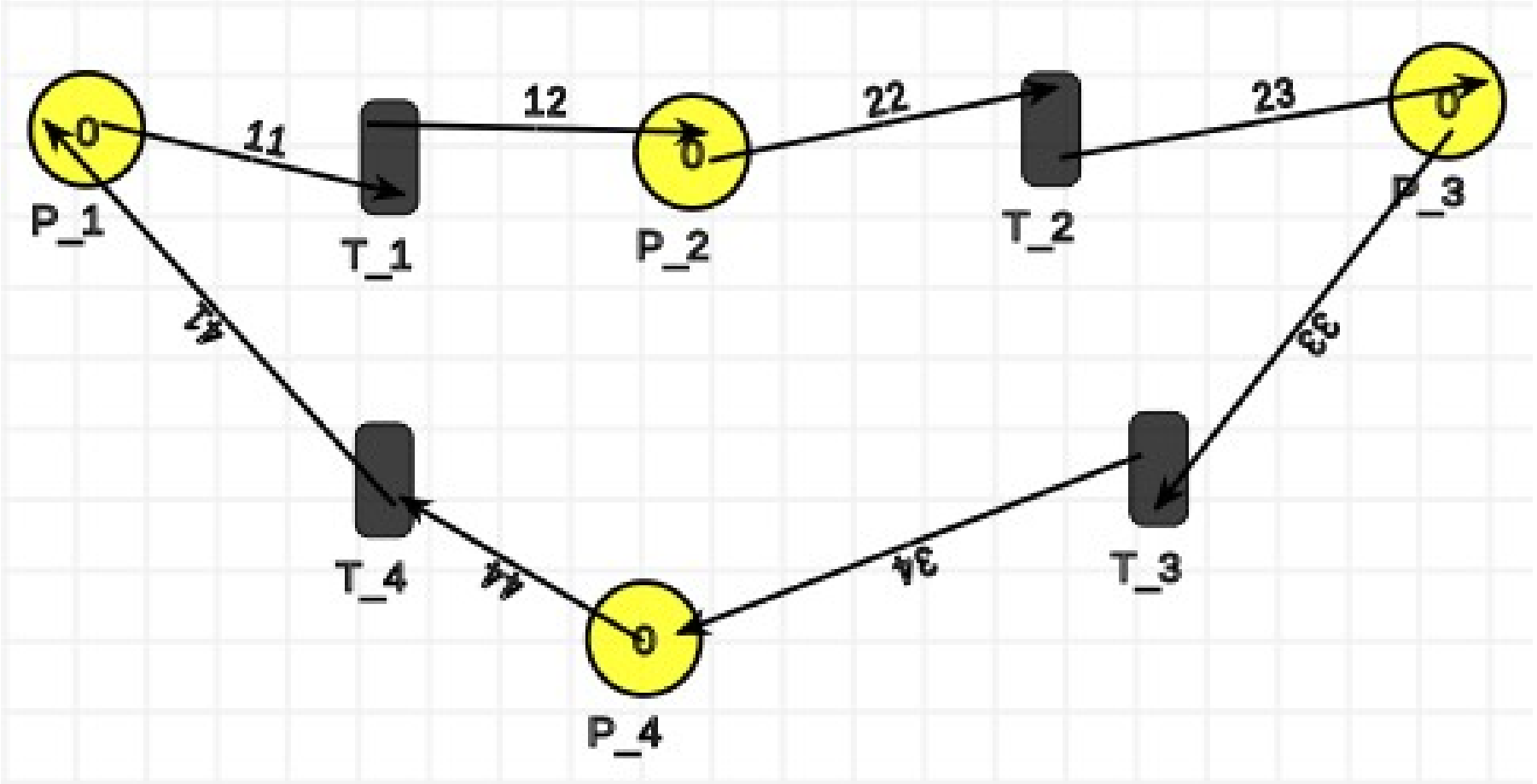


4. Circle layout algorithm

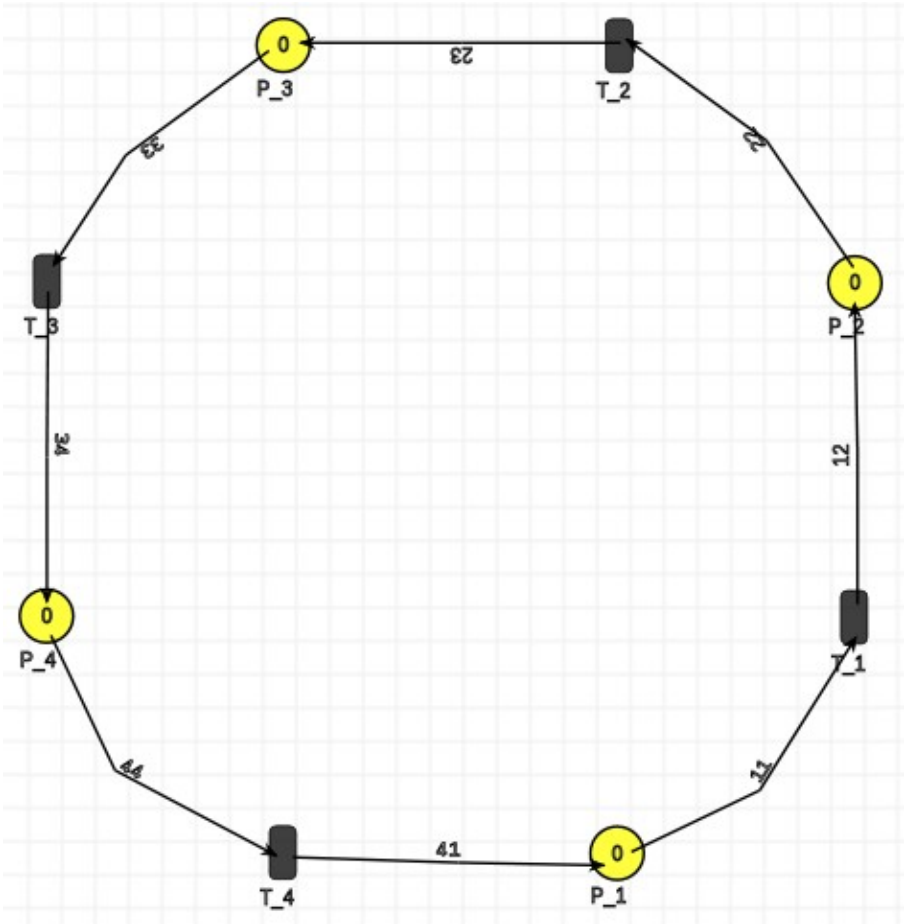
1. Sort vertices topologically
2. Calculate perimeter of circle
3. Calculate interval fraction

- Subgraphs or small graphs
- Preprocessing step for force directed method

PetriNet before Circle layout



PetriNet after Circle layout



5. Conclusion

- Small PetriNets: Circle layout
- Big PetriNets: Spring-Embedder
- Stop overlap: Force-Transfer
- Mostly combination of multiple algorithms

- Layout = fun!