GASP: A Computing Framework For Markov Chain Monte Carlo Spatial Problems

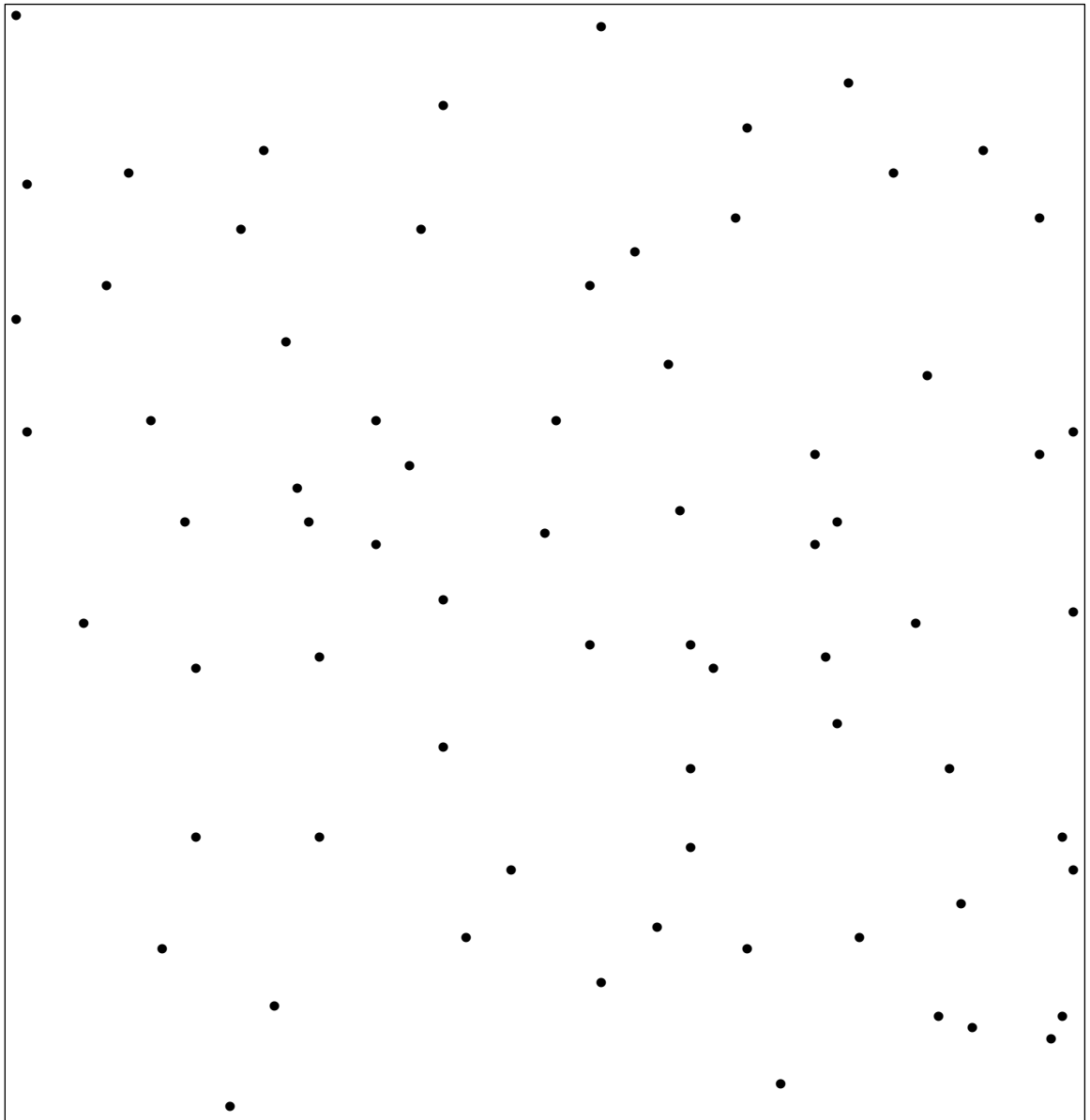
James Henstridge
james@daa.com.au

Overview

- An example with mathematical model
- Description of GASP
- Demonstration

Example

- positions of pine trees



Point Processes

- Point processes are processes of configurations of points in the plane
- Probability of a configuration defined by density function

Strauss Process

- The Strauss Process is a model for looking at point clustering patterns.

$$f(\mathbf{x}) = \alpha\beta^{\mathbf{n(x)}}\gamma^{\mathbf{s(x)}}$$

- Difficult to generate instances of the probability density function directly
- Need some other method to generate these configurations

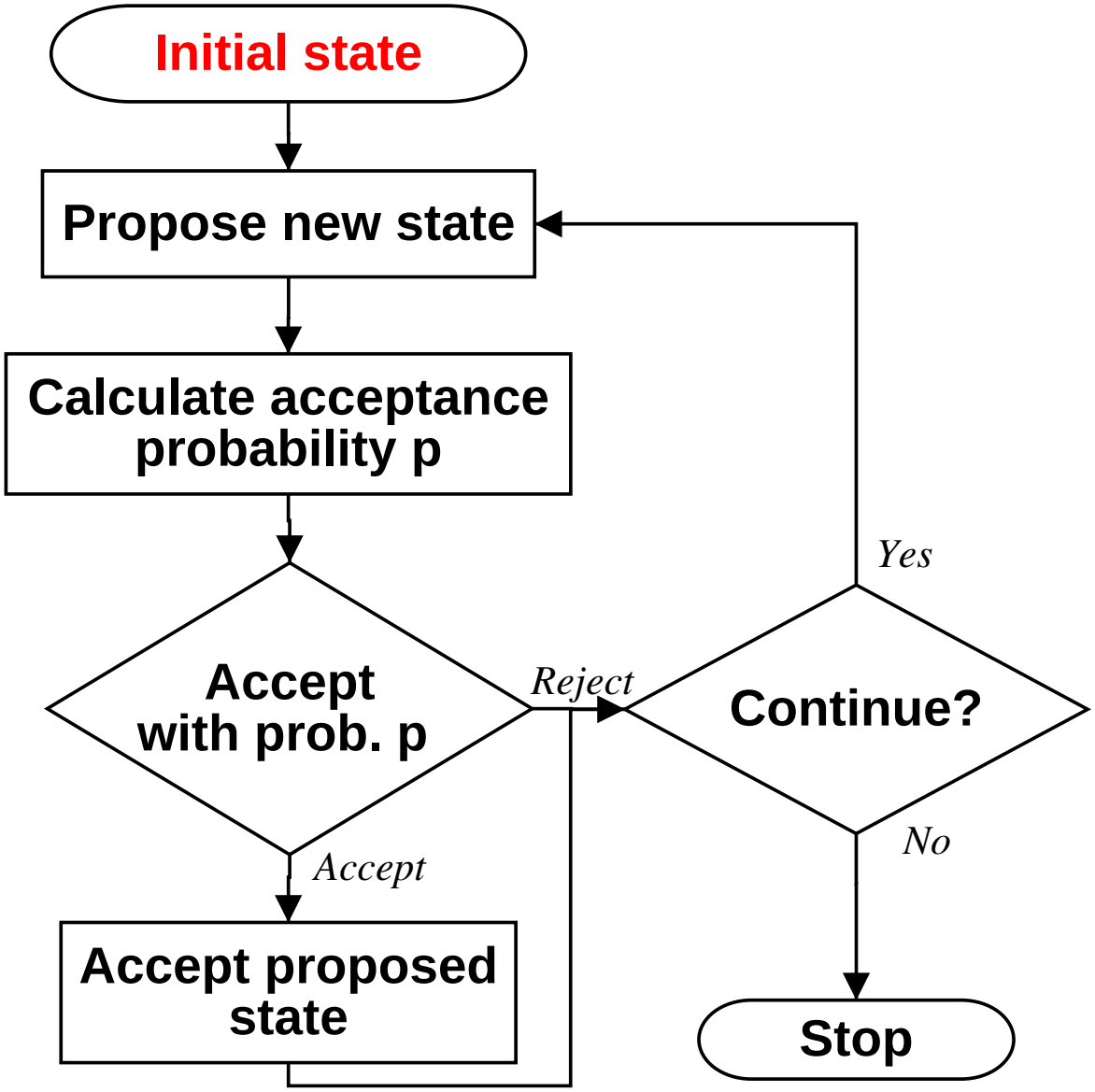
Simulation Point Processes

- Iterative algorithms are used to simulate the point processes.
- Compare producing a random ordering for a pack of cards vs. shuffling the pack.
- One such algorithm is the Metropolis-Hastings Algorithm
  - propose a change to the configuration
  - calculate probability of accepting the change
  - make a random decision whether to accept the change
  - if we accept, apply the change to the configuration.
  - repeat

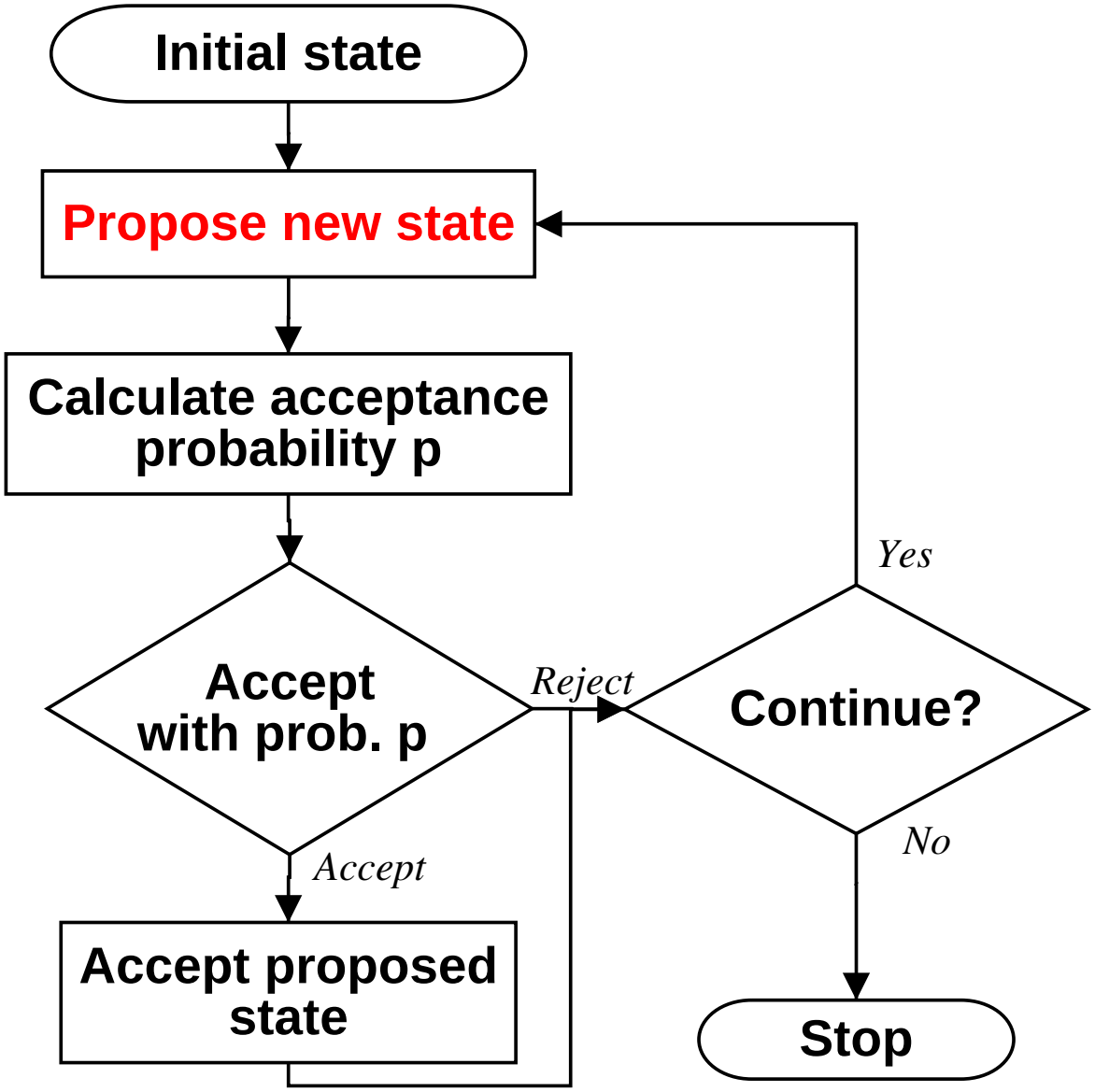
GASP

- I designed it as a framework to implement such simulations
  - (Generating Algorithms for Spatial Patterns)
- Can be used for algorithms similar to Metropolis-Hastings
- Written with the GAP package.
  - computer algebra package
  - freely available
  - interactive

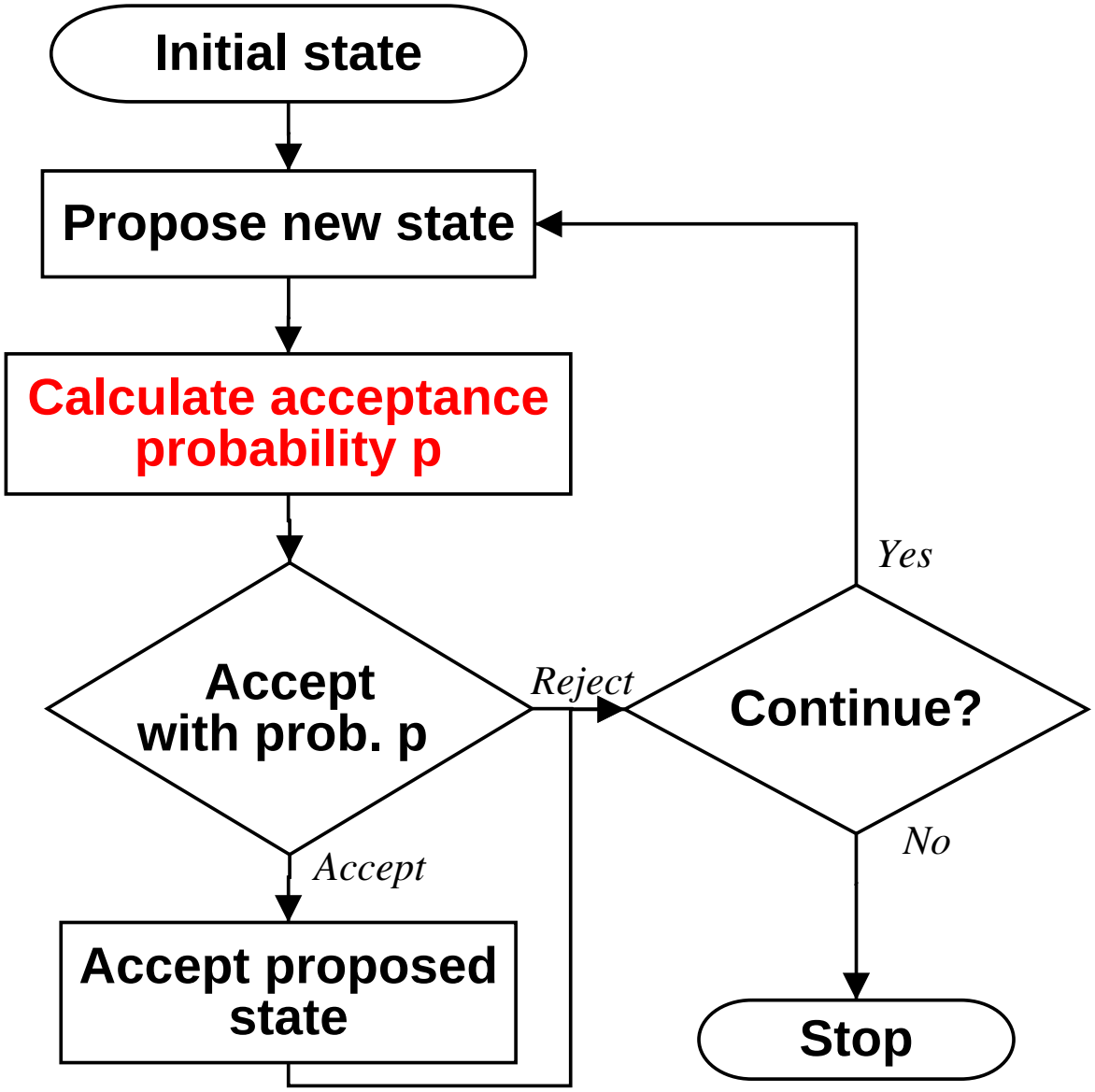
GASP Simulation Framework



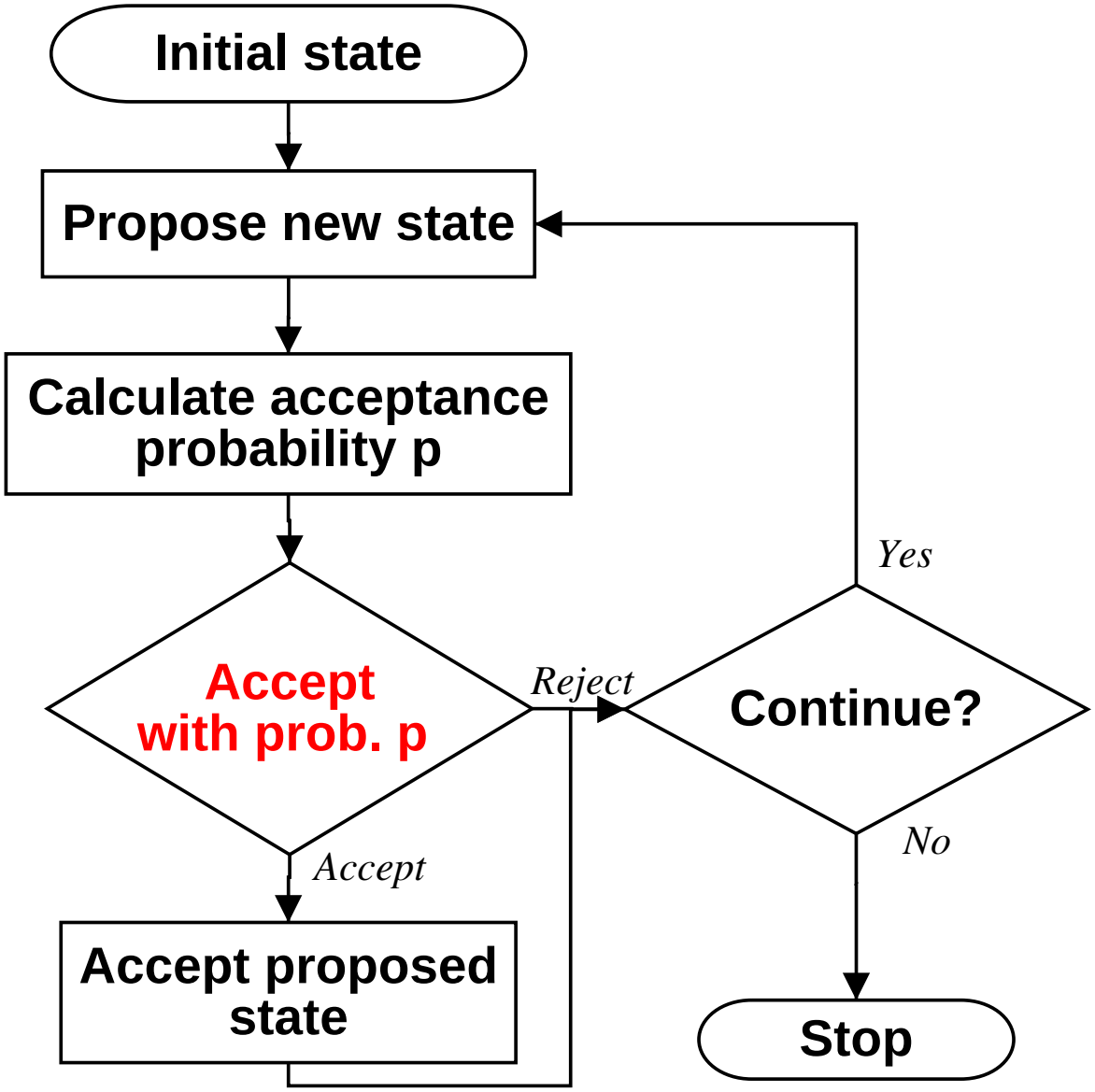
GASP Simulation Framework



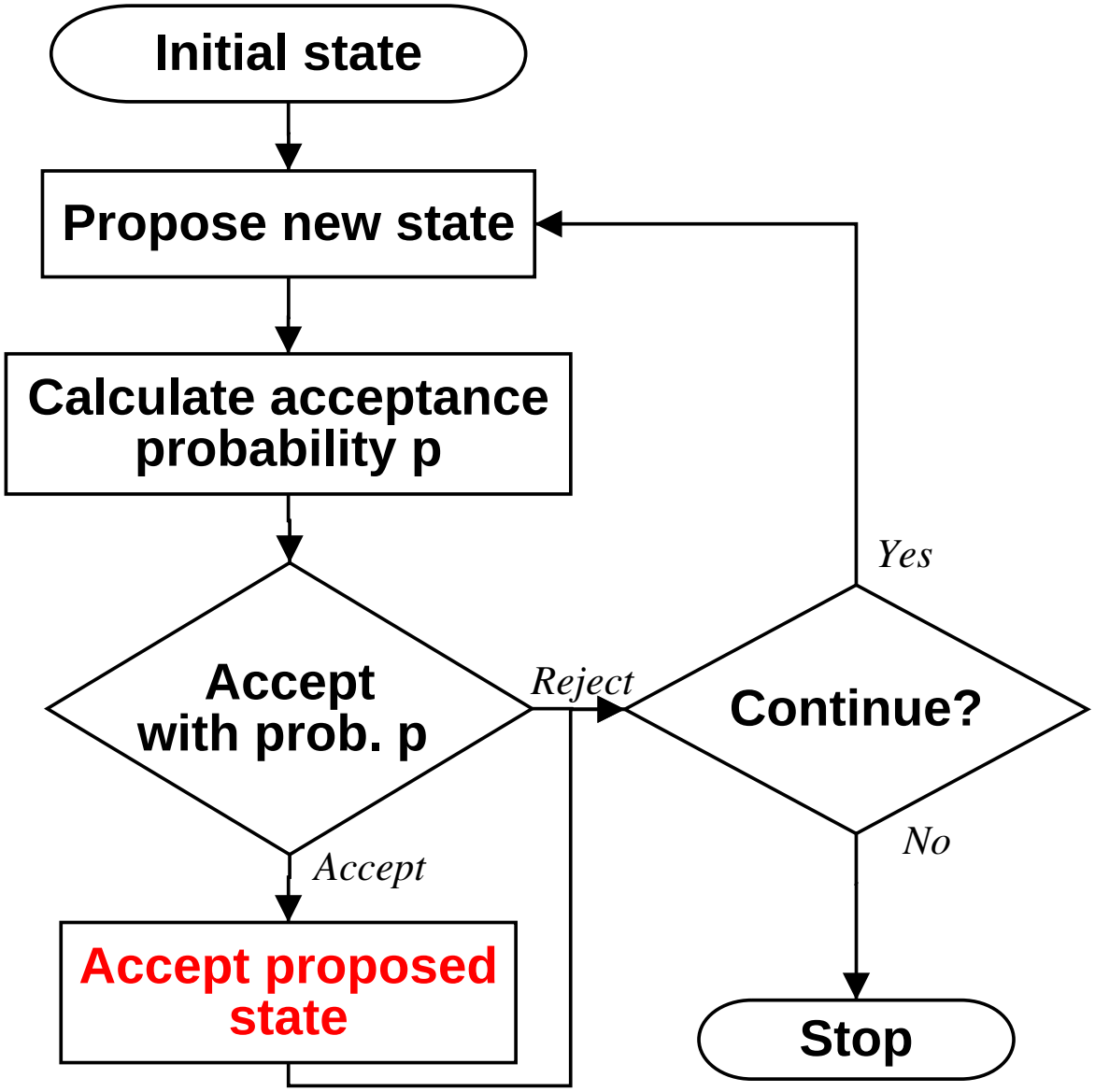
GASP Simulation Framework



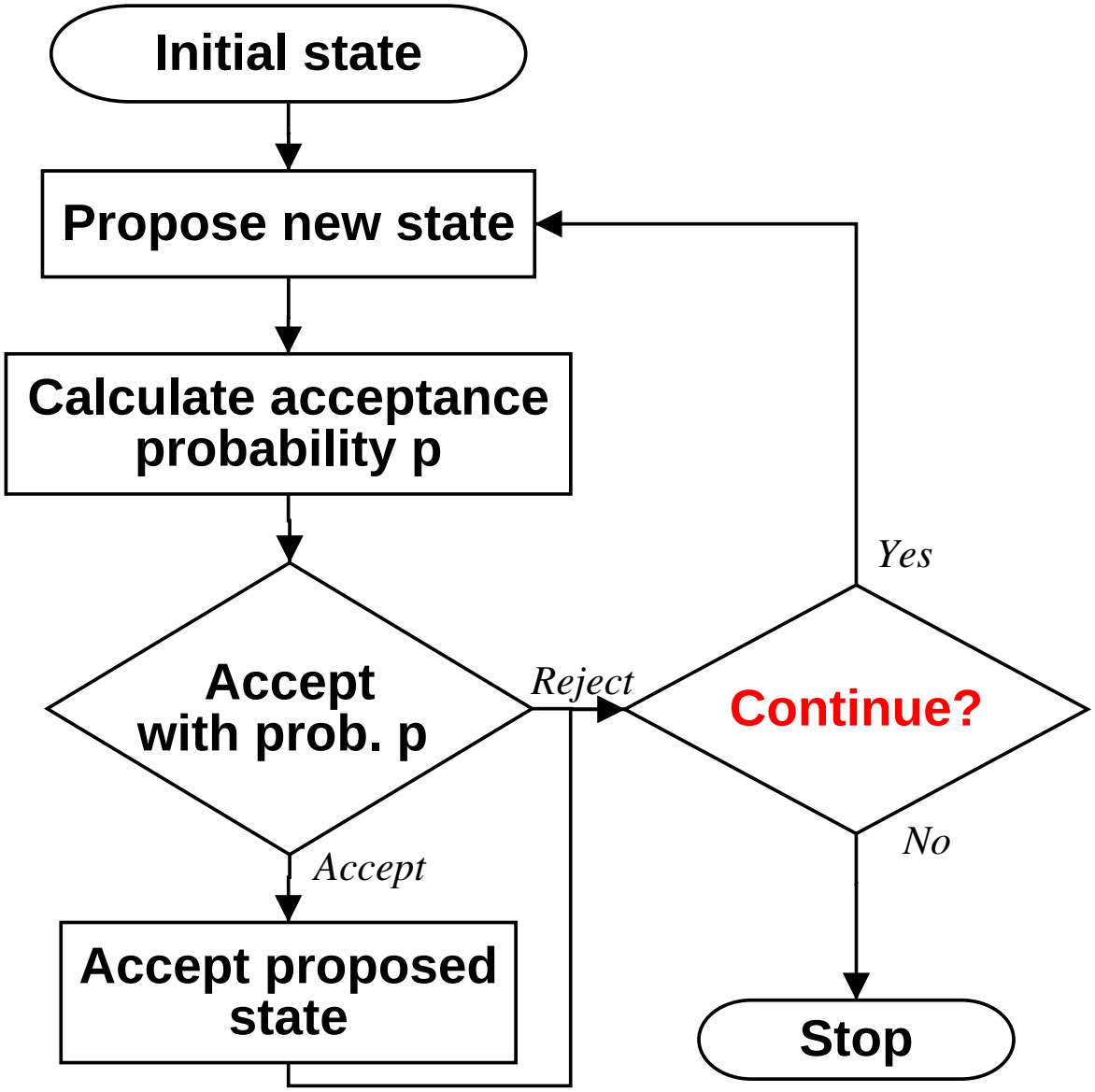
GASP Simulation Framework



GASP Simulation Framework



GASP Simulation Framework



The Change Log

- Most algorithms work with changes to the configuration
- Changes are interesting when analysing a simulation.
- sufficient to reproduce simulation
- Allows us to create more detailed logs

Applications of the Change Log

- Replay the simulation at high speed, pausing for certain occurences.
- Looking at runs of rejections.
- plot trajectories of:
  - number of points
  - a derived ‘score’ for the configuration

Example

- config := PointConfiguration(0,0,300,300);
- propose := CreateSimpleFlipPropose(1/2);
- check := CreateStraussCheck(1/900, 9/10, 15);
- GUISimulate(config, "Strauss", 300, 300, propose, check);

Example

- config := PointConfiguration(0,0,300,300);
- propose := CreateSimpleFlipPropose(1/2);
- check := CreateStraussCheck(1/900, 9/10, 15);
- GUISimulate(config, "Strauss", 300, 300, propose, check);

Example

- config := PointConfiguration(0,0,300,300);
- propose := CreateSimpleFlipPropose(1/2);
- check := CreateStraussCheck(1/900, 9/10, 15);
- GUISimulate(config, "Strauss", 300, 300, propose, check);

Example

- config := PointConfiguration(0,0,300,300);
- propose := CreateSimpleFlipPropose(1/2);
- check := CreateStraussCheck(1/900, 9/10, 15);
- GUISimulate(config, "Strauss", 300, 300, propose, check);

Conclusion

- Implement new algorithms
- Submit as official share package for GAP
- Extend functionality