

An aerial, high-angle view of a multi-lane highway interchange. The image is heavily blurred to create a sense of motion, with light trails from cars appearing as streaks of red, orange, and yellow. A bright sun or light source is visible in the upper center, casting a strong glow over the scene. The overall color palette is dominated by blues, greys, and the warm tones of the light trails.

# Statistical Traffic Model

Mathematical World

# CASE STUDY



# WHAT IS A MODEL?

A mathematical model is a description of a system using mathematical concepts and language.



In our case we use a simple model to try to describe traffic flow.

# STOCHASTIC PROCESS

A stochastic process is a collection of random variables that take values in a set  $S$ , the state space. The state space can be

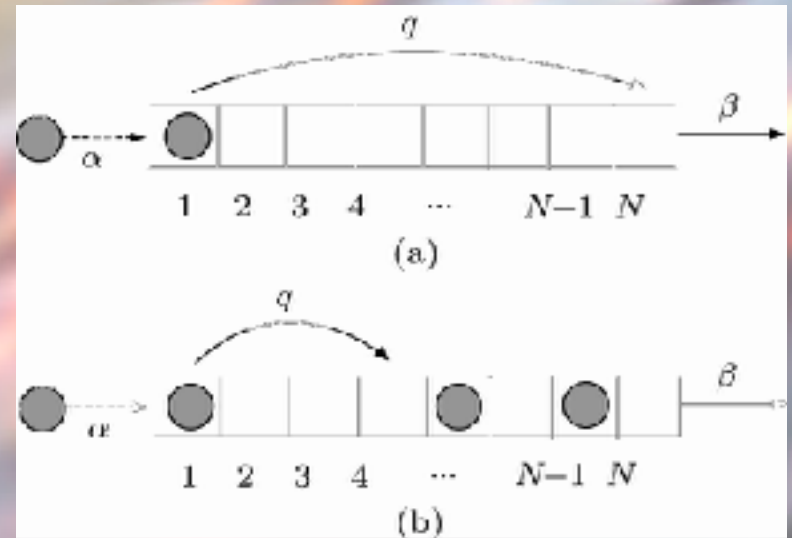
finite, countably infinite, or uncountable, depending on the application. In order to be able to analyze a stochastic process, we need to make assumptions on the dependence between the random variables.





# TASEP

The Totally Asymmetric Simple Exclusion Process is a stochastic model studied in non-equilibrium statistical physics. Important practical applications of TASEP spans between modelling intracellular transport, traffic flows both in networks and vehicular, productive processes ecc.



# PARALLEL TASEP

At each timestep every car moves forward if the following site is empty. Being a pure **deterministic** process it's very easy to show that there are only two stationary configurations:  
either all cars on even sites or on the odd ones.



WHAT HAPPENS IF  $P$  IS  $<1$ ?

In this case the model behaves more realistically but also more difficult to study since there is no particular symmetry.

# SERIAL ALEATORY TASEP

## **New dynamics:**

only one car, chosen randomly, moves forward if the following site is empty.

In this model all possible configurations are eventually reached

The number of possible configurations is the same as before:

$$\binom{2n}{n}$$



# SERIAL ALEATORY TASEP

The transition matrix, which represents the probability to pass from a configuration  $s$  to  $t$ , has a particular property: it's doubly Markov.

This implies that each configuration has the same probability to appear: the stationary measure is uniform.

$$P(\text{car in site}) = \frac{1}{2} \quad P(\text{site empty}) = \frac{1}{2}$$



Hence, the average current is the product of the probability for a site to be occupied and for the next one to be empty:  $\frac{1}{4}$

# CURRENTS COMPARISON

Parallel deterministic  
TASEP:

$$J=1/2$$

This model, presenting just two stages, has the maximum possible average current

Parallel aleatory  
TASEP:

$$1/4 < J < 1/2$$

The value of the current in this model floats between max and min currents depending on the dynamics

Serial aleatory  
TASEP:

$$J=1/4$$

In this model the average current always tends to the minimum

# BLOCKAGE

The blockage is a local decrement of speed caused by some sort of bottleneck.

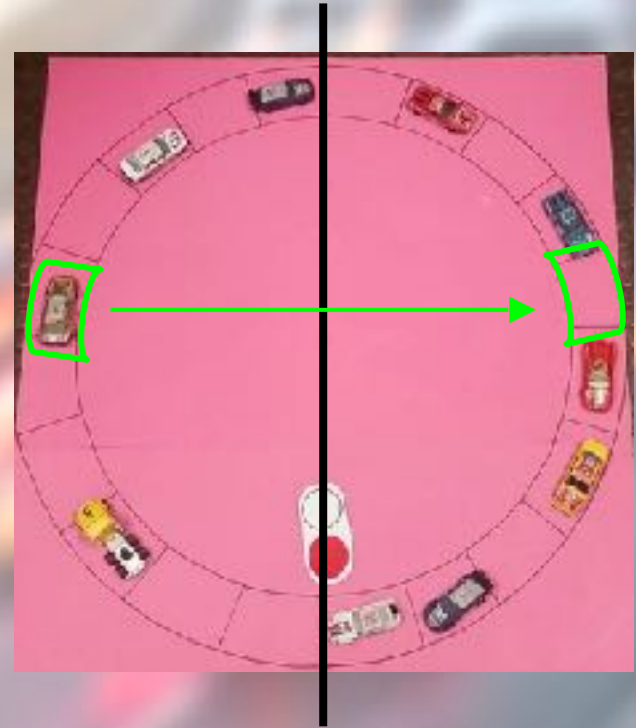
For example a semaphore placed between sites 0 and 1 having probability to being red( $\epsilon$ ).



# BLOCKAGE

As we see, if we introduce a local blockage in the  $P=1$  TASEP, once stationary conditions are reached we can see a PARTICLE-HOLE symmetry.

Symmetries are very powerful tool in physical sciences

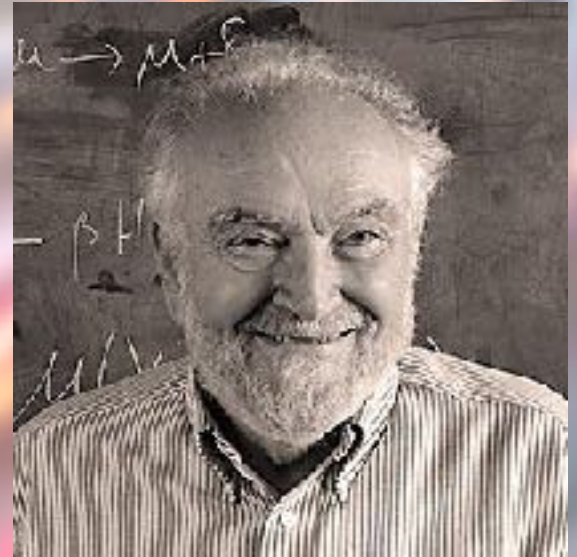




# BLOCKAGE

More difficult is to define what happens in a serial tasep when we introduce blockage.

So we used computer simulation of our model led by RUBY script, to analyse data. What we got is the result already analysed by Joel Lebowitz in his conjecture



# BLOCKAGE

```
cycles.times do |c|  
  
  #calcolo la percentuale  
  if (c % [cycles / 100] == 0)  
    print " " * 10  
  
    if(percent % 10 == 0)  
      | print "(#{percent}%)\\n"  
    end  
  
    percent += 1  
  end  
  
  #estraggo un singolo sito nello stato  
  i = random.rand(s.length):  
  
  #incontrare la cella adiacente alla sinistra  
  if i == (s.length - 1)  
    j = 0  
  else  
    j = i - 1  
  end  
  
  #applico la transizione di momento del modello  
  if s[i] == 1 and s[j] == 0  
    if not(i == 0 and random.rand < red)  
      | s[i] = 0  
      | s[j] = 1  
    end  
  end  
  
end
```

