A comparison of methods to evaluate the probability of excessive waiting in the *M(t)/G/s(t)+G* queue

Stefan Creemers (IESEG & KU Leuven)

Mieke Defraeye (KU Leuven)

Inneke Van Nieuwenhuyse (KU Leuven)

IÉSEG SCHOOL OF MANAGEMENT

KATHOLIEKE UNIVERSITEIT LEUVEN

Problem Setting

- Service systems with:
  - Time-varying demand for service/supply of service
  - Abandonments
  - Exhaustive service discipline
  - General service & abandonment distributions

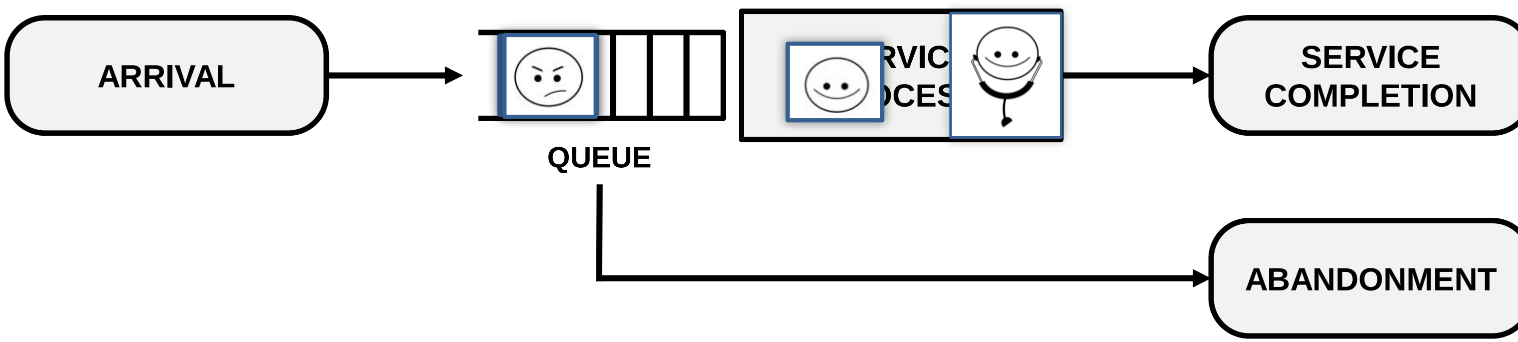
Problem Setting

- Service systems with:
  - Time-varying demand for service/supply of service
  - Abandonments
  - Exhaustive service discipline
  - General service & abandonment distributions
- $M(t)/G/s(t)+G$ queue

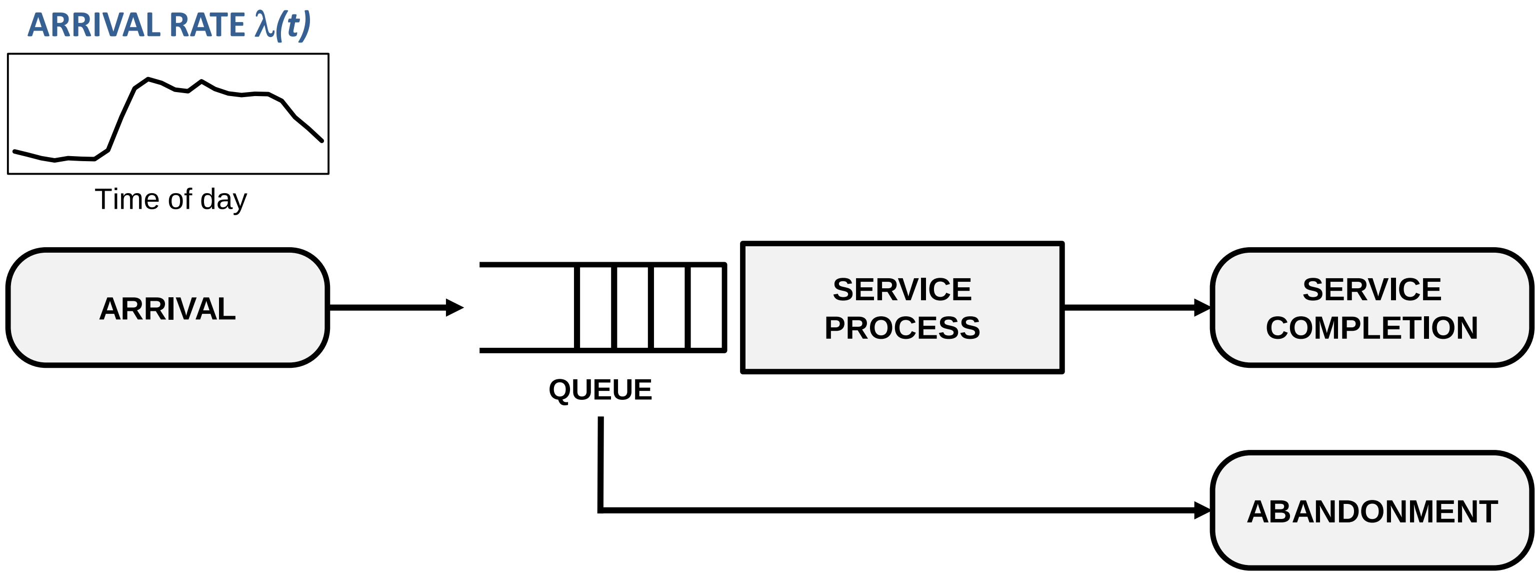
Problem Setting

- Service systems with:
  - Time-varying demand for service/supply of service
  - Abandonments
  - Exhaustive service discipline
  - General service & abandonment distributions
- $M(t)/G/s(t)+G$ queue
- Examples: Emergency departpments, call centers, fastfood restaurants, supermarkets, retail stores, banks…

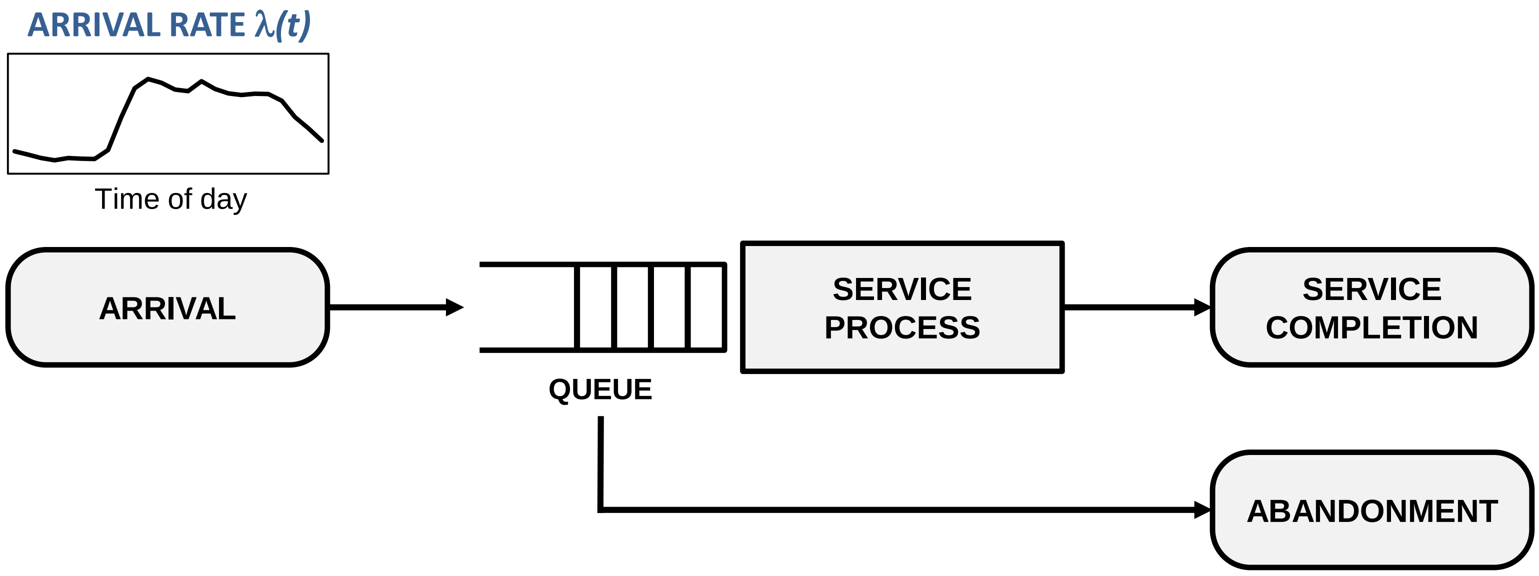
Problem Setting



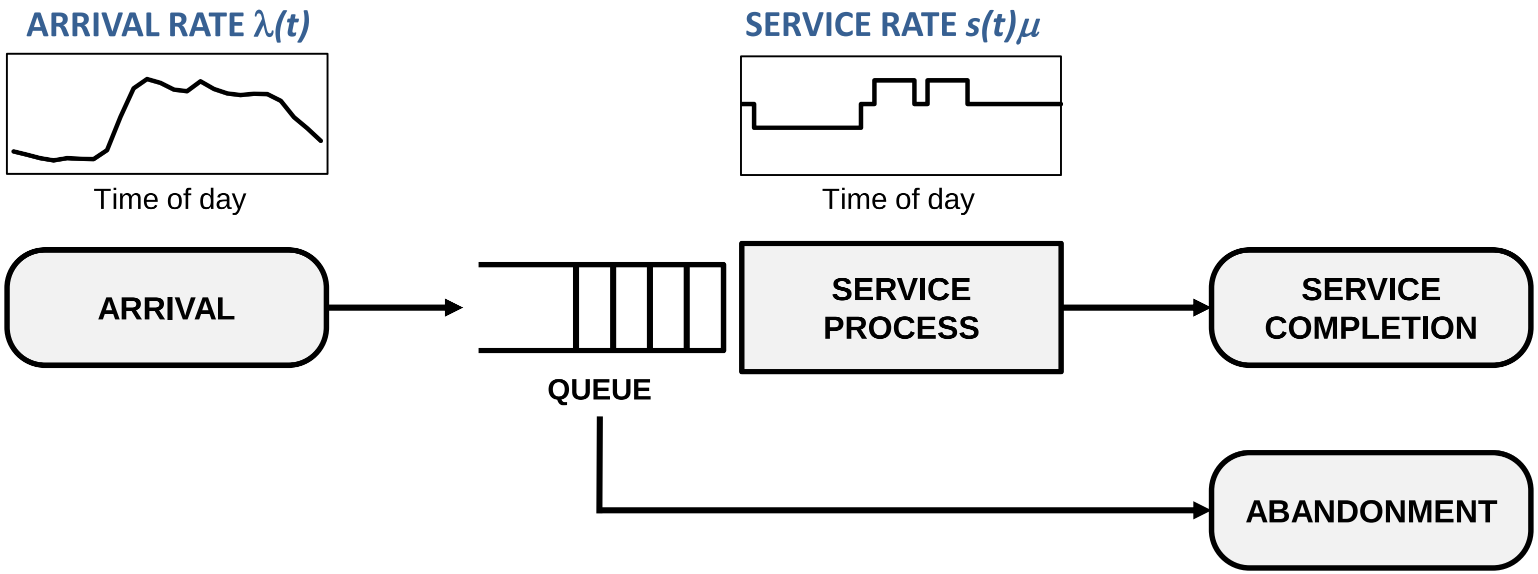
Problem Setting



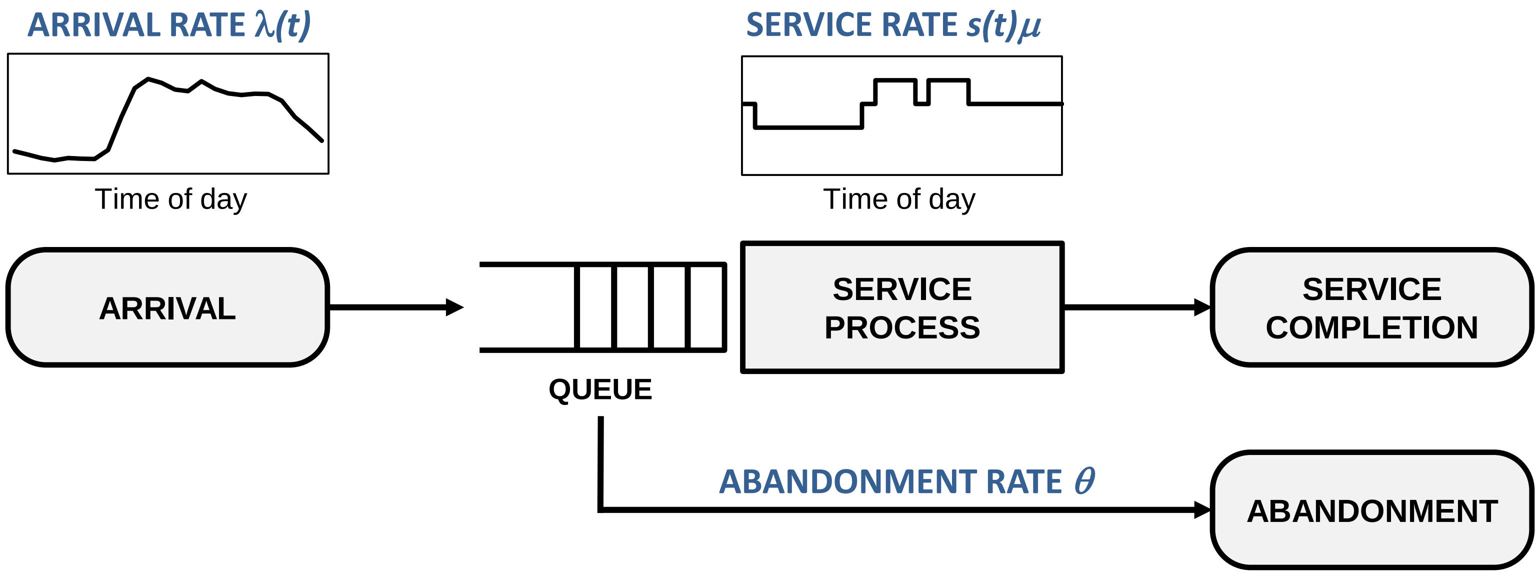
Problem Setting



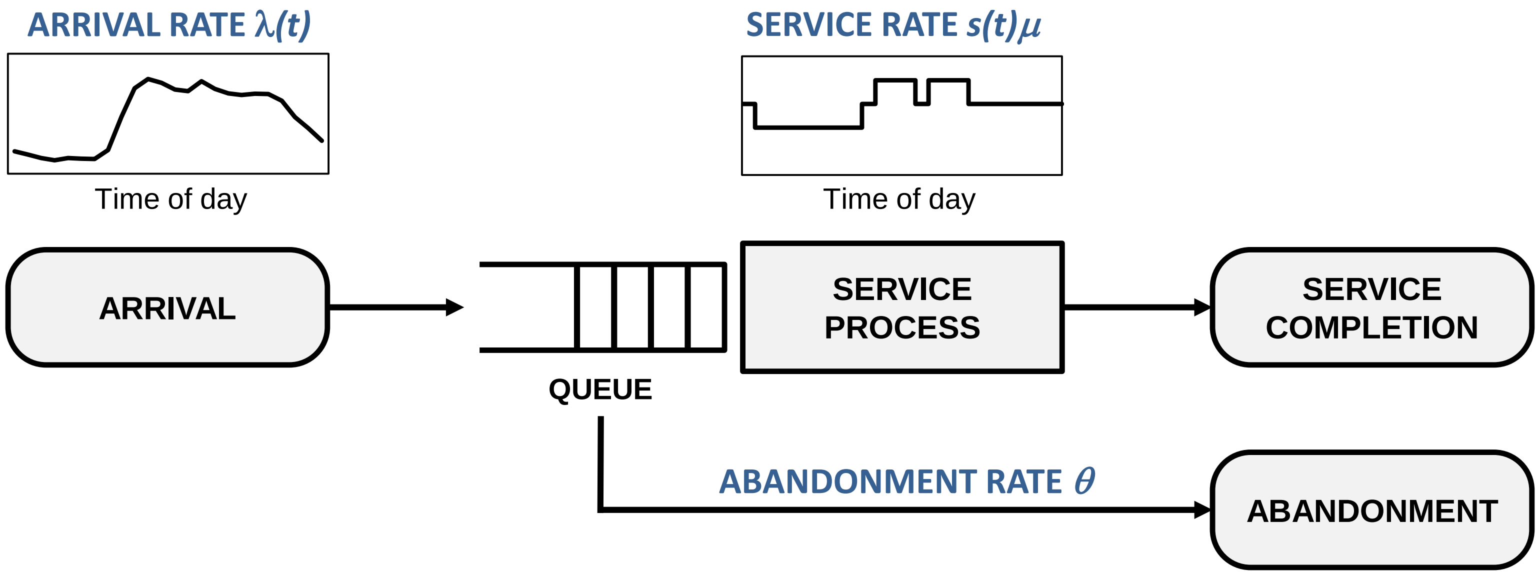
Problem Setting



Problem Setting



Problem Setting



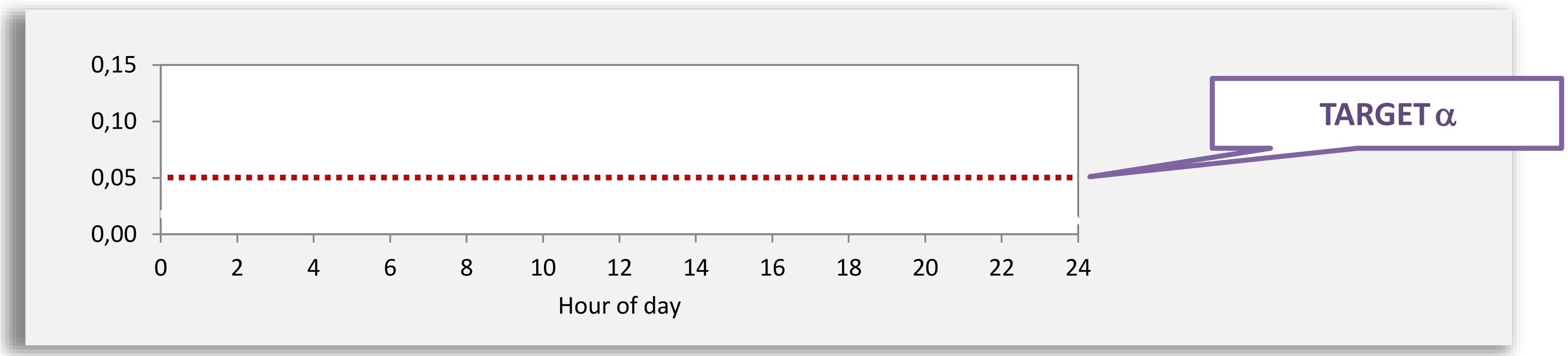
$$M(t)/G/s(t)+G$$

Research Question

How can we measure the probability of excessive waiting, given this time-varying demand for service/supply of service?

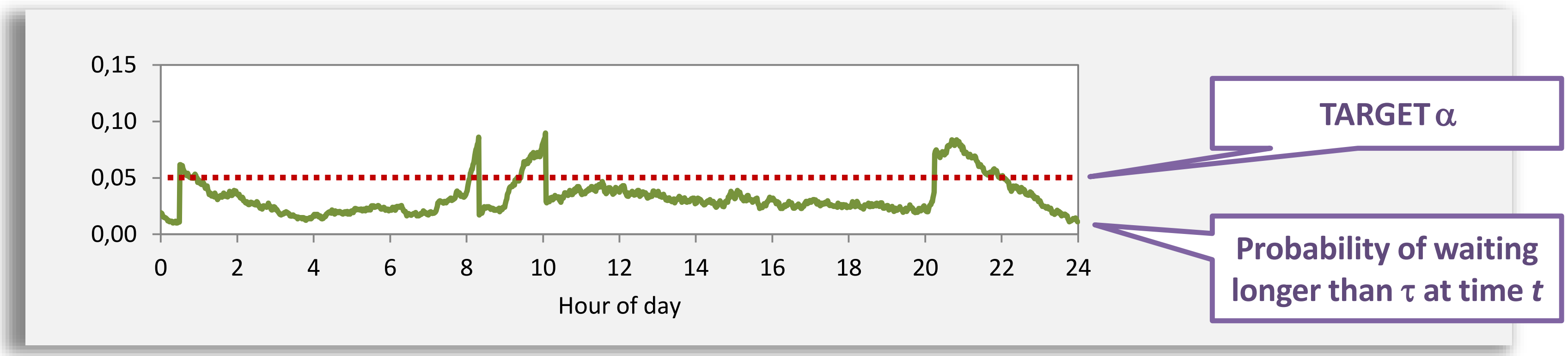
Research Question

How can we measure the probability of excessive waiting, given this time-varying demand for service/supply of service?



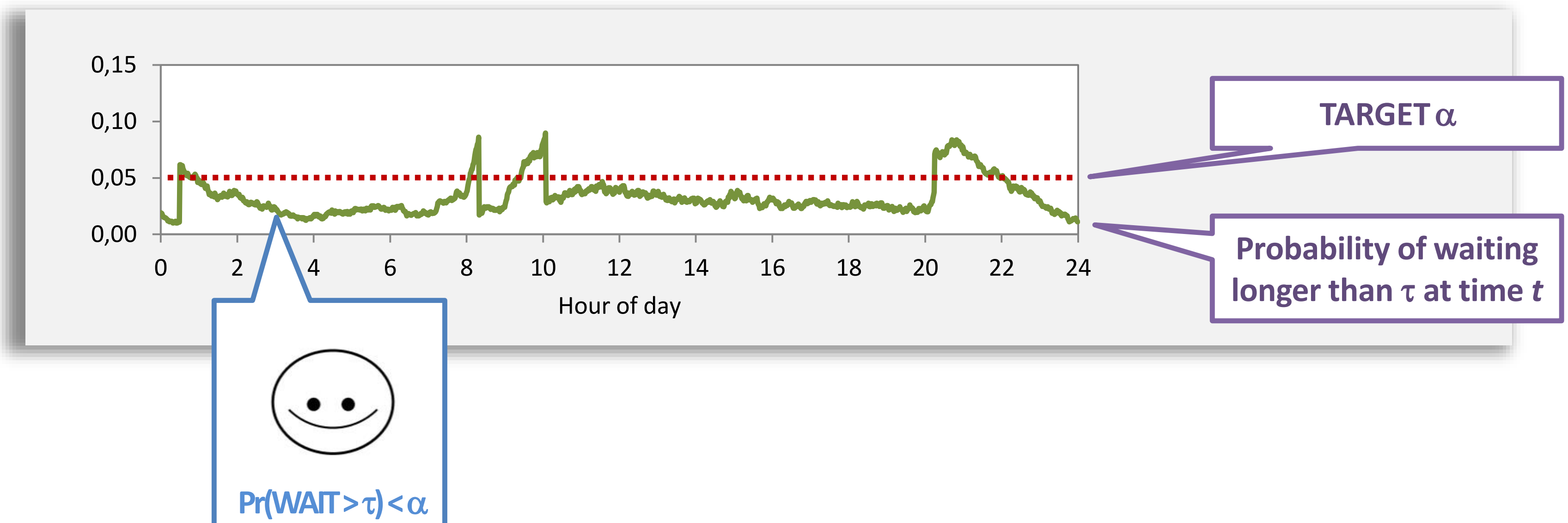
Research Question

How can we measure the probability of excessive waiting, given this time-varying demand for service/supply of service?



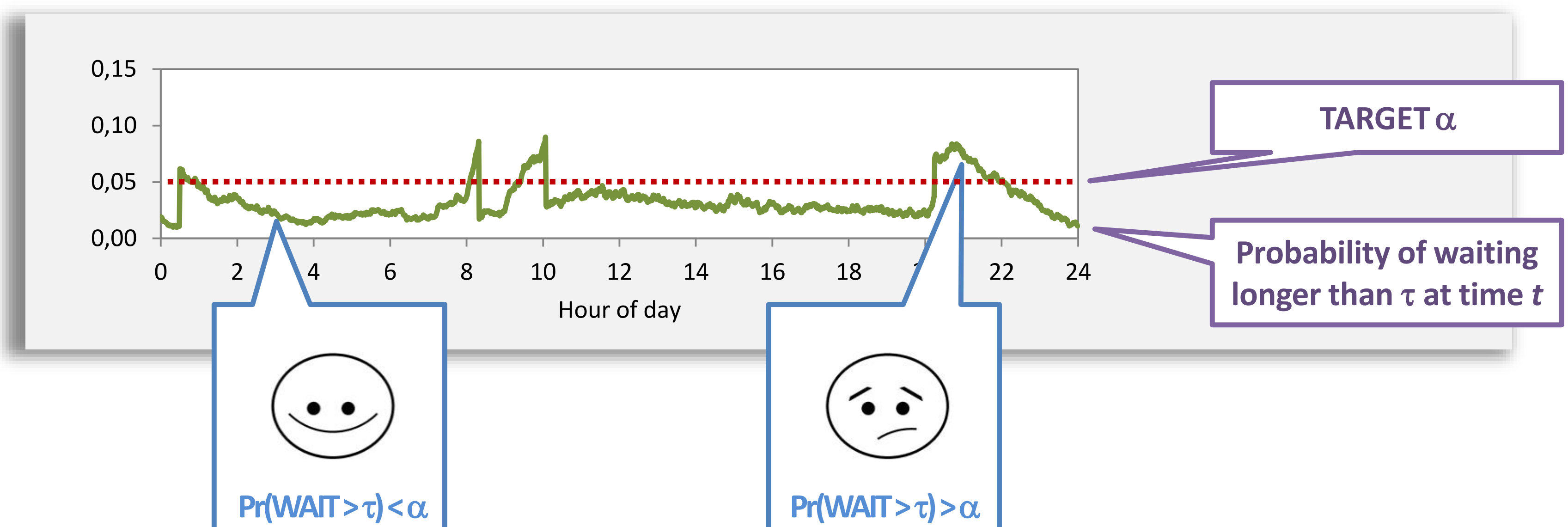
Research Question

How can we measure the probability of excessive waiting, given this time-varying demand for service/supply of service?



Research Question

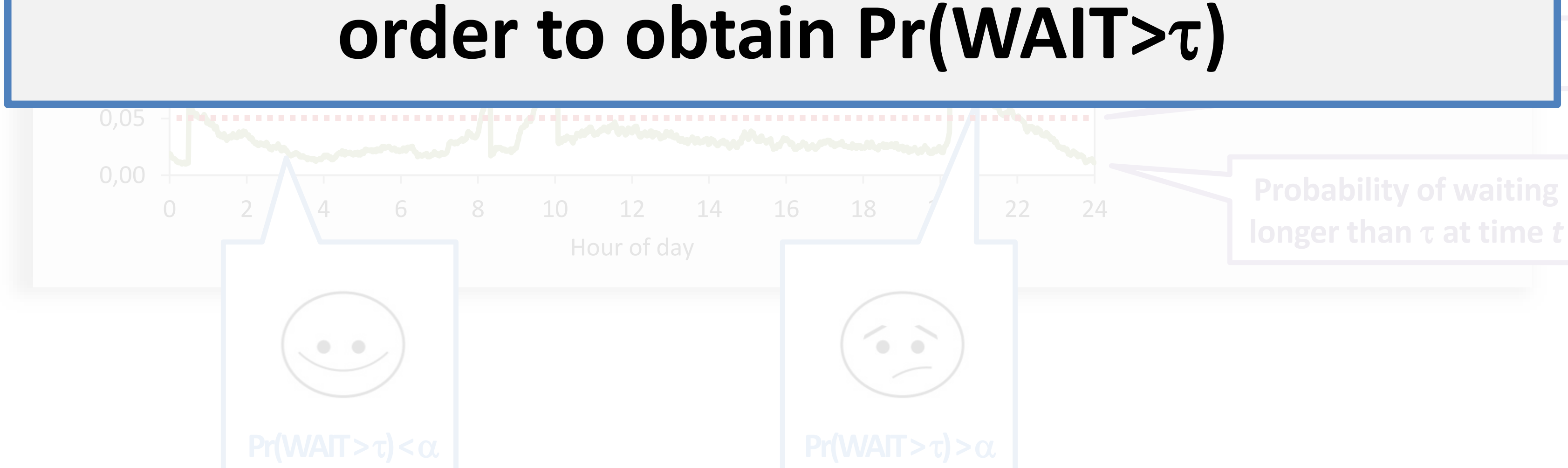
How can we measure the probability of excessive waiting, given this time-varying demand for service/supply of service?



Research Question

How can we measure the probability of excessive waiting, given this time-varying demand for service/supply of service?

For every instant in time *t*, we need to compute the waiting time distribution in order to obtain Pr(WAIT>τ)

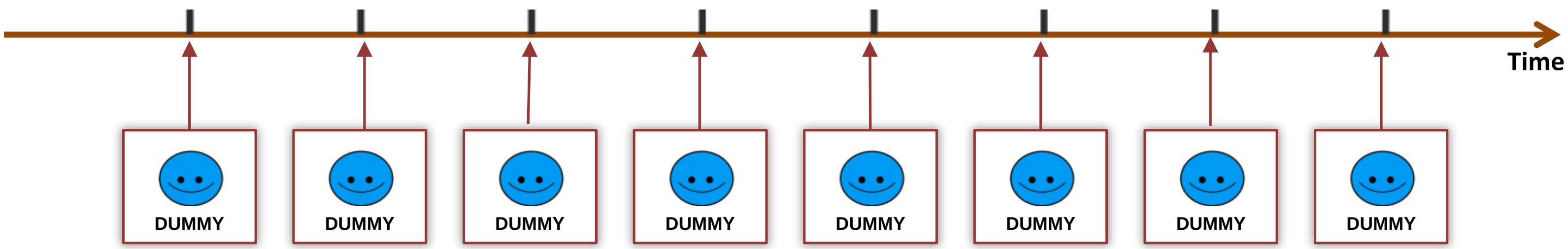


Methodology

- Comparison of three methods that allow to assess the probability of excessive waiting:
  - Simulation
  - MOL (Modified Offered Load)
  - G-RAND (General Randomization)
- We compare these methods based on accuracy and CPU-time

Methodology
Simulation

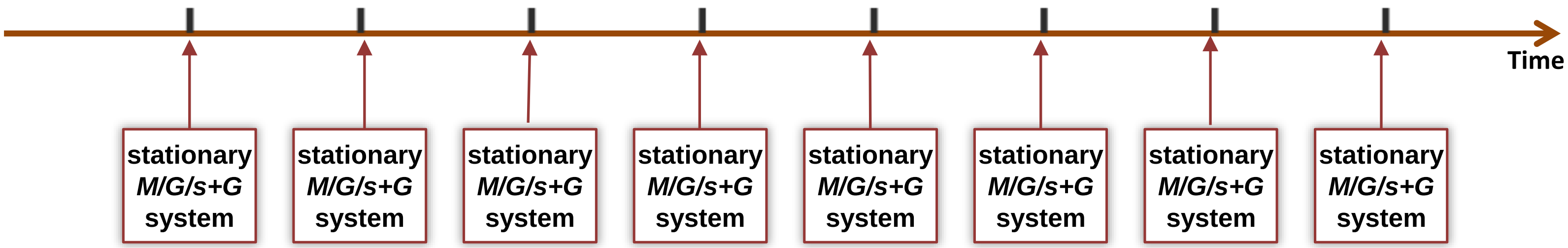
- Virtual waiting times (i.e., we insert a dummy customer in the model and observe how long it would take before he/she would receive service).



- Number of simulation replications: 250, 500, 1000, 2000, 4000, 8000

Methodology
MOL (Modified Offered Load)

- At every instant in time time *t* we solve a stationary *M/G/s+G* system using a modified arrival rate $\lambda_{MOL}(t)$



- See for instance Jagerman (1975), Jennings et al. (1996), etc.

Methodology
MOL (Modified Offered Load)

- The modified arrival rate is computed as follows:

$$\lambda_{\mathrm{MOL}}(t) \equiv m_\infty(t)\mu\,,$$
$$\text{where } m_\infty(t) = \int_{-\infty}^{t} [1 - G(t-u)]\lambda(u)\,\mathrm{d}u\,.$$

Methodology

MOL (Modified Offered Load)

- The modified arrival rate is computed as follows:

$$\lambda_{\mathrm{MOL}}(t) \equiv m_\infty(t)\mu\,,$$

$$\text{where } m_\infty(t) = \int_{-\infty}^{t} [1 - G(t-u)]\lambda(u)\,\mathrm{d}u\,.$$

Modified Offered Load

Methodology
MOL (Modified Offered Load)

- The modified arrival rate is computed as follows:

$$\lambda_{\text{MOL}}(t) \equiv m_\infty(t)\mu\,,$$

$$\text{where } m_\infty(t) = \int_{-\infty}^{t} [1 - G(t-u)]\lambda(u)\,\mathrm{d}u\,.$$

Individual service rate

Modified Offered Load

Methodology
MOL (Modified Offered Load)

- The modified arrival rate is computed as follows:

$$\lambda_{\text{MOL}}(t) \equiv m_\infty(t)\mu\,,$$

Individual service rate

$$\text{where } m_\infty(t) = \int_{-\infty}^{t} [1 - G(t-u)]\lambda(u)\,\mathrm{d}u\,.$$

Modified Offered Load

Service CDF

Methodology

MOL (Modified Offered Load)

- The modified arrival rate is computed as follows:

$$\lambda_{\mathrm{MOL}}(t) \equiv m_\infty(t)\mu\,,$$

$$\text{where } m_\infty(t) = \int_{-\infty}^{t} [1 - G(t-u)]\lambda(u)\,\mathrm{d}u\,.$$

Individual service rate

Modified Offered Load

Service CDF

Original arrival rate

Methodology
MOL (Modified Offered Load)

- The modified arrival rate is computed as follows:

$$\lambda_{\mathrm{MOL}}(t) \equiv m_{\infty}(t)\mu\,,$$

Individual service rate

$$\text{where } m_{\infty}(t) = \int_{-\infty}^{t} [1 - G(t-u)]\lambda(u)\,\mathrm{d}u\,.$$

Modified Offered Load

Service CDF

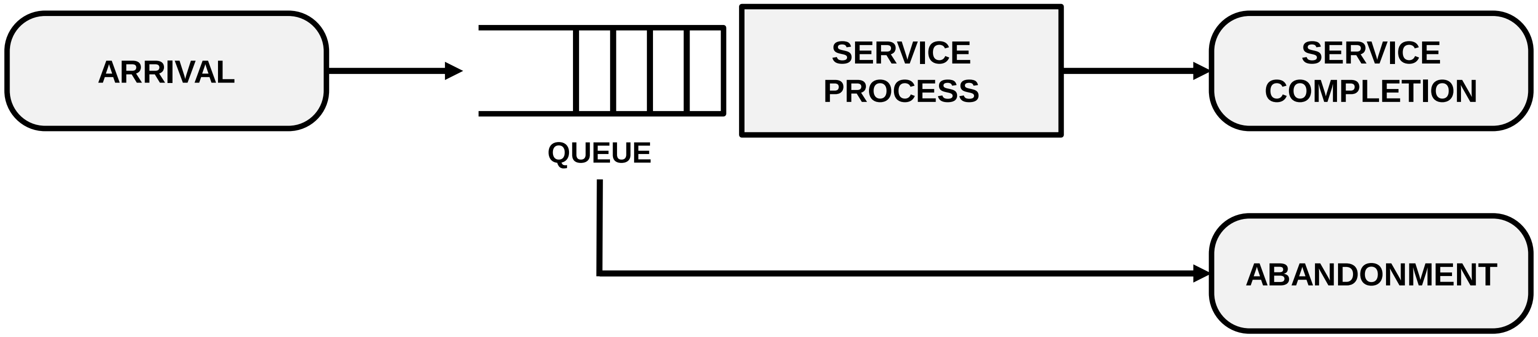
Original arrival rate

- In order to solve the stationary *M/G/s+G,* we adopt two approaches:
  - We simulate the *M/G/s+G* queue.
  - We use the approximation of Whitt (1995). Note that Whitt (1995) approximates the *M/G/s+G* queue by means of an *M/M/s+M* queue.

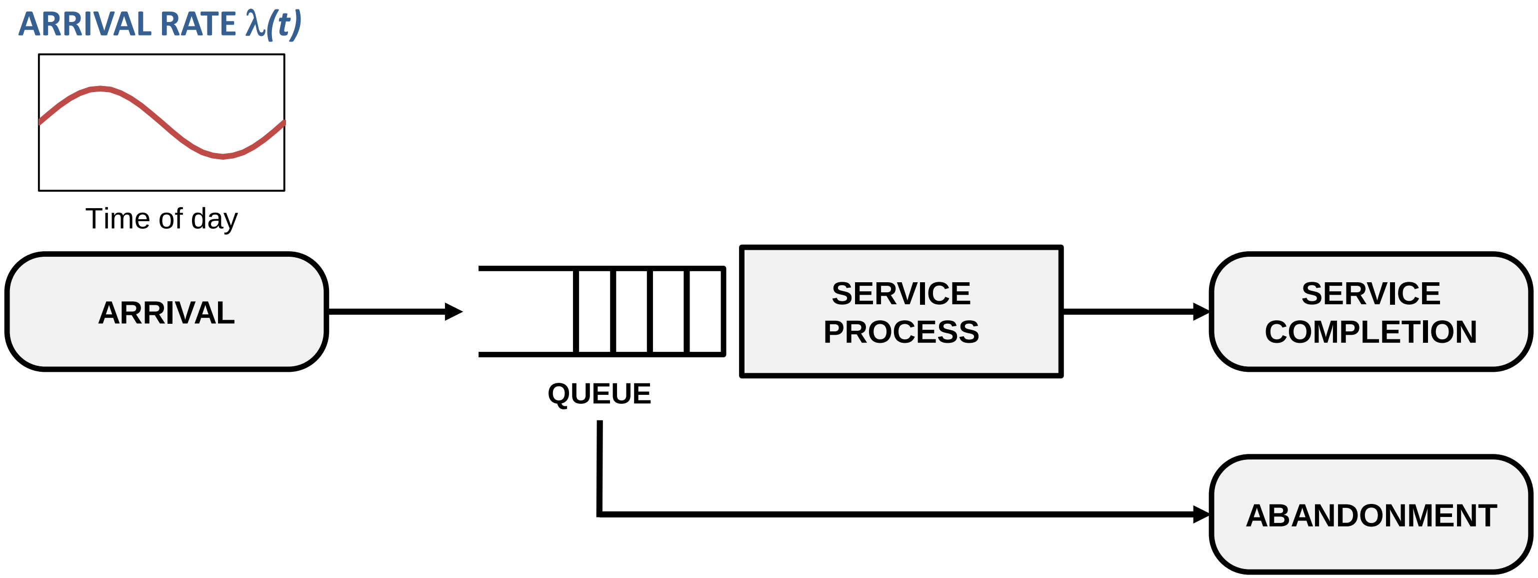
Methodology
G-RAND (General Randomization)

- Approximation of the $G(t)/G(t)/s(t)+G(t)$ queue
- Randomization/Uniformization method => observes the state of the system at discrete moments in time
- The more often you observe the system, the more accurate the method
- All-around method that allows the stationary as well as the transient analysis of a wide range of performance measures
- Uses Phase-Type distributions to approximate the general arrival, service, and abandonment processes
- In our experiment, we use simple PH distributions that match the first two moments.
- Time in between observations: 0.125, 0.25, 0.5, 1, 2 minutes

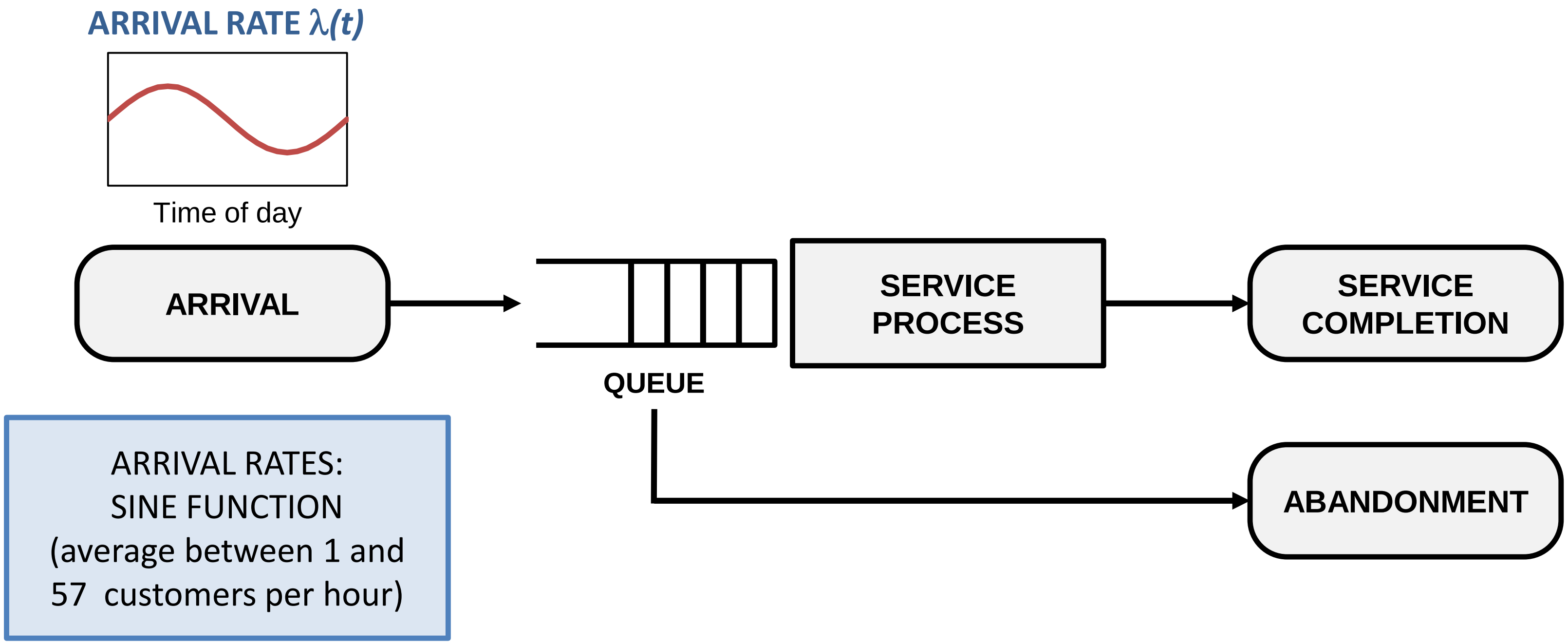
Experiment Setup



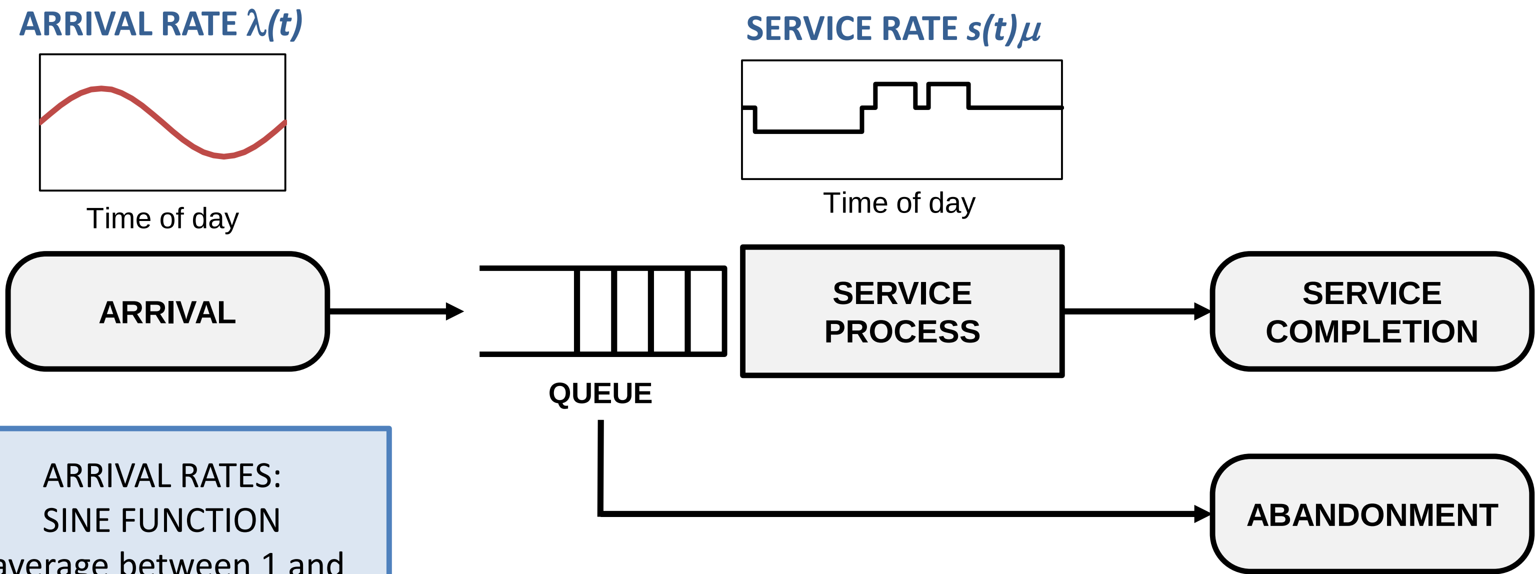
Experiment Setup



Experiment Setup



Experiment Setup

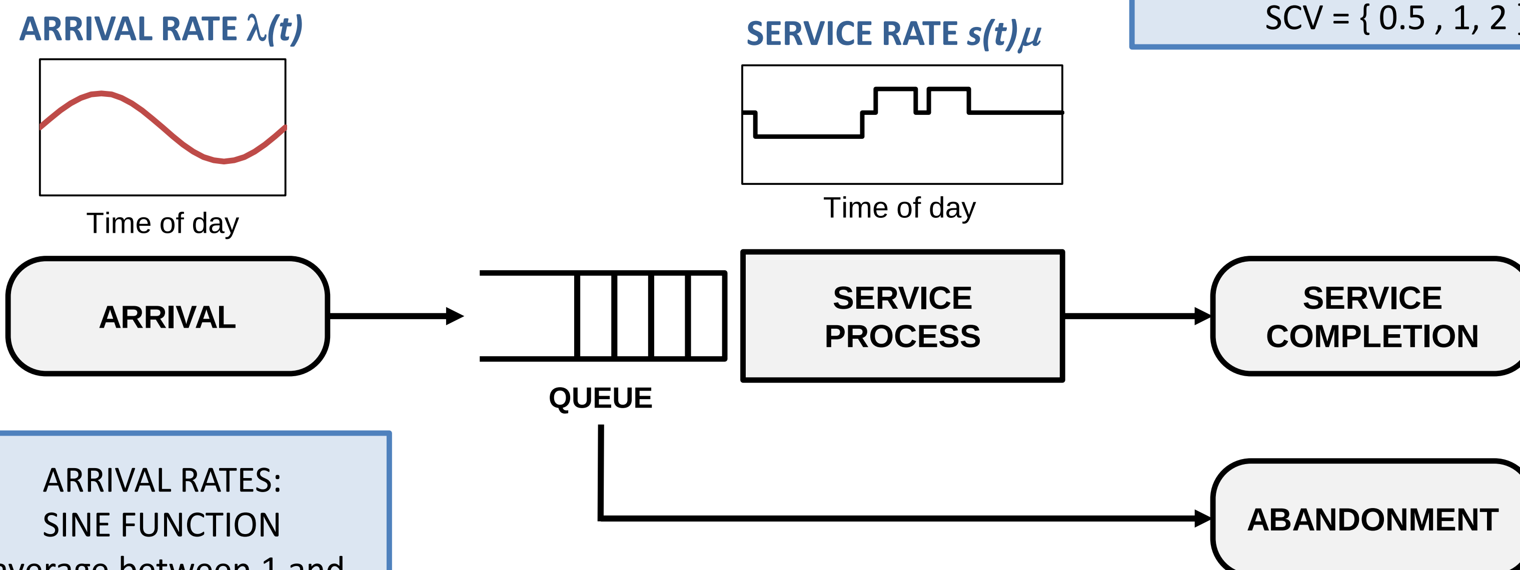


ARRIVAL RATES:
SINE FUNCTION
(average between 1 and
57 customers per hour)

Experiment Setup

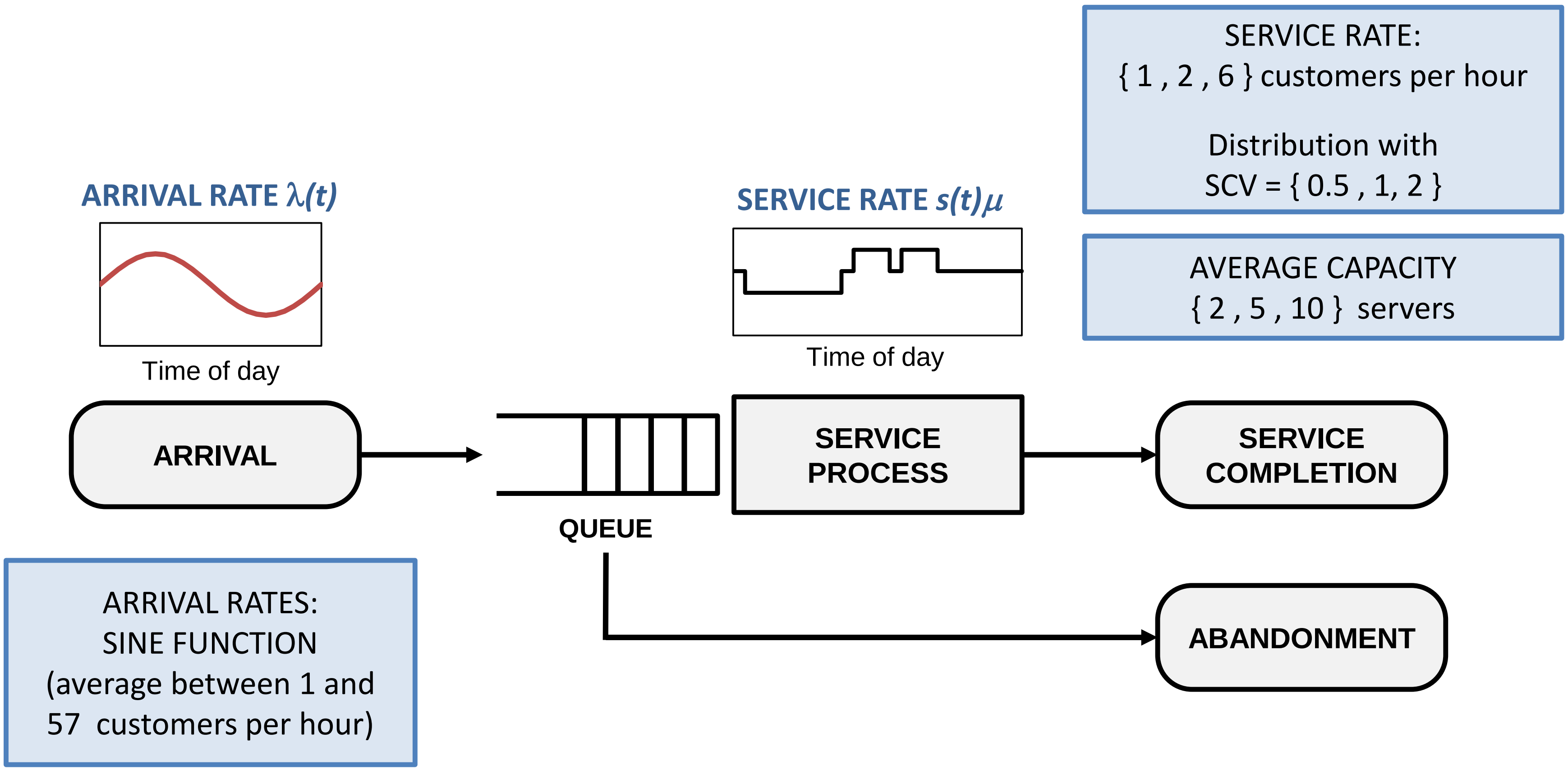
SERVICE RATE:
{ 1 , 2 , 6 } customers per hour

Distribution with
SCV = { 0.5 , 1, 2 }

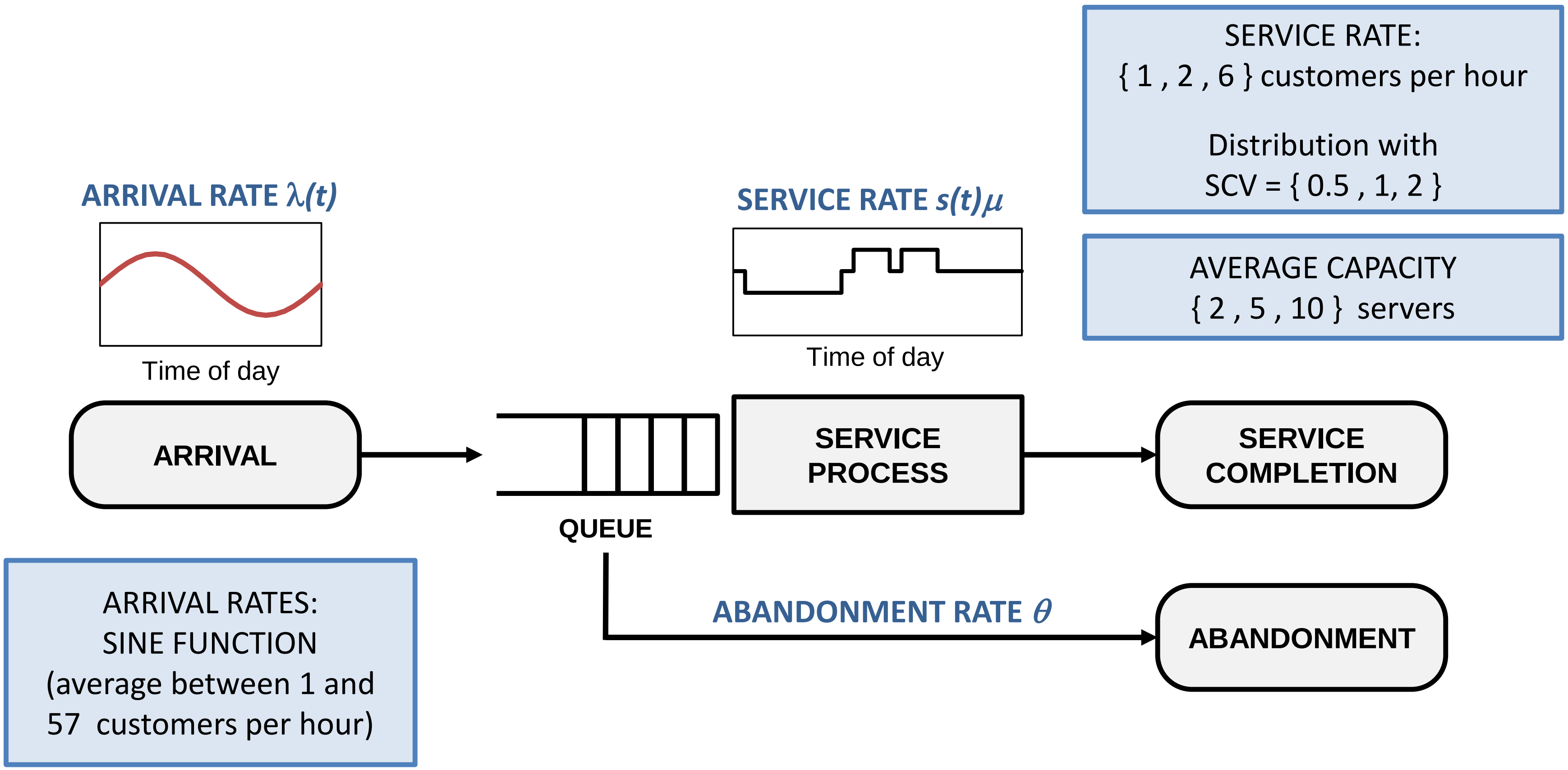


ARRIVAL RATES:
SINE FUNCTION
(average between 1 and
57 customers per hour)

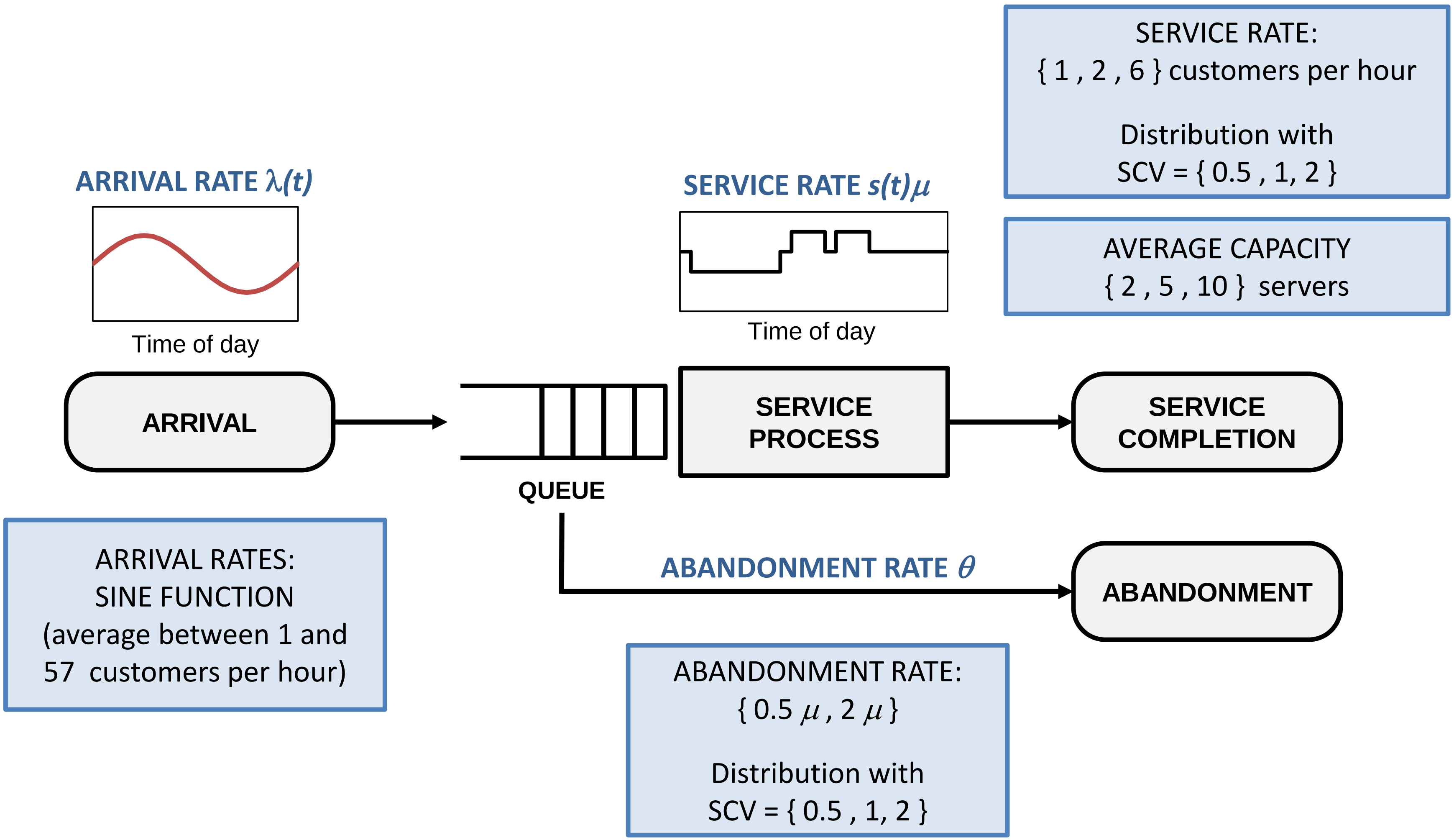
Experiment Setup



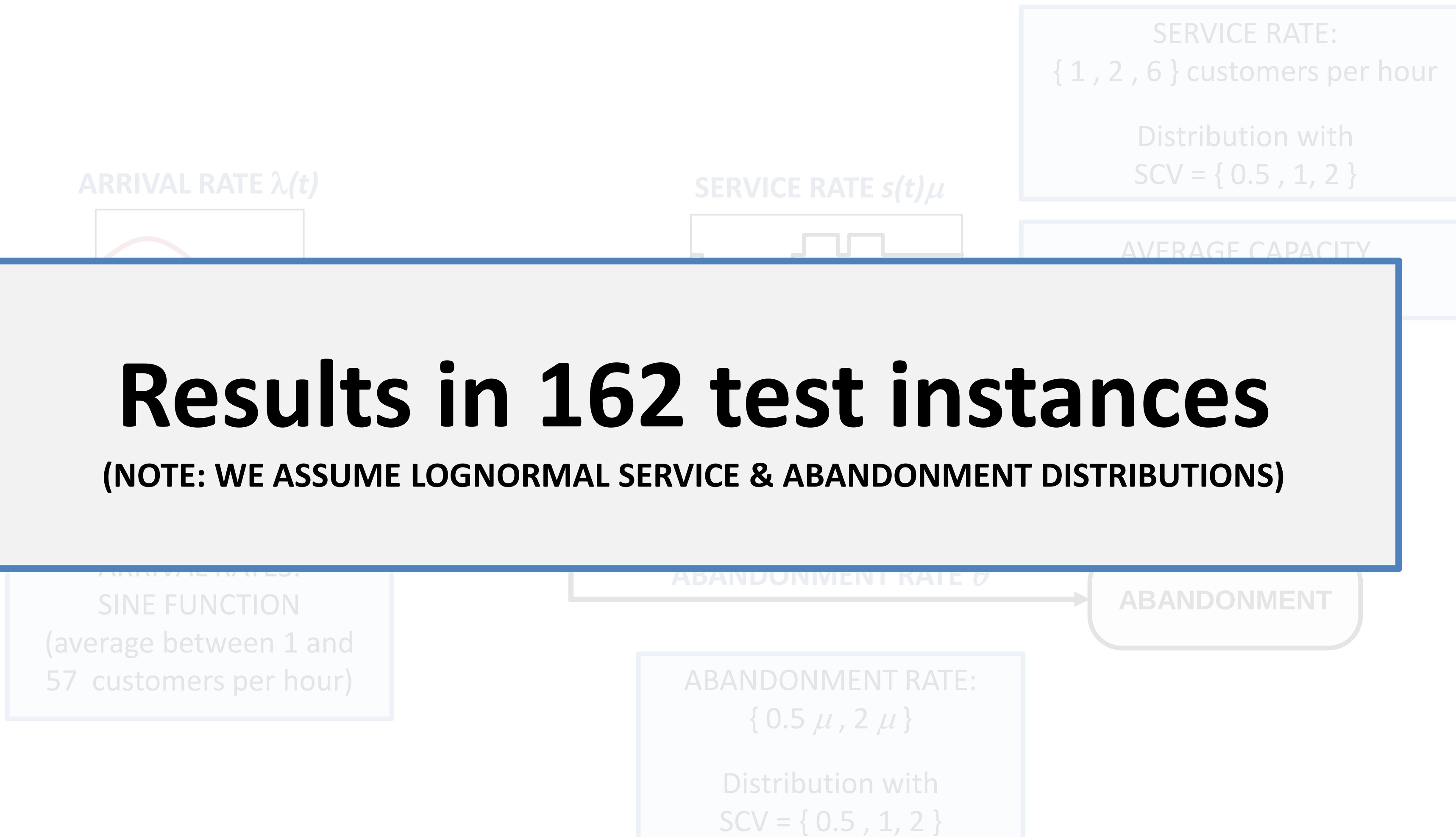
Experiment Setup



Experiment Setup



Experiment Setup



Results in 162 test instances

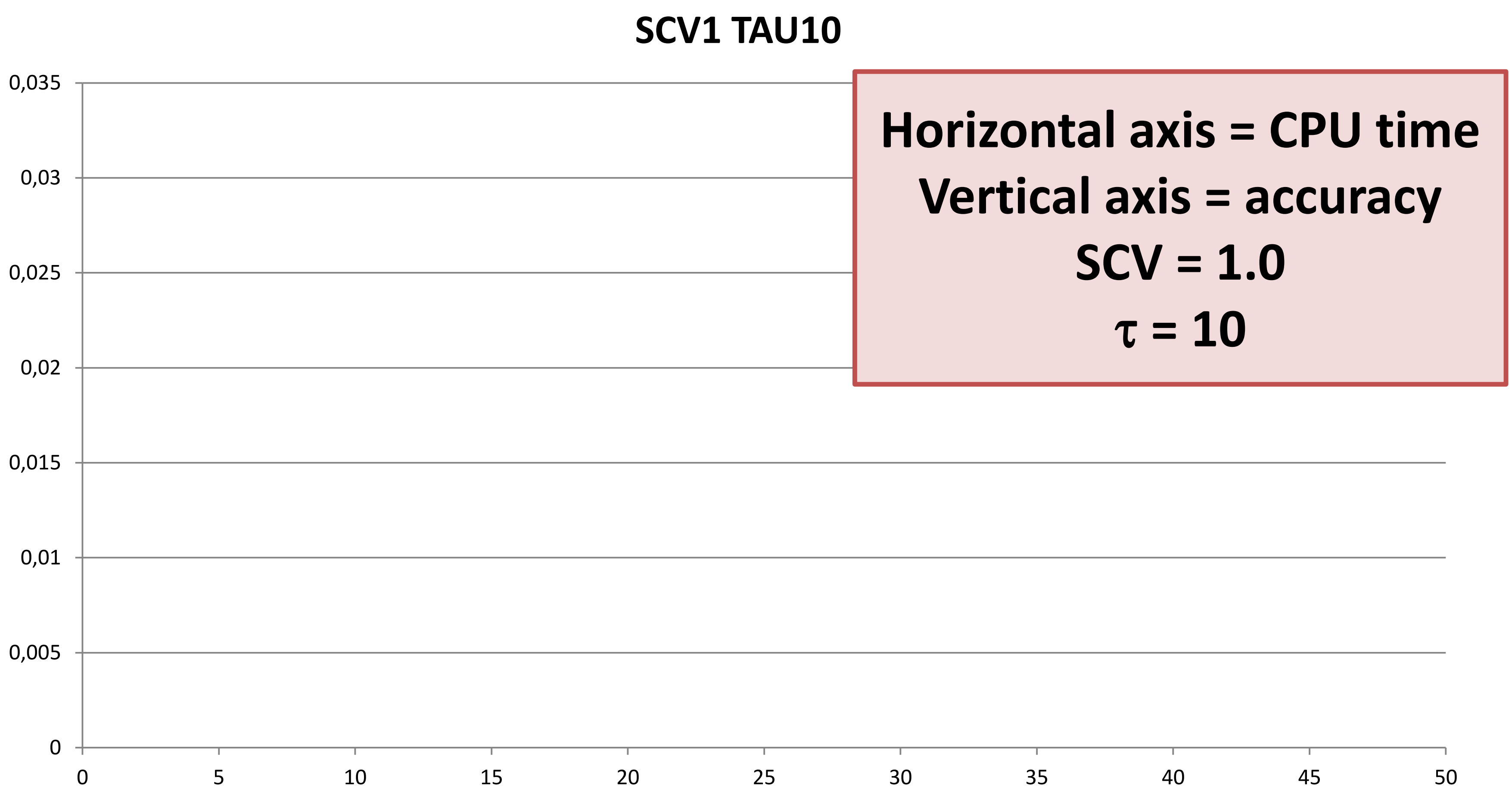
(NOTE: WE ASSUME LOGNORMAL SERVICE & ABANDONMENT DISTRIBUTIONS)

Performance Evaluation

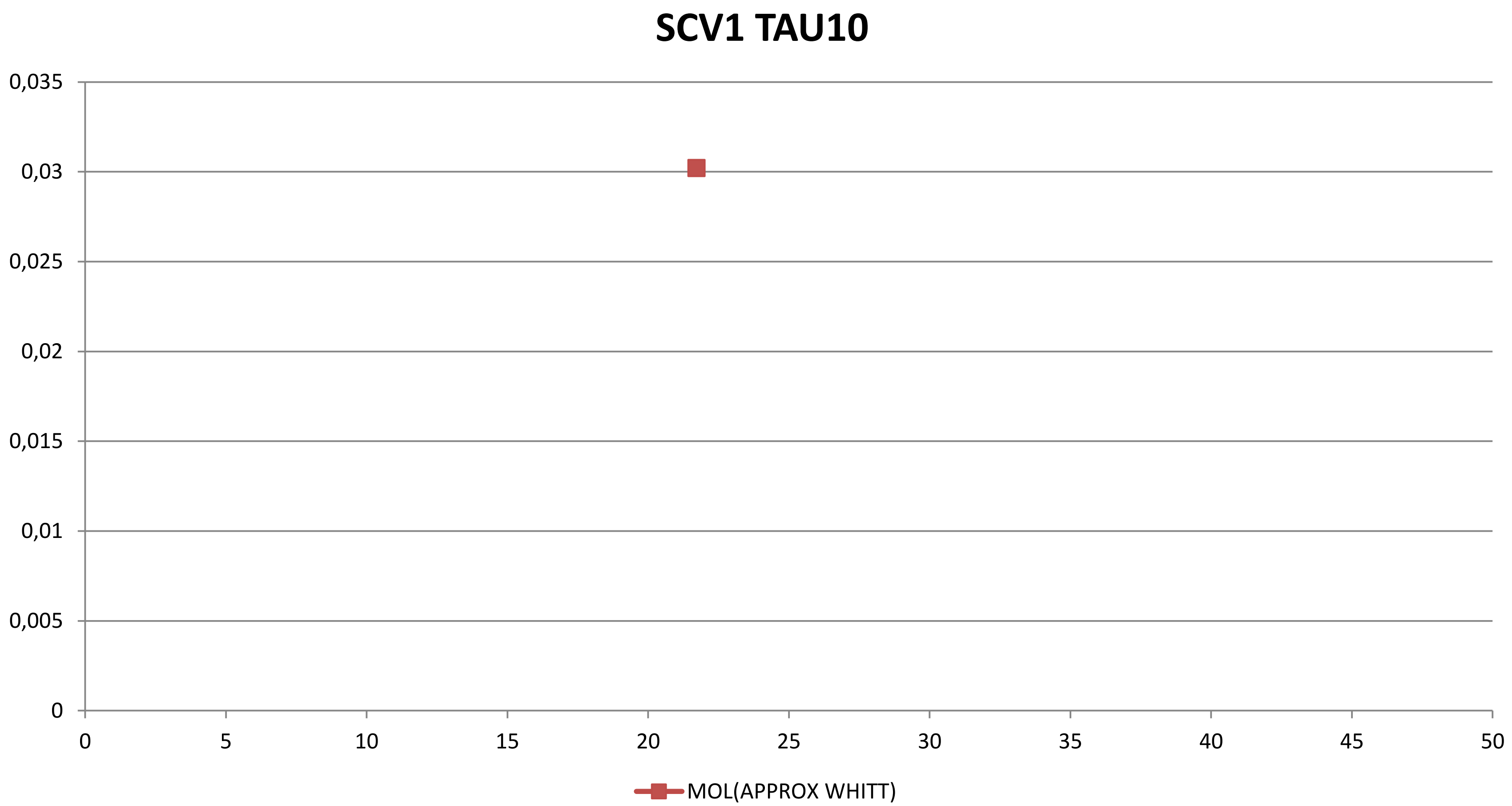
- The methods are compared based on accuracy & CPU time
- All tests are run on a Intel I7 3.4 GHz with 8 GB RAM
- Accuracy is expressed as the mean absolute error:

$$(1/T) * \Sigma|Pr(WAIT>\tau)_{TRUE} - Pr(WAIT>\tau)_{EST}|$$

Results

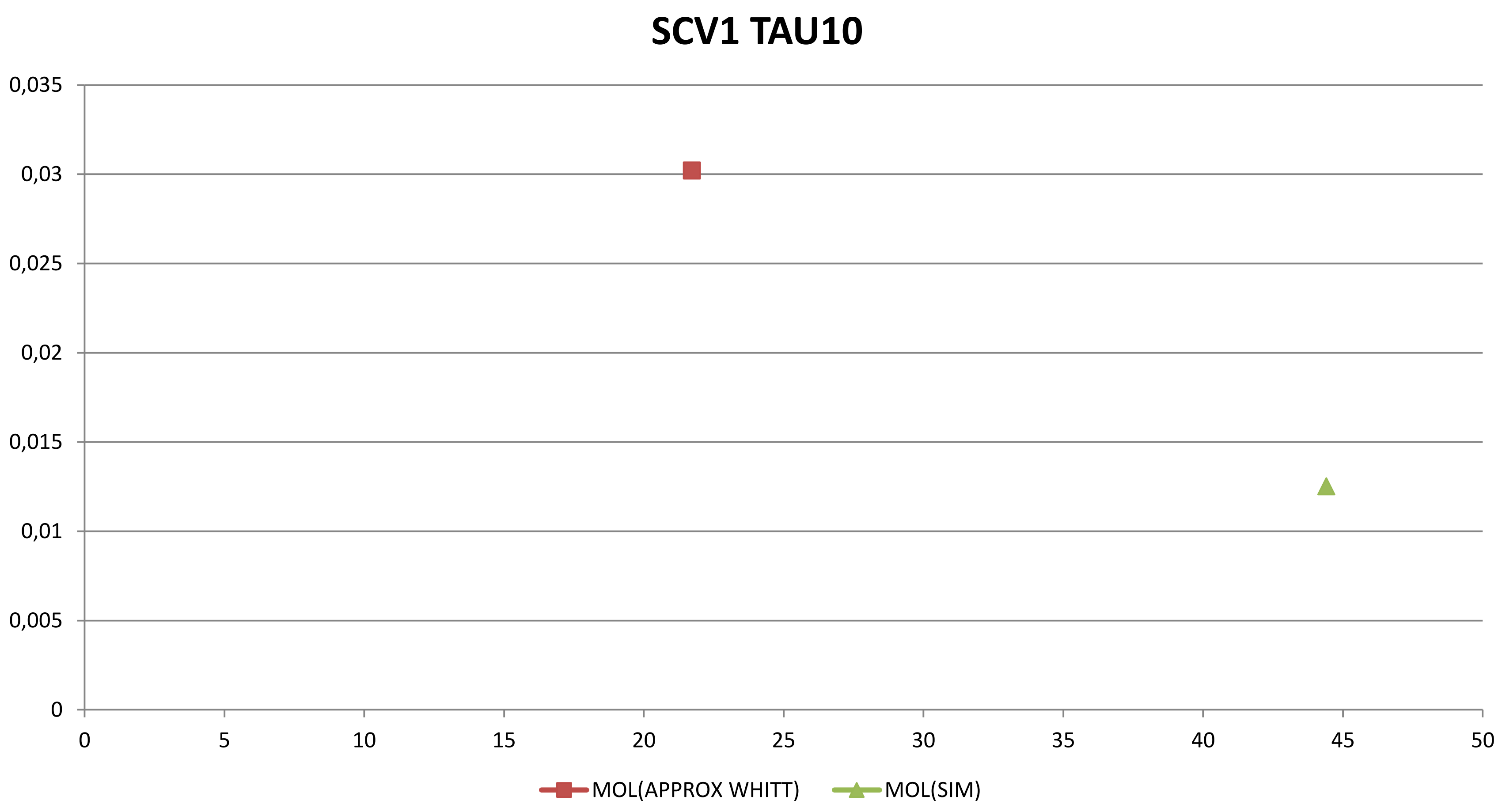

SCV1 TAU10
Horizontal axis = CPU time
Vertical axis = accuracy
SCV = 1.0
τ = 10
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
5
10
15
20
25
30
35
40
45
50

Results



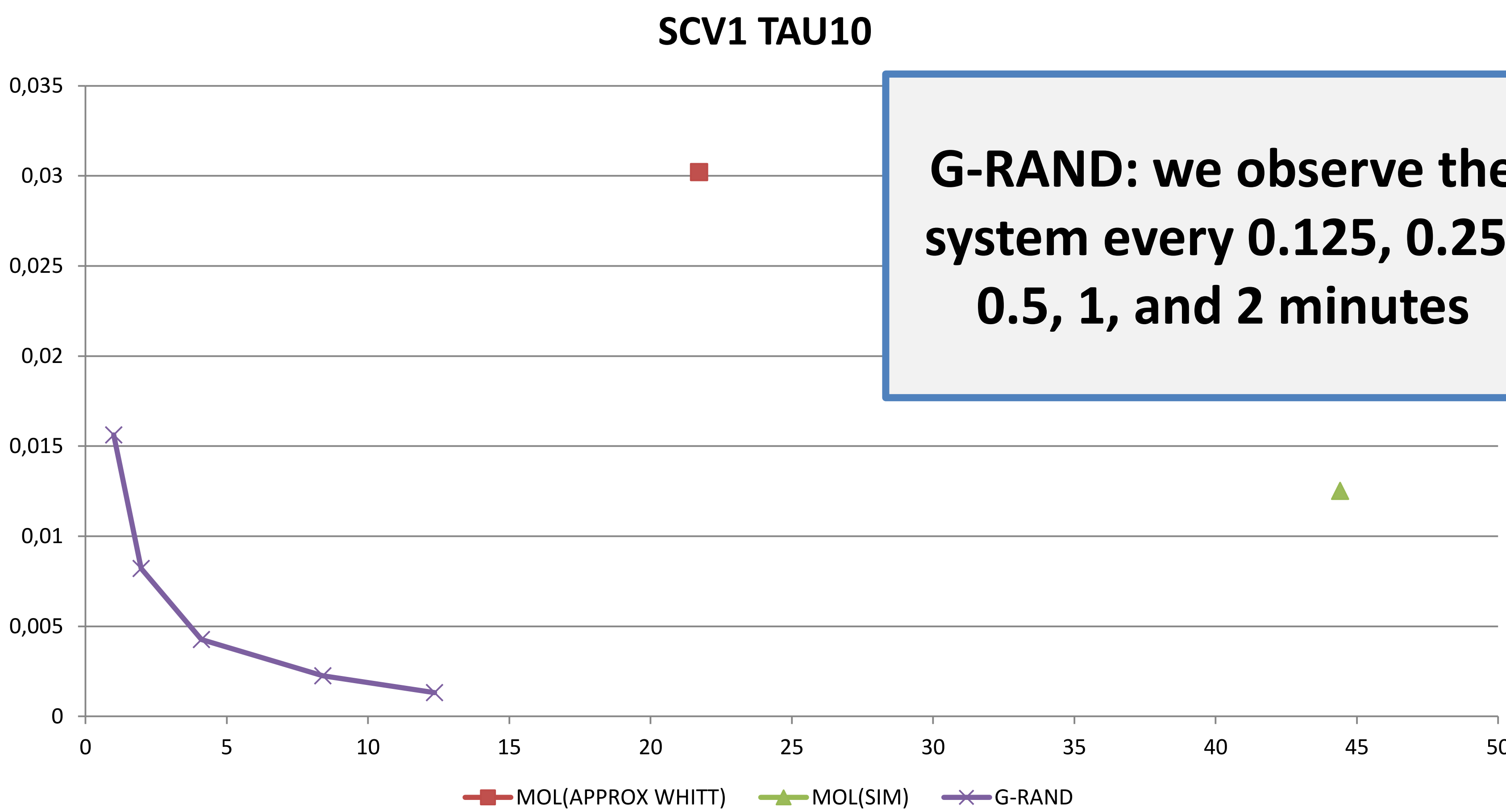
SCV1 TAU10
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
5
10
15
20
25
30
35
40
45
50
MOL(APPROX WHITT)

Results



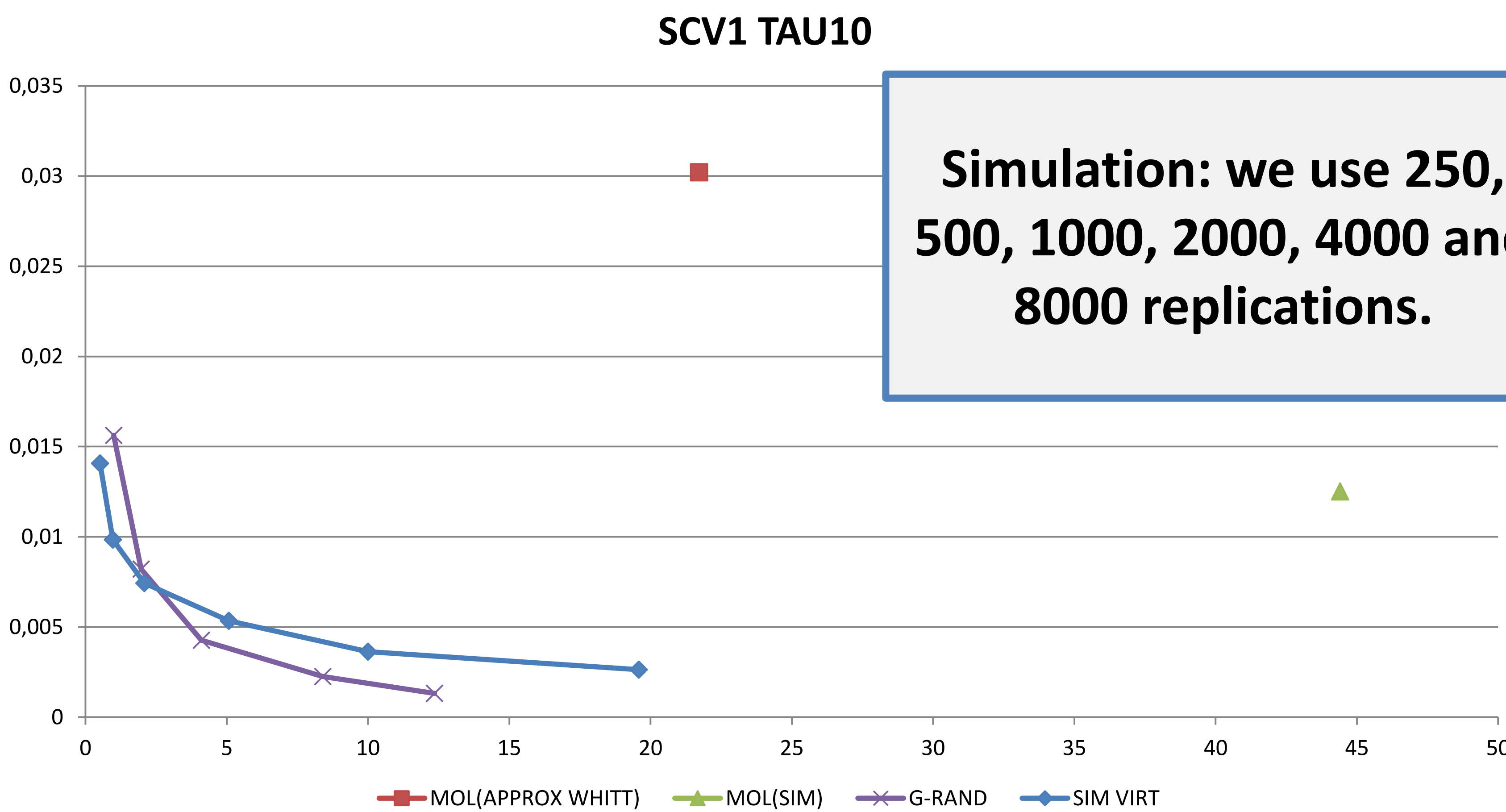
SCV1 TAU10
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
5
10
15
20
25
30
35
40
45
50
MOL(APPROX WHITT)
MOL(SIM)

Results



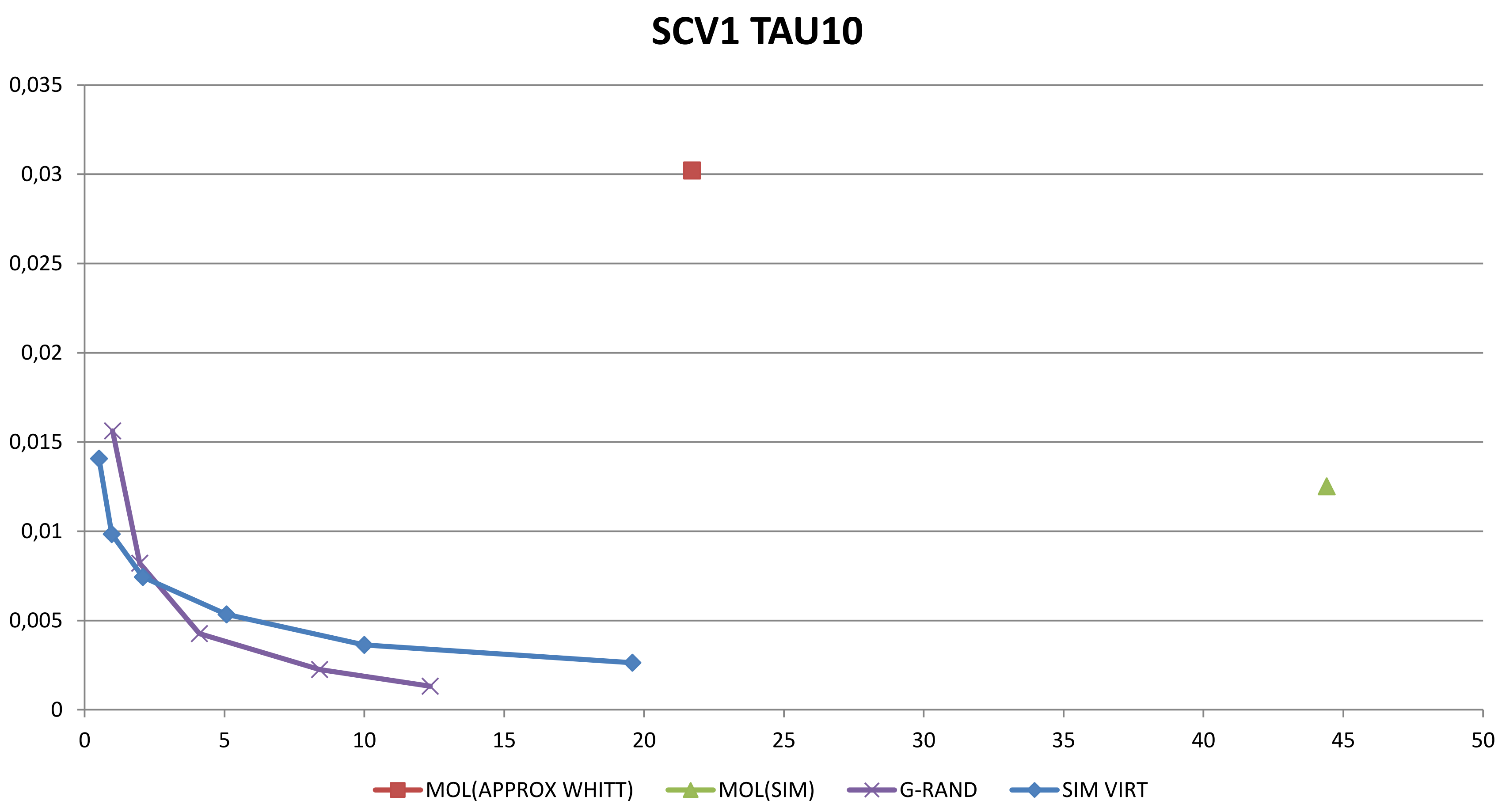
G-RAND: we observe the system every 0.125, 0.25, 0.5, 1, and 2 minutes

Results



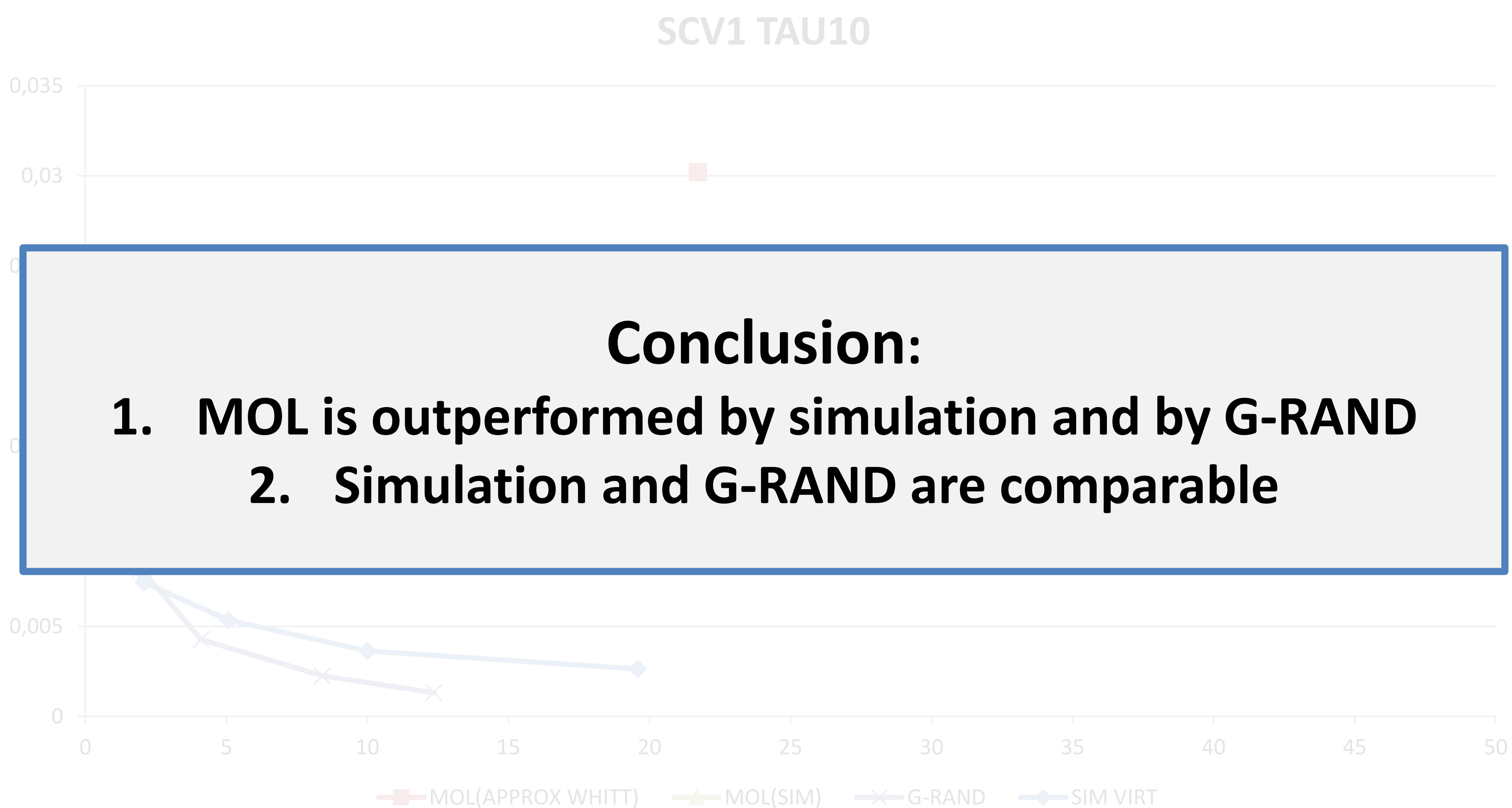
Simulation: we use 250, 500, 1000, 2000, 4000 and 8000 replications.

Results



SCV1 TAU10
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
5
10
15
20
25
30
35
40
45
50
MOL(APPROX WHITT)
MOL(SIM)
G-RAND
SIM VIRT

Results

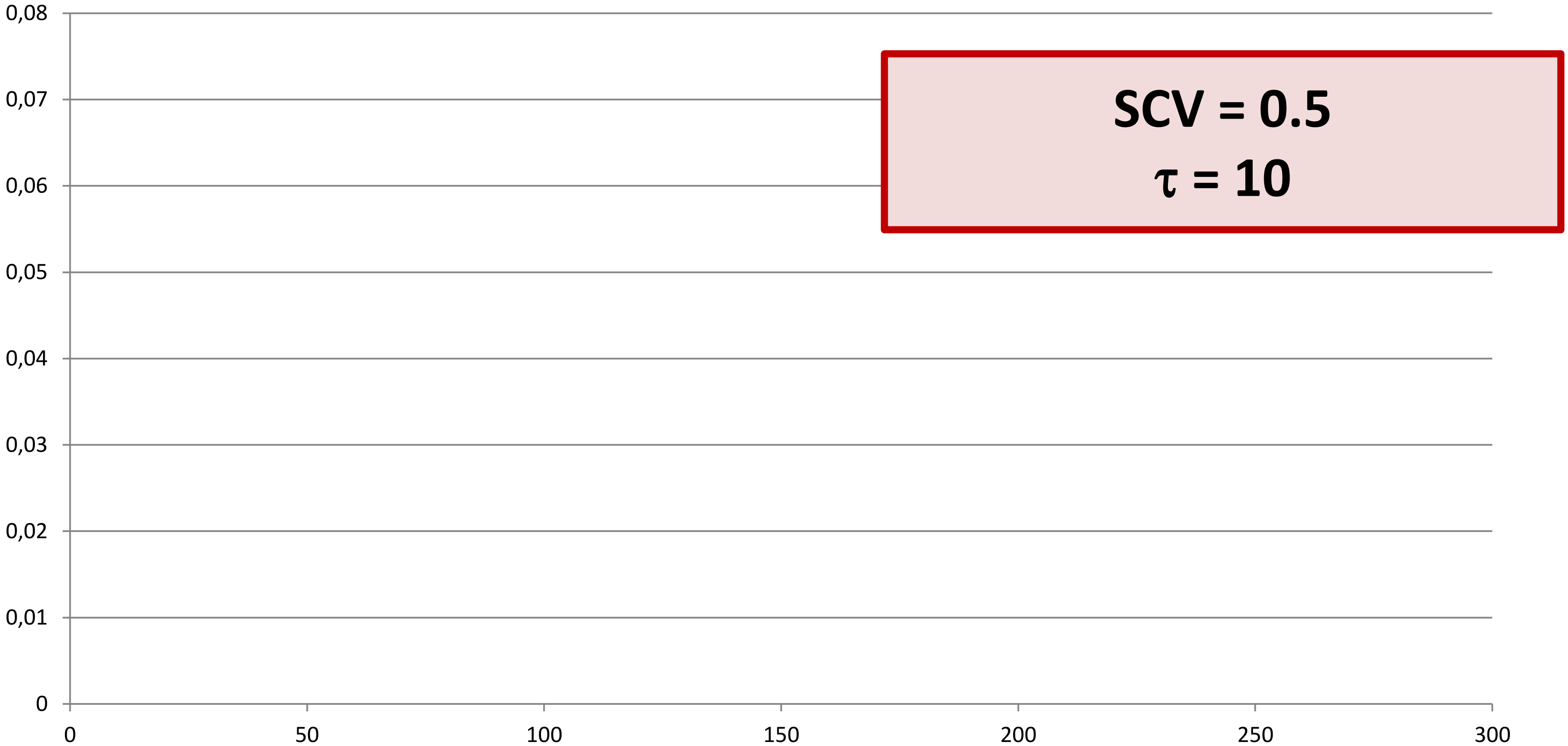


Conclusion:

1. **MOL is outperformed by simulation and by G-RAND**
2. **Simulation and G-RAND are comparable**

Results

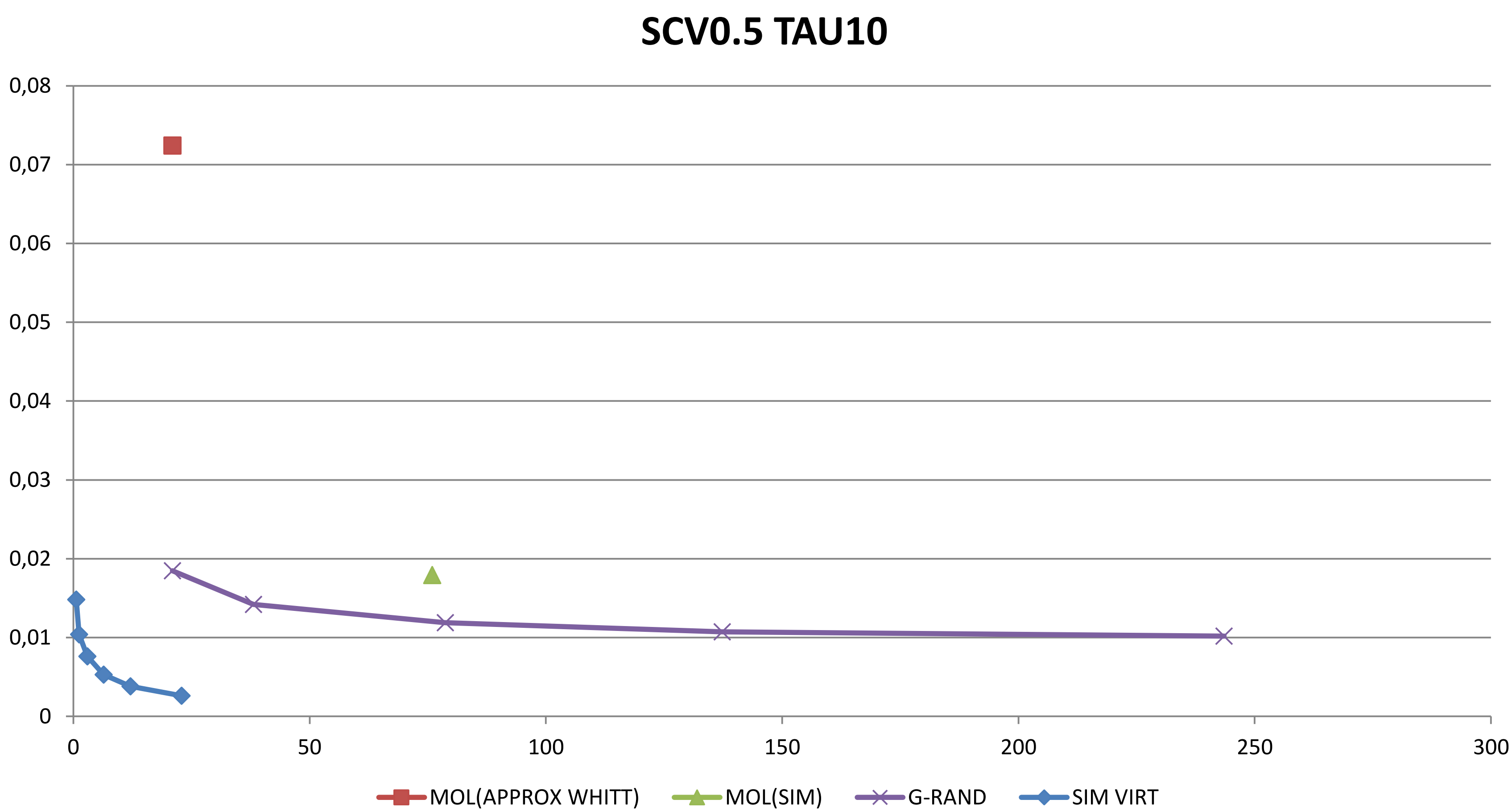
SCV0.5 TAU10



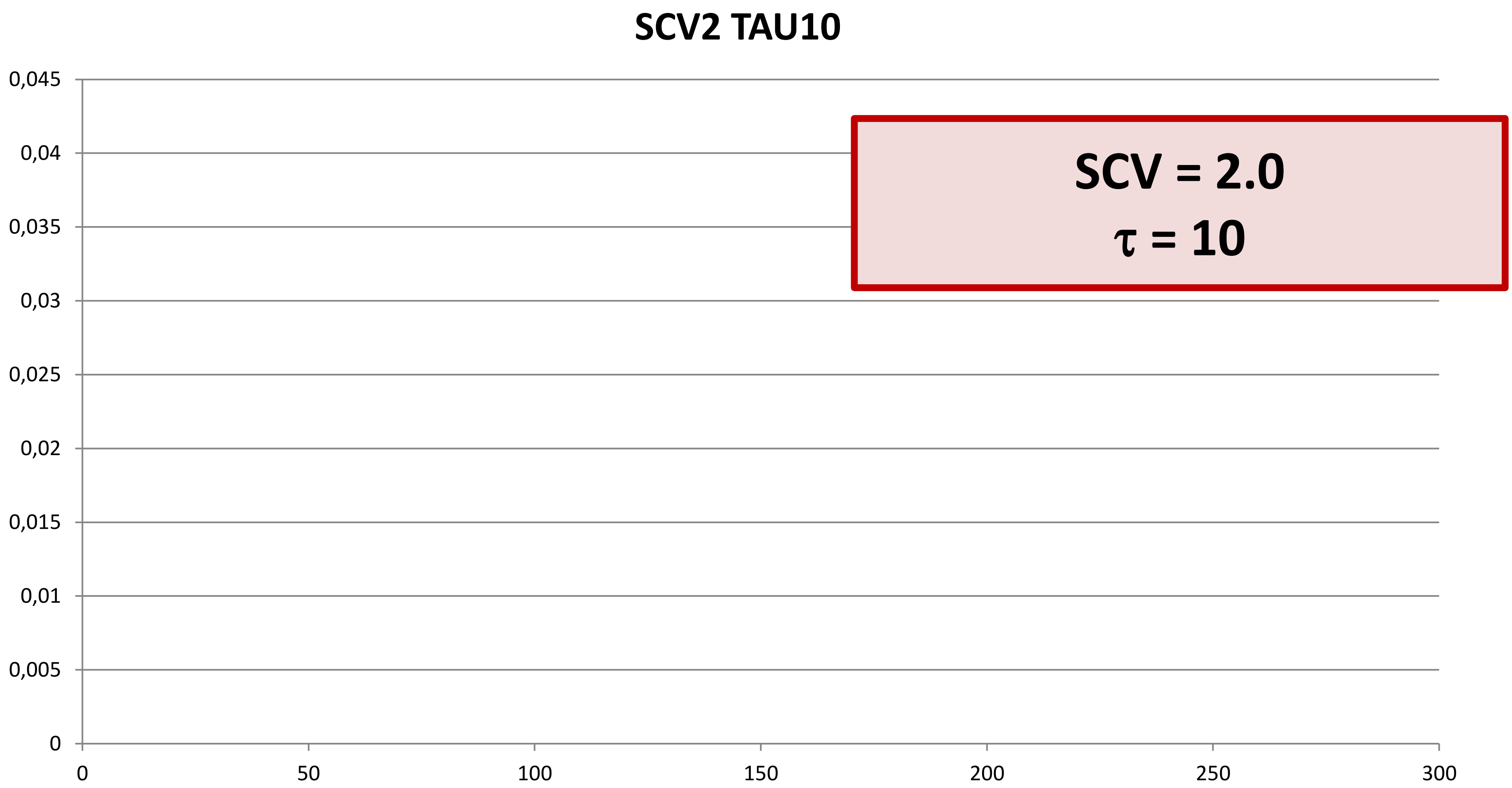
SCV = 0.5

τ = 10

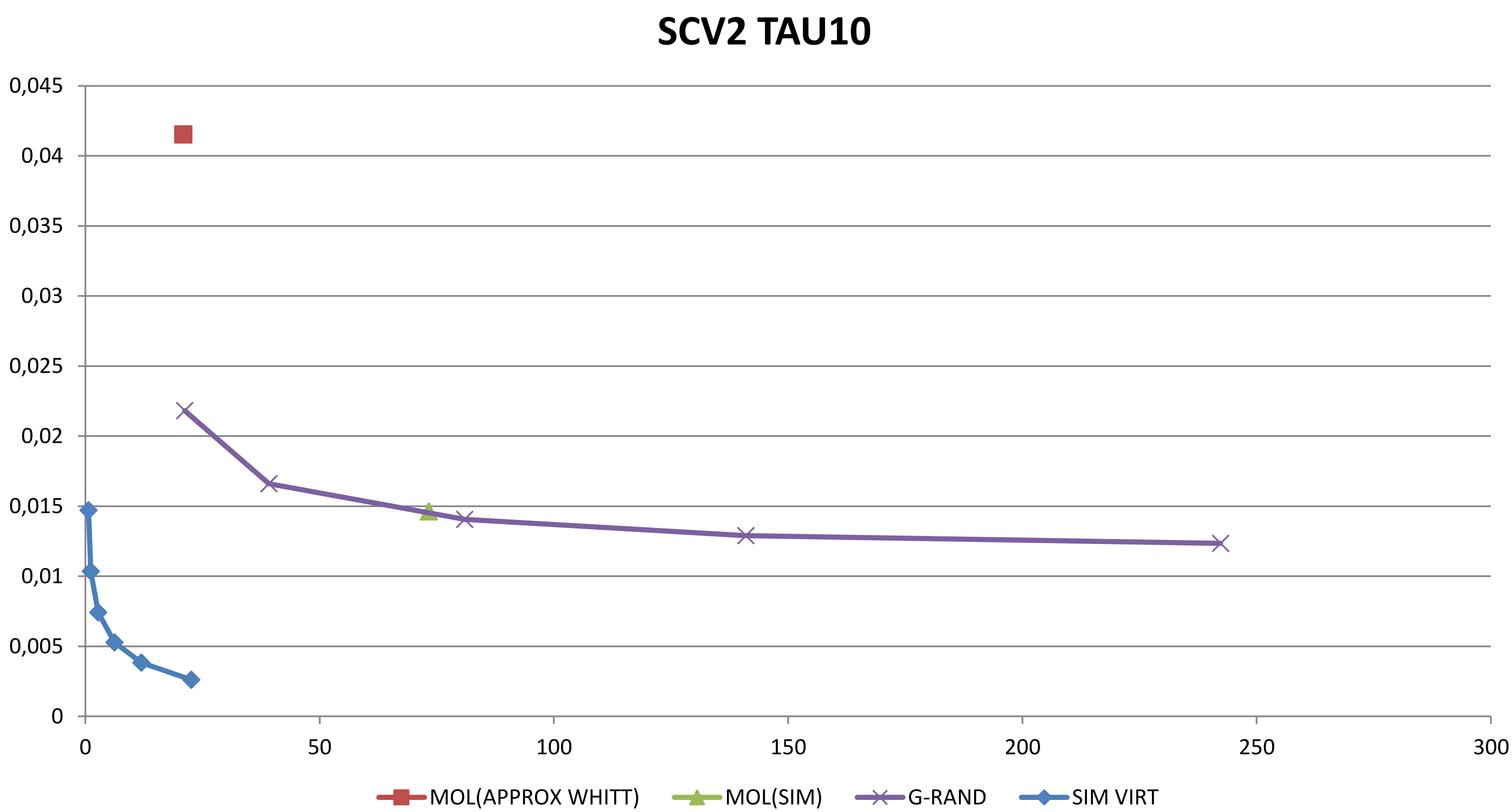
Results


SCV0.5 TAU10
0,08
0,07
0,06
0,05
0,04
0,03
0,02
0,01
0
0
50
100
150
200
250
300
MOL(APPROX WHITT)
MOL(SIM)
G-RAND
SIM VIRT

Results


SCV2 TAU10
0,045
0,04
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
50
100
150
200
250
300
SCV = 2.0
τ = 10

Results


SCV2 TAU10
0,045
0,04
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
50
100
150
200
250
300
MOL(APPROX WHITT)
MOL(SIM)
G-RAND
SIM VIRT

Conclusions

- Results are similar for other values of τ
- Conclusions:
  - MOL is outperformed by simulation as well as by G-RAND
  - In general simulation outperforms G-RAND
- Note however:
  - The accuracy of G-RAND can be improved by adopting more precise moment-matching procedures (in our experiment, we only match the first two moments of the lognormal distributions).
  - Computing the waiting time distribution is a CPU-intensive process as it requires the analysis of a death process. Other KPI's (e.g., queue size, abandonment probability) can be calculated much faster.