SCHOOL OF RANDOM DEPARTMENT NAMES

This is the title of the template report

Thomas Mari, Mário Uhrík
May 22, 2018

The subject of this internship is first modeling some system in the PrismGSMP. That is a continuous system with rates instead of probabilistic transition. Transitiom are syncronized with event which follows some probabilistic laws like exponential law or dirac. One good property about the exponenial law is the events are markovian. That is the probability of the event of happening doenst depends on the time already spend.

1 QUEUE : A TOY EXAMPLE

We consider a D/M/1/n queue.

1.1 FIRST MODEL

1.2 SECOND MODEL(USING PHASE TYPE FITTING)

The timeout part is replaced by k μ transition following the exponential law of parameter μ. Here on the figure, k=5, which mean that you need 5 consecutive μ transition to $produce$.

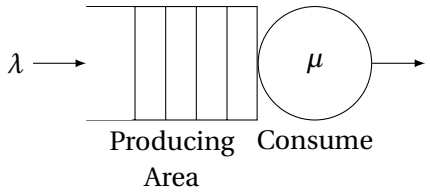

Figure 1.1: a queue

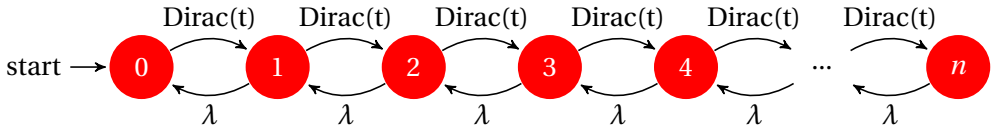

Figure 1.2: D/M/1/n queue with capacity n. The Dirac-distributed arrival event does not reset when the exponential service event occurs.

```
gsmp

const int qCapacity = 10;

const double timeout = 0.02;
const double lambda = 1/timeout;

module main
event prod = dirac(timeout);

qSize : [0..qCapacity] init 0;

[produce] (qSize < qCapacity)--prod -> (qSize' = qSize+1);
[consume] (qSize > 0) -> lambda: (qSize' = qSize - 1);

endmodule
```

Figure 1.3: PRISM GSMP model of a D/M/1/n queue as shown in 1.2.

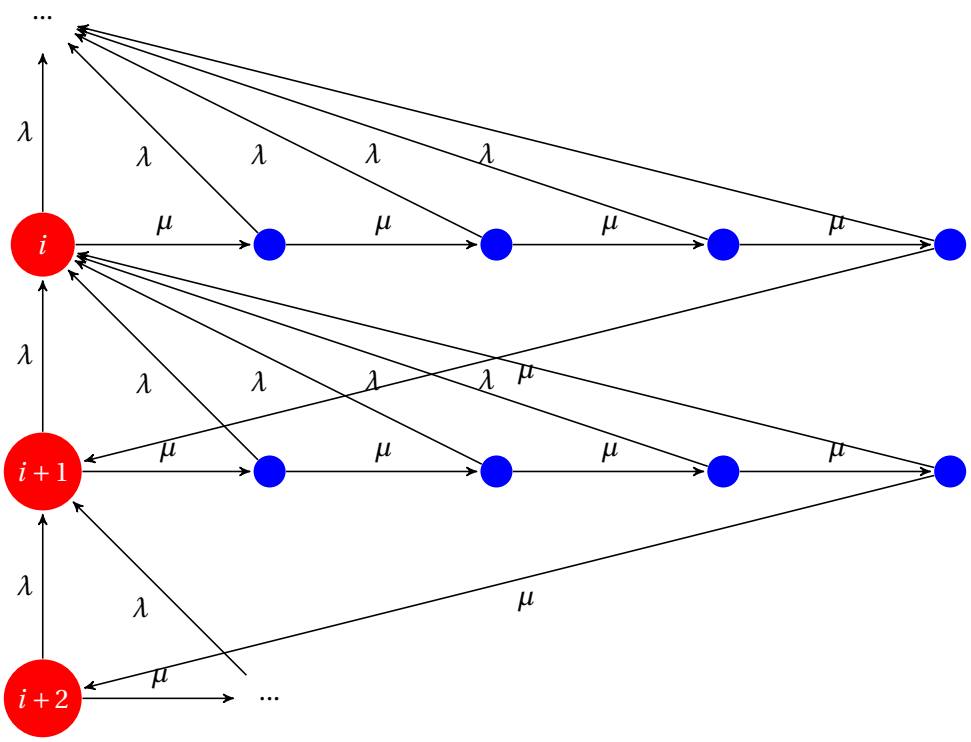

Figure 1.4: Queue with phase type fitting of parameter k.

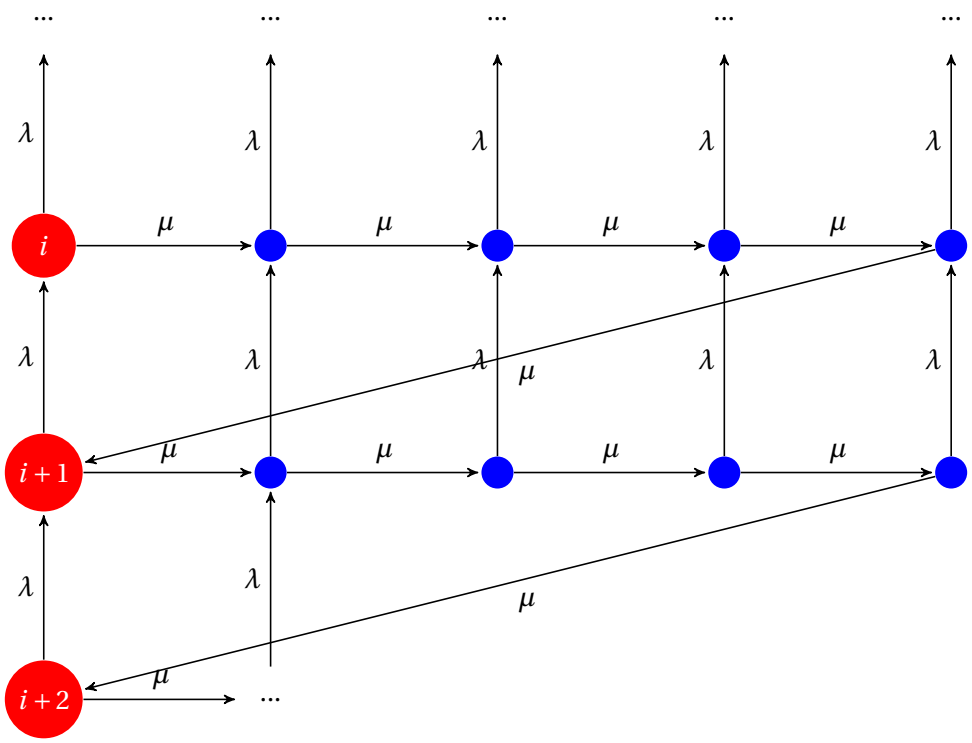

Figure 1.5: Queue with phase type fitting of parameter k.

```
gsmp

const int k = 10;
const int qCapacity = 10;

const double timeout = 10;
const double lambda = 0.2;


module main

qSize : [0..qCapacity] init 0;

[produce] (qSize < qCapacity) -> (qSize' = qSize+1);
[consume] (qSize > 0) -> lambda: (qSize' = qSize - 1);

endmodule

module trigger

i : [1..k+1];

[] i < k -> k/timeout : (i'=i+1);
[produce] i = k -> k/timeout : (i'=1);
//[consume] true -> (i'=1);

endmodule
```

Figure 1.6: PRISM CTMC model of the D/M/1/n queue as shown in 1.2 with phase-type fitting to approximate the deterministic arrivals. The phase-type module *trigger* is used as suggested on the PRISM website.

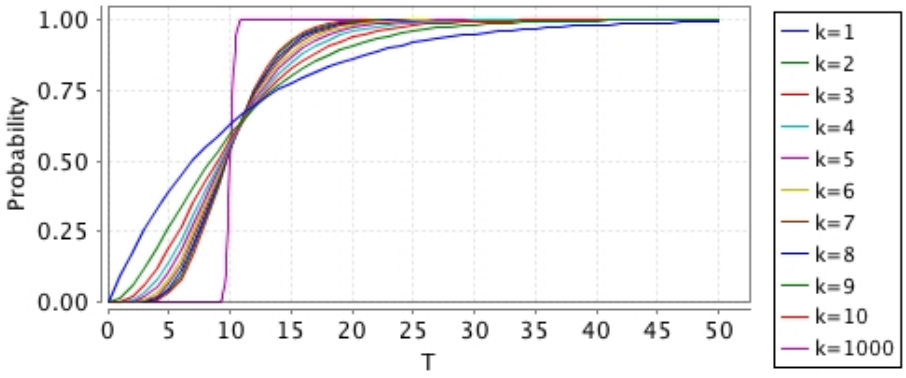

Figure 1.7: The effect of k, the number of time points within the sequence of exponential transitions, on the shape of the resulting phase-type distribution. Increasing k makes the phase-type distribution more deterministic. This picture was taken from the PRISM website.

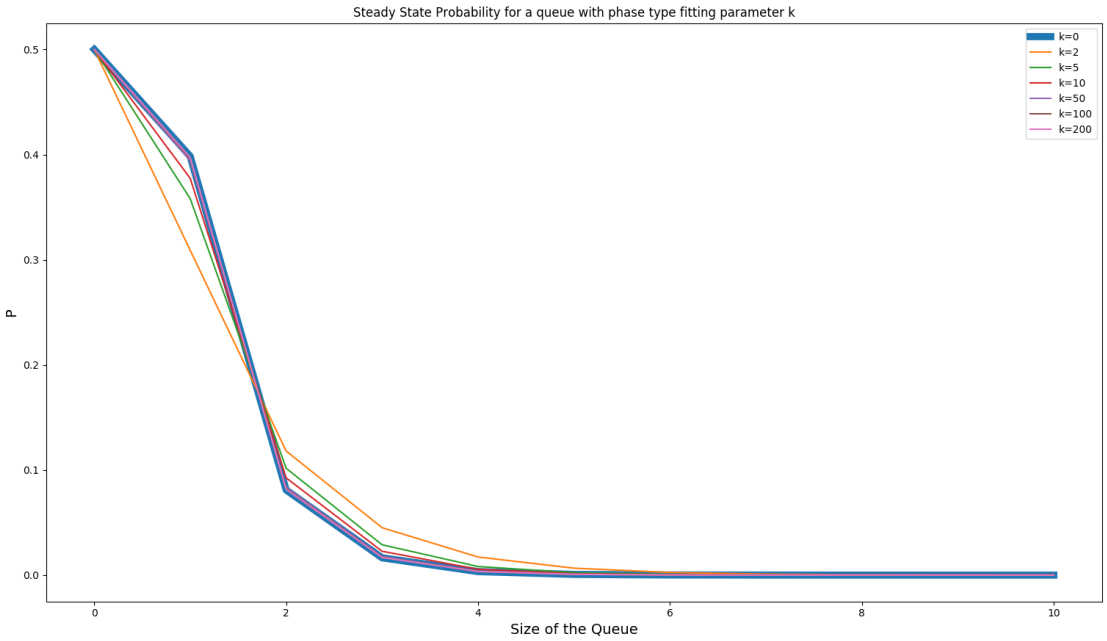

Figure 1.8: The blue line is the result without phase type fitting.

1.3 FINDING μ

According to Prism documentation on phase type fittimg, we should take $\mu = t/k$. The mean of diract(t) is t, and the mean of exponential(μ) is $1/\mu$. Hence $\mu = t/k$.

1.4 RESULTS

For a queue capacity of 10, we compare the steady state probabilities according to k the parameter of phase type fitting.

1.4.1 PARAMETERS

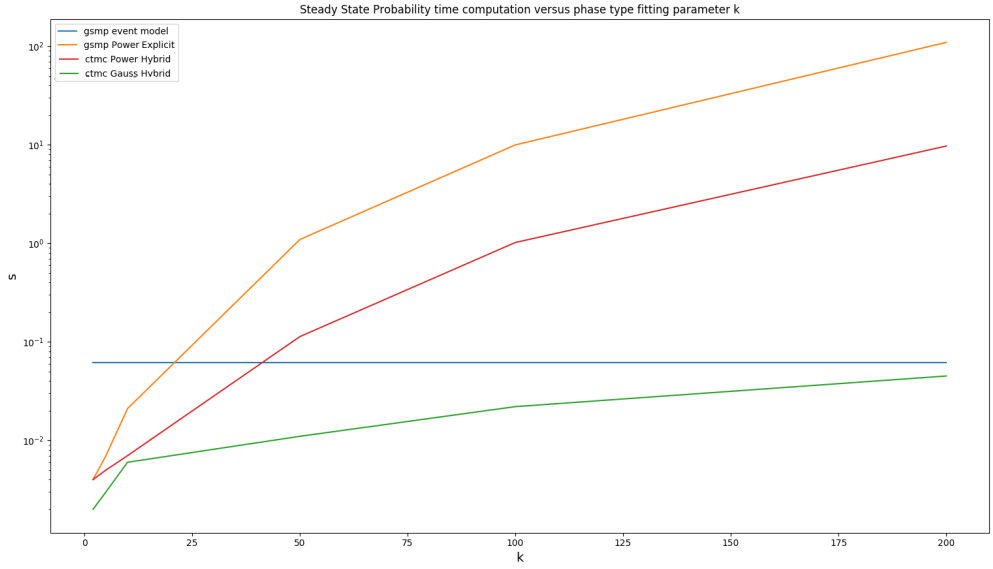

Figure 1.9: Computation time for different values of k. The growing lines represent computations of models with phase-type distributions. For about $k > 200$, GSMP implementation starts outperforming all phase-typed computations, despite the GSMP implementation being relatively poorly optimized.

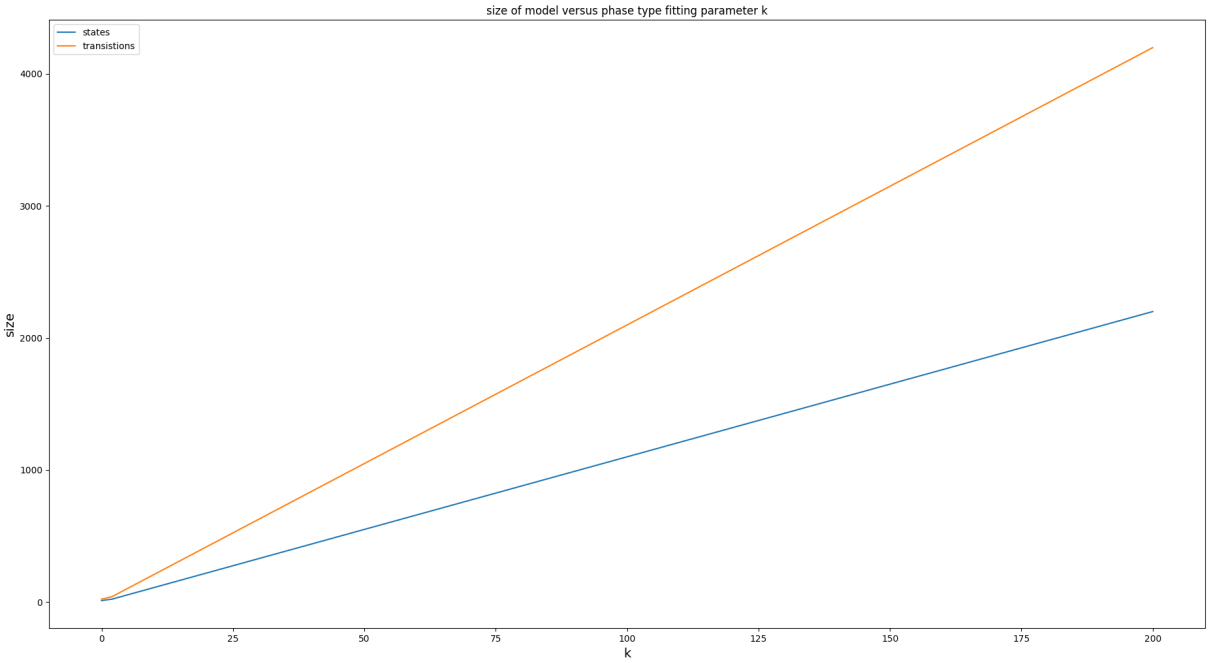

Figure 1.10: Size of model for different values of k.

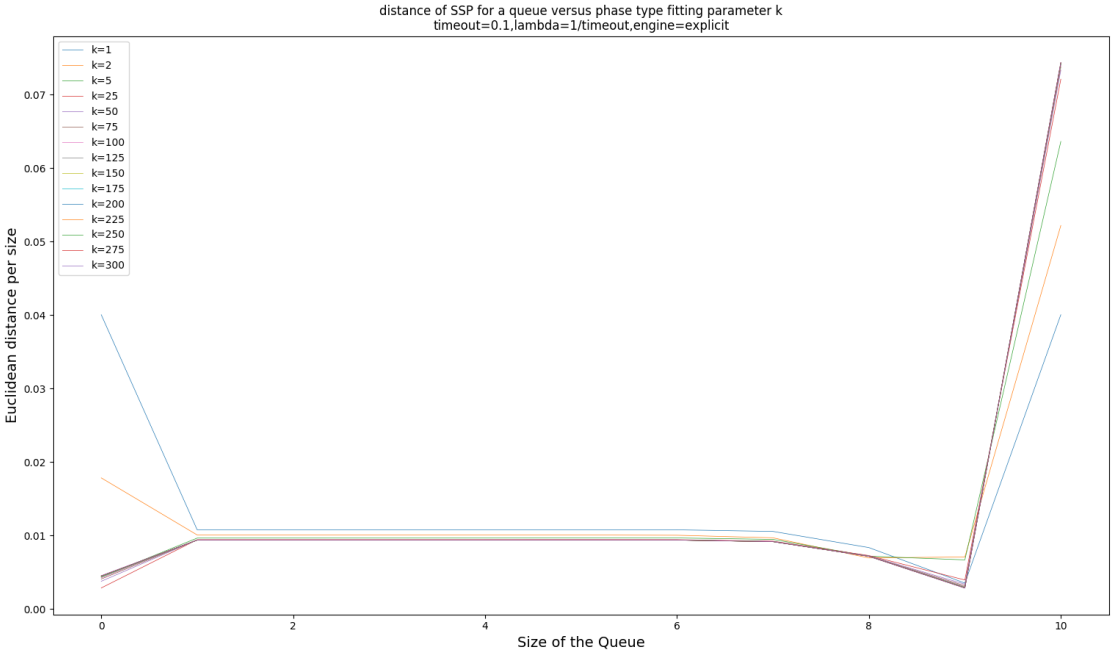

Figure 1.11: Distance of result of the SSP computation between the event model and the PTF model using explicit engine, $t = 0.1, \lambda = 1/t, \epsilon = 10^{-5}$

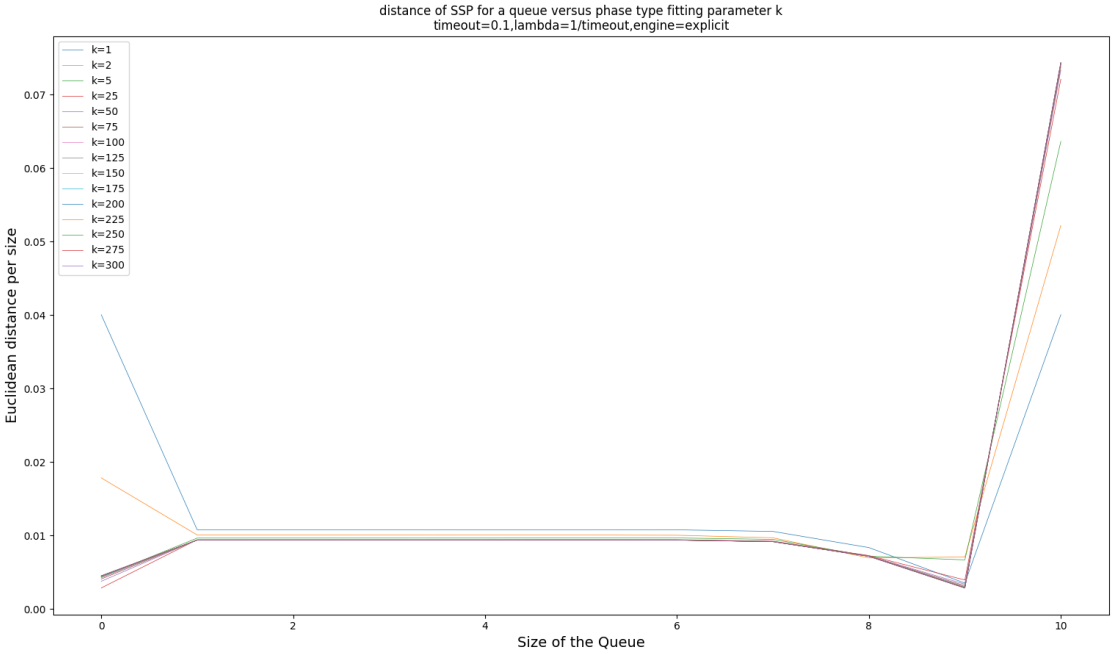

Figure 1.12: Distance of result of the SSP computation between the event model and the PTF model using hybrid engine, $t = 0.1, \lambda = 1/t, \epsilon = 10^{-5}$

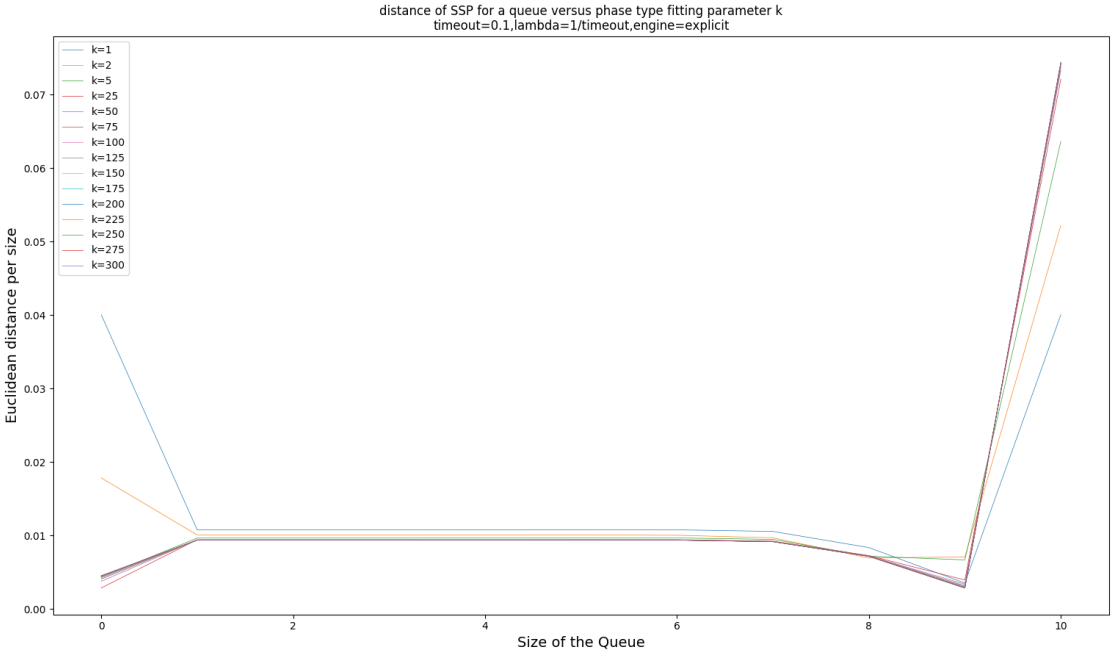

Figure 1.13: Distance of result of the SSP computation between the event model and the PTF model using explicit engine, $t = 0.1, \lambda = 1/t, \epsilon = 10^{-5}$

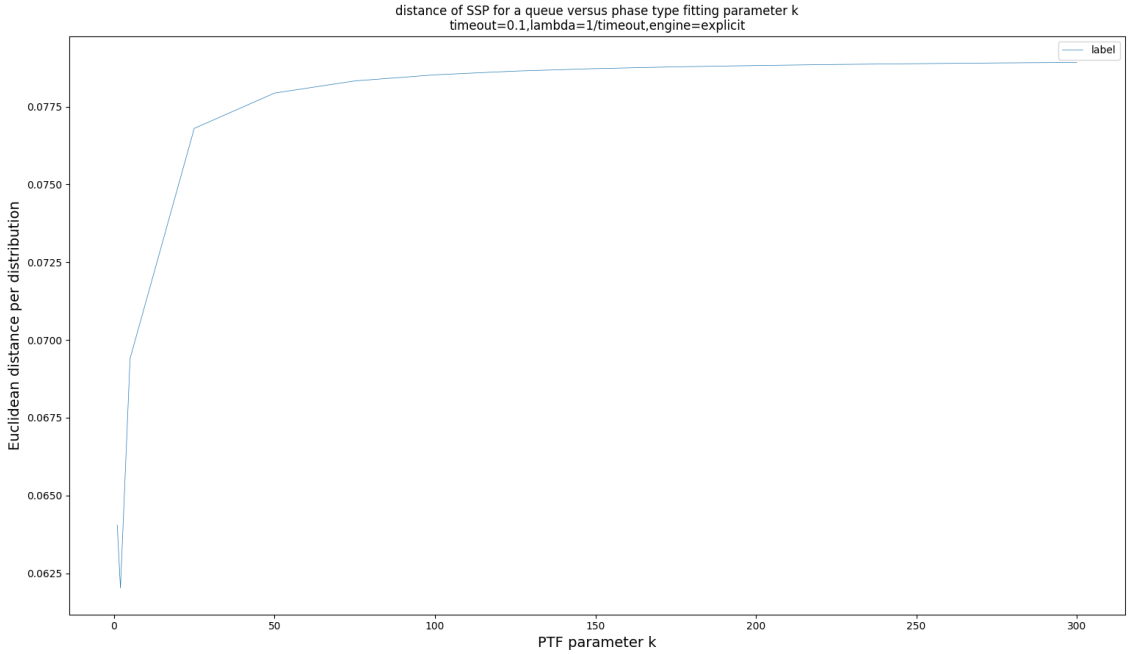

Figure 1.14: Euclidean Distance of distribution of the SSP computation between the event model and the PTF model using explicit engine versus the PTF parameter k, $t = 0.1, \lambda = 1/t, \epsilon = 10^{-5}$