Inspiration  
The problem is inspired by the following scenarios: imagine a server that has to do ‘’jobs’’ from users within a certain timeframe. However the cost depends on a certain rate; when the rate is higher, the cost is lower. Now the problem is to create a decision rule/algorithm that minimizes the cost as good as possible.

Problem formulation  
To be a bit more specific for the problem, the following assumptions are made:  
- The time of the deadline is known, and the same for every job.  
- All jobs can be finished instantly whenever it is decided to do so.  
- All jobs are independent of each other, meaning we can look at the case of doing a single job.  
- The underlying process for the rate is according to a Rayleigh Process.  
- The relation between the cost and the rate is Cost = 1/Rate.  
- (A good estimate for) the parameters of the Rayleigh process is known.  
- The amount of jobs that have to be handled is known beforehand.

Algorithm  
The developed algorithm relies on a technique called Value Iteration: In a discrete Markov process, when the cost of each state and transition probabilities are known by value iteration one can find an optimal policy to minimize the cost. To apply this we discretize the state space the rates can take (based on the percentiles of the Rayleigh distribution with our estimated parameters). Then for the first 20% of the jobs we train the algorithm, by estimating a probability matrix for our “Markov process” (the Rayleigh process is not a Markov process, but we approximate it as one). This is done by checking in each step in which state we are (if the rate is 2.1, and our closest discretized states are 2.0 and 2.3 we say we are in state 2.0), and what our previous state was.  
After this is done, we now use Value Iteration to approximate the best policy, which we then apply to the remaining 80% of the algorithm (in which time new data is used to further improve the approximation) to test the algorithm

Pseudocode

Results  
The algorithm has been tested with certain parameters and different values for T, the amount of epochs we have to make a decision when to finish the job, and the amount of states by which we discretize our state space of the Rate. To test this, we look at the ratio between the average cost of our process, and what the best possible cost could have been (which would require being psychic in practice). The results, which were obtained by simulation so likely contain some variance, can be seen in Figure 1 below. The total amount of epochs is fixed at 100,000, so if T=100 then 1,000 runs were done for a single simulation, if T=10 then 10,000 runs were done for a single simulation. Every combination of T and the number of states were simulated XXX times, and the result averaged.

Extension  
Since we approximate the Rayleigh Process to be a Markov Process, which it clearly is not, it might seem to be a good idea to at least keep track of whether the process is increasing or decreasing. This was also implemented and the results can be seen below, but since the results depend heavily on the amount of states by which the state space is discretized it seems this is not an improvement.

Further research  
A few things can be researched further to improve the algorithm:  
- The amount of states by which the state space is discretized affects the algorithm, which leads to the question how one should choose what this amount is.  
- The testing method was to compare with the absolute best (and practically impossible to implement) policy, there may be better testing methods to get a better idea of how good it performs.  
- There are many more parameters that can be tested against, which may lead to new inspirations or results.