Process algebras for quantitative analysis: Lecture 6 — Response-time analysis and safety

Stephen Gilmore

School of Informatics
The University of Edinburgh
Scotland

February 11, 2010

Overview of this talk

- This is a collaboration between LSS, UEDIN and LMU on applying the Sensoria Development Environment (SDE) to the analysis of the automotive crash rescue scenario.

Overview of this talk

- This is a collaboration between LSS, UEDIN and LMU on applying the Sensoria Development Environment (SDE) to the analysis of the automotive crash rescue scenario.
- We used FSP and the LTSA model-checker to perform the safety analysis.

Overview of this talk

- This is a collaboration between LSS, UEDIN and LMU on applying the Sensoria Development Environment (SDE) to the analysis of the automotive crash rescue scenario.
- We used FSP and the LTSA model-checker to perform the safety analysis.
- We used PEPA and the `ipclib` and `smcgraph` tools to perform response-time and sensitivity analysis.

Overview of this talk

- This is a collaboration between LSS, UEDIN and LMU on applying the Sensoria Development Environment (SDE) to the analysis of the automotive crash rescue scenario.
- We used FSP and the LTSA model-checker to perform the safety analysis.
- We used PEPA and the `ipclib` and `smcgraph` tools to perform response-time and sensitivity analysis.

Overview of this talk

- This is a collaboration between LSS, UEDIN and LMU on applying the Sensoria Development Environment (SDE) to the analysis of the automotive crash rescue scenario.
- We used FSP and the LTSA model-checker to perform the safety analysis.
- We used PEPA and the `ipclib` and `smcgraph` tools to perform response-time and sensitivity analysis.

Outcomes

Successful application of the Sensoria methodology, using process calculi cooperatively. Made connections between the FSP and PEPA calculi. Planned integration of tools using scripting.

Outline

1. Introduction
   - Service providers
   - SENSORIA Development Environment (SDE)
2. Service description
   - Service description
   - UML activity diagram
3. Safety analysis with FSP
   - Finite State Processes (FSP)
   - Analysis using LTSA
4. Response-time analysis with PEPA
   - Response-time analysis
   - Sensitivity analysis

Outline

1. Introduction
   - Service providers
   - SENSORIA Development Environment (SDE)
2. Service description
   - Service description
   - UML activity diagram
3. Safety analysis with FSP
   - Finite State Processes (FSP)
   - Analysis using LTSA
4. Response-time analysis with PEPA
   - Response-time analysis
   - Sensitivity analysis

Introduction

- Service providers who sell services which are concerned with human health and human safety have a responsibility to assess the *quality of the service* which they provide in terms of both its correctness of function and its speed of response.

Introduction

- Service providers who sell services which are concerned with human health and human safety have a responsibility to assess the *quality of the service* which they provide in terms of both its correctness of function and its speed of response.
- One way to carry out such an assessment is to construct a *precise formal model* of the service and perform the analysis on the model to shed light on the behaviour of the service itself.

Introduction

- Such an assessment exercises the ability to apply both *qualitative methods* (such as model-checking) and *quantitative methods* (such as transient analysis) in service evaluation.

Introduction

- Such an assessment exercises the ability to apply both *qualitative methods* (such as model-checking) and *quantitative methods* (such as transient analysis) in service evaluation.
- The service providers delivering these critical services may not themselves have the *technical skills* to apply methods such as these.

Introduction

- Such an assessment exercises the ability to apply both *qualitative methods* (such as model-checking) and *quantitative methods* (such as transient analysis) in service evaluation.
- The service providers delivering these critical services may not themselves have the *technical skills* to apply methods such as these.
- Even if they are able to source the necessary skills from expert users elsewhere, they may not be happy to take advantage of this because they would then *risk revealing information about their current service provision* which they might be unwilling to disclose to anyone outside their organisation.

Technology transfer

- One possible way in which the stakeholders of formal analysis methods can contribute to alleviating this problem is by *embedding their analysers in modelling environments* which lower the barrier to use of the methods.

Technology transfer

- One possible way in which the stakeholders of formal analysis methods can contribute to alleviating this problem is by *embedding their analysers in modelling environments* which lower the barrier to use of the methods.
- These environments can then be adopted and applied by the service providers in-house, allowing them to *evaluate their service provision without revealing sensitive information* about their current level of service.

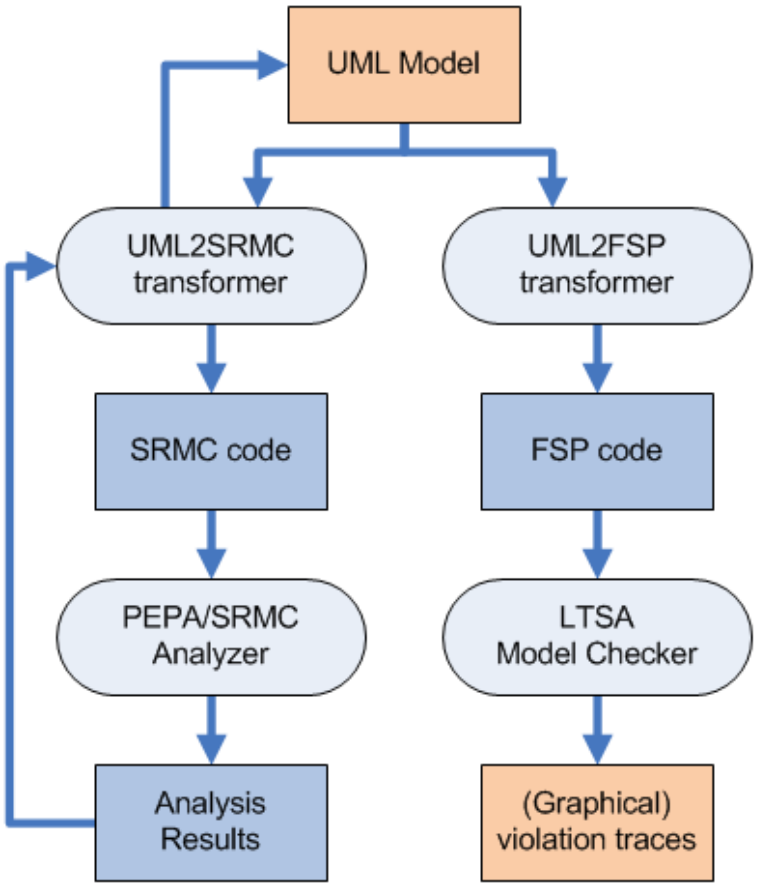
SENSORIA Development Environment (SDE)

- The **SENSORIA Development Environment (SDE)** assists us in the goal of bringing state-of-the-art analysis methods closer to the service providers who need to apply them.

SENSORIA Development Environment (SDE)

- The **SENSORIA Development Environment (SDE)** assists us in the goal of bringing state-of-the-art analysis methods closer to the service providers who need to apply them.
- We use the SDE and other tools to assess an accident assistance service against both safety properties (using model-checking over labelled transition systems) and response-time properties (using transient analysis of continuous-time Markov chains).

SENSORIA Development Environment (SDE)



Outline

1. Introduction
   - Service providers
   - SENSORIA Development Environment (SDE)
2. Service description
   - Service description
   - UML activity diagram
3. Safety analysis with FSP
   - Finite State Processes (FSP)
   - Analysis using LTSA
4. Response-time analysis with PEPA
   - Response-time analysis
   - Sensitivity analysis

Service description

- We are considering a subscription service which uses the on-board diagnostic and communication systems in high-end cars to provide an accident assistance service.

Service description

- We are considering a subscription service which uses the on-board diagnostic and communication systems in high-end cars to provide an accident assistance service.
- The service is triggered by any impact or collision which causes the car airbag to deploy. Immediately after the airbag has deployed the on-board communication module transmits to the assistance service a report with as much information as it can obtain from the car's diagnostic system.

Accident report

- This report includes information about the state of the car itself obtained from sensors in the engine and the braking system. The report also specifies the speed of the car at the moment of impact and, most importantly, the geographical location of the car as obtained from the on-board GPS.

Accident report

- This report includes information about the state of the car itself obtained from sensors in the engine and the braking system. The report also specifies the speed of the car at the moment of impact and, most importantly, the geographical location of the car as obtained from the on-board GPS.

Accident report

- This report includes information about the state of the car itself obtained from sensors in the engine and the braking system. The report also specifies the speed of the car at the moment of impact and, most importantly, the geographical location of the car as obtained from the on-board GPS.

Example

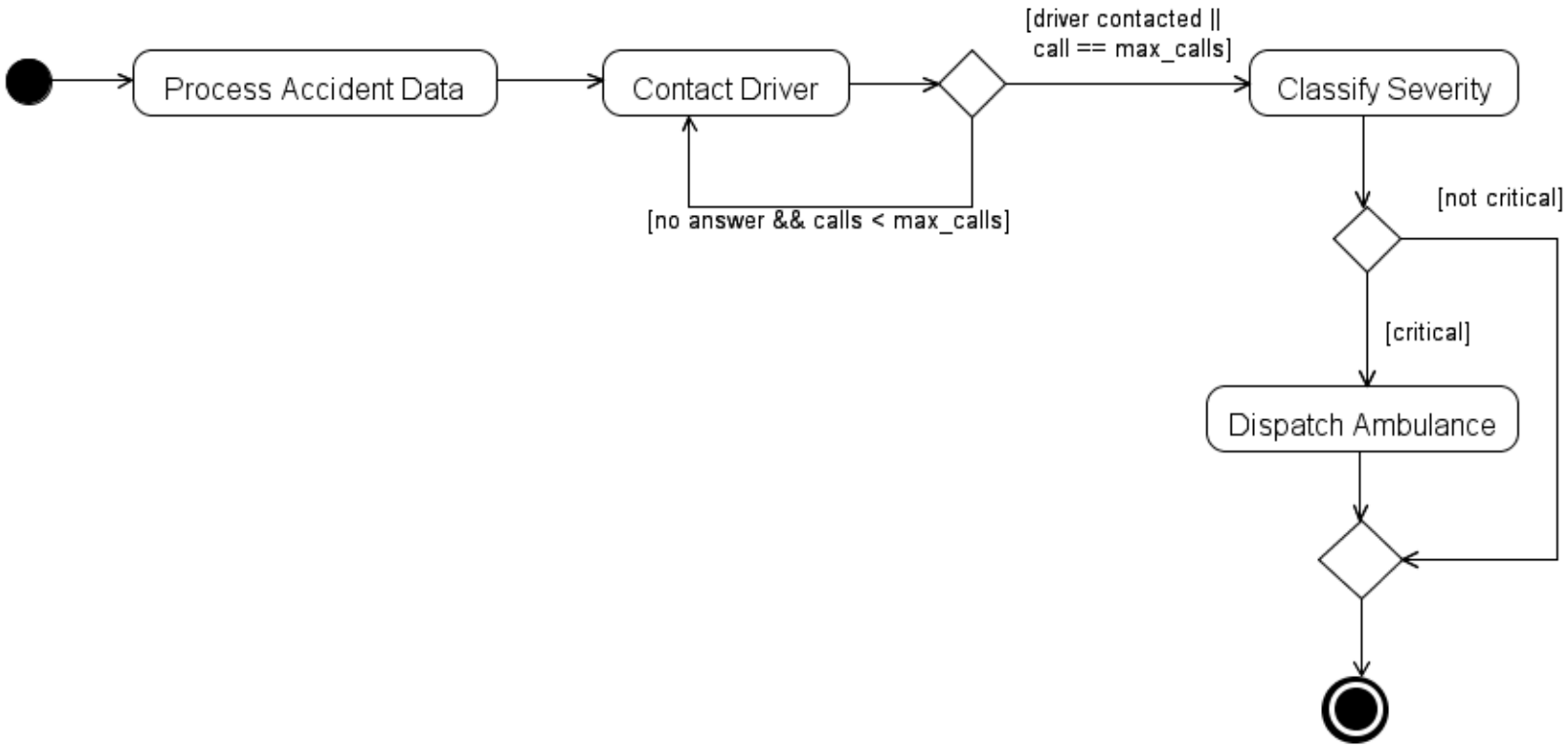
OnStar service from General Motors (`http://www.onstar.com/`)

OnStar in action

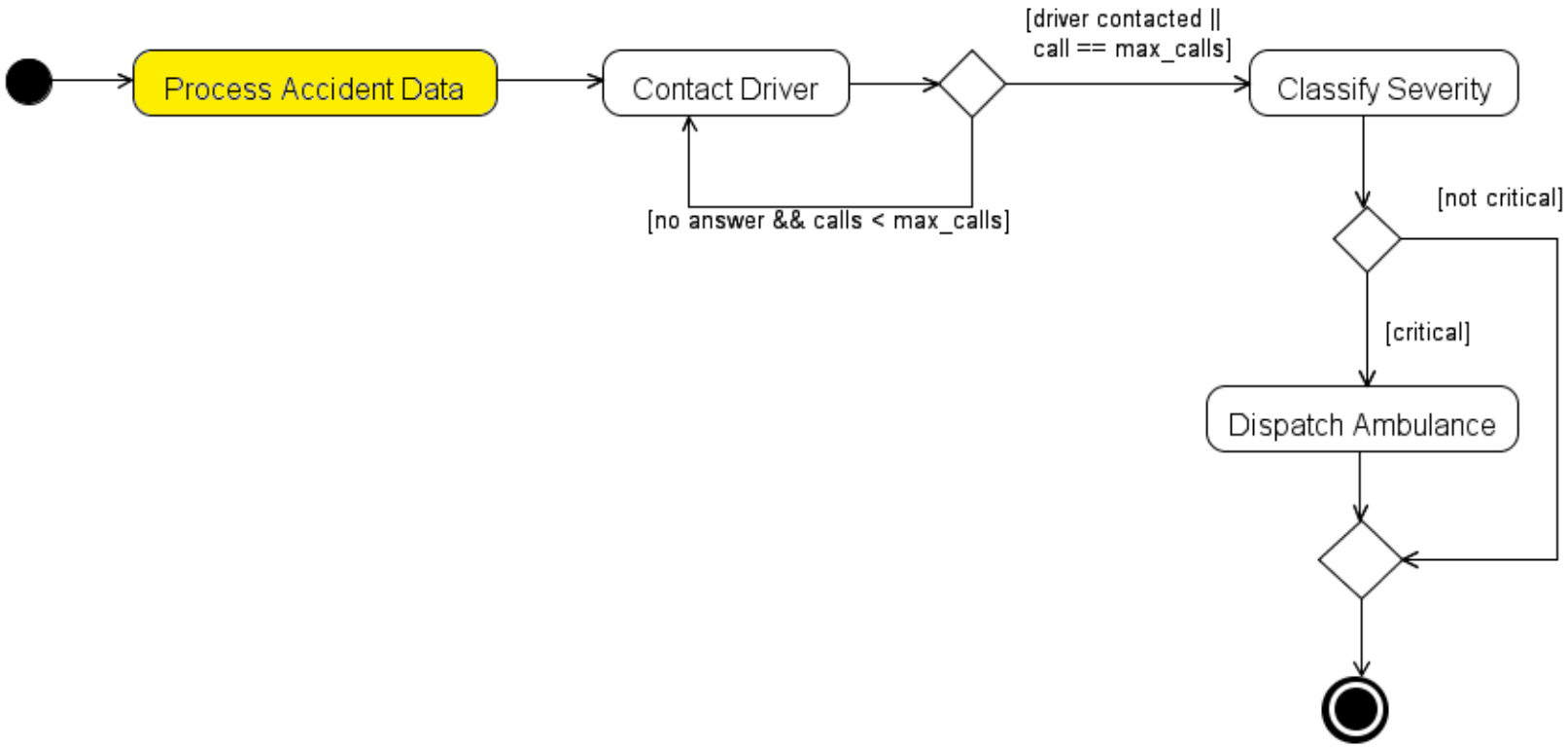
The actor Charlie Sheen received a call from OnStar on the morning of February 5th, 2010 after the airbags deployed on one of his cars. His Mercedes Benz was found in a canyon in Beverley Hills after plunging off of Mulholland Drive.

Charlie Sheen's car, February 5, 2010

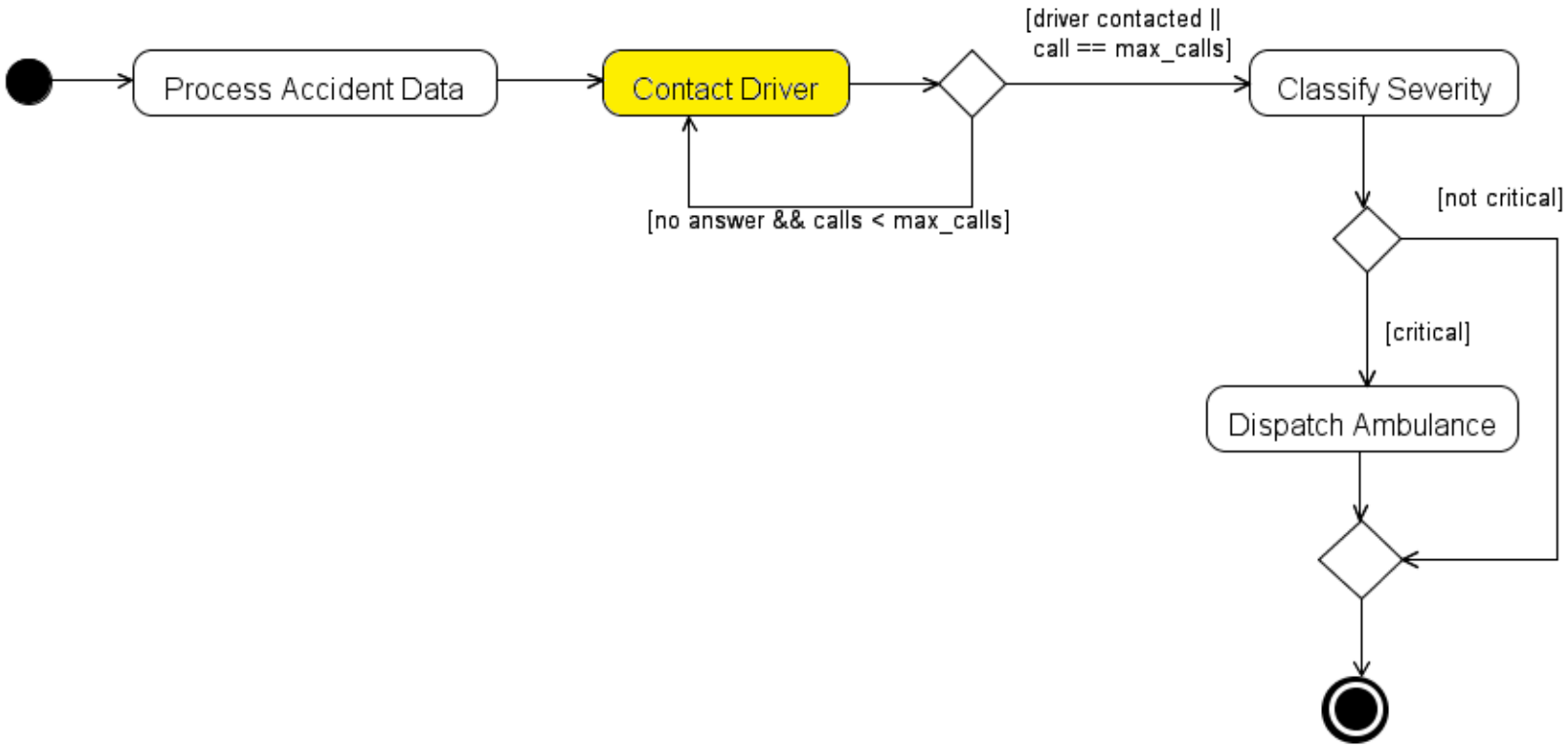
UML activity diagram of the Accident Assistance Service



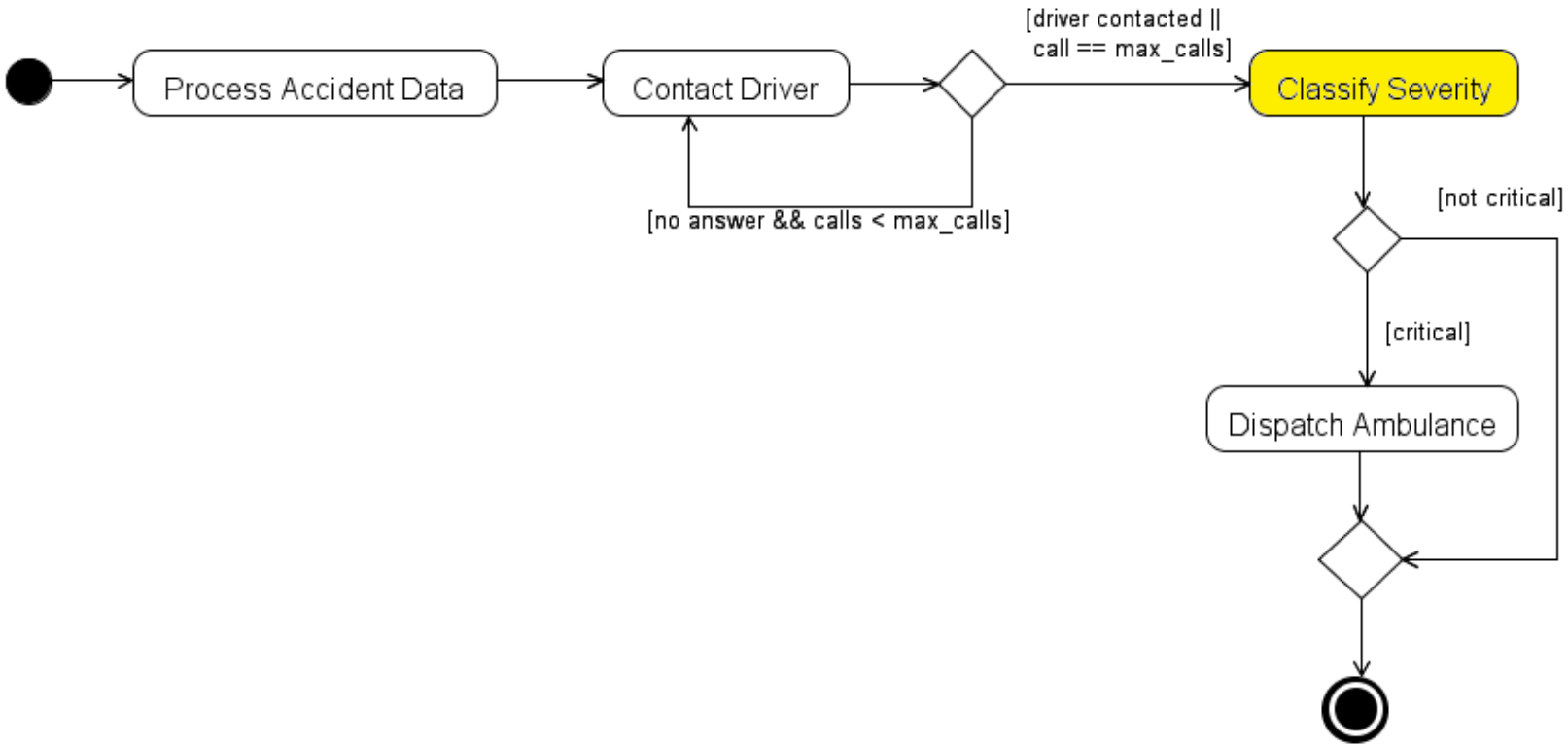
UML activity diagram of the Accident Assistance Service



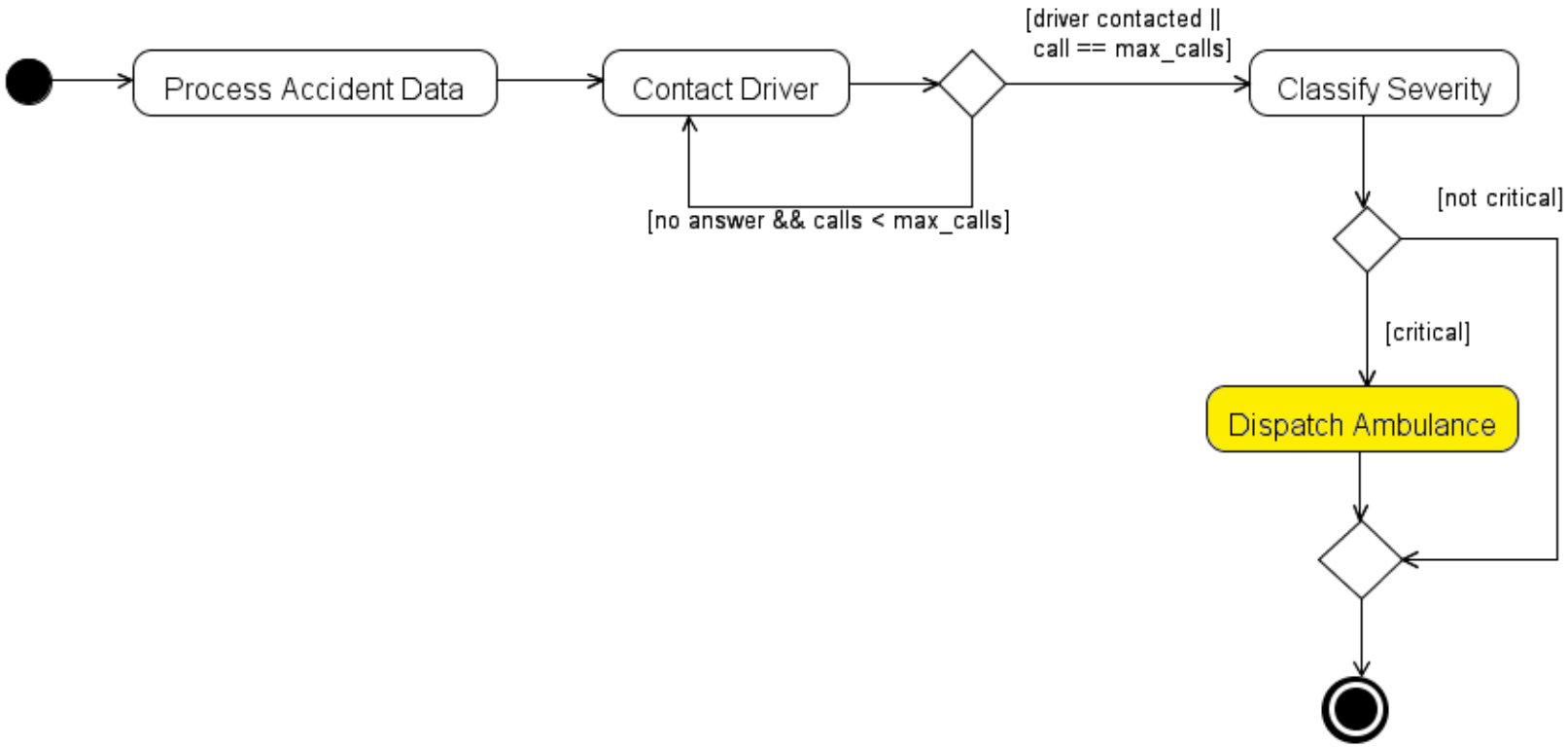
UML activity diagram of the Accident Assistance Service



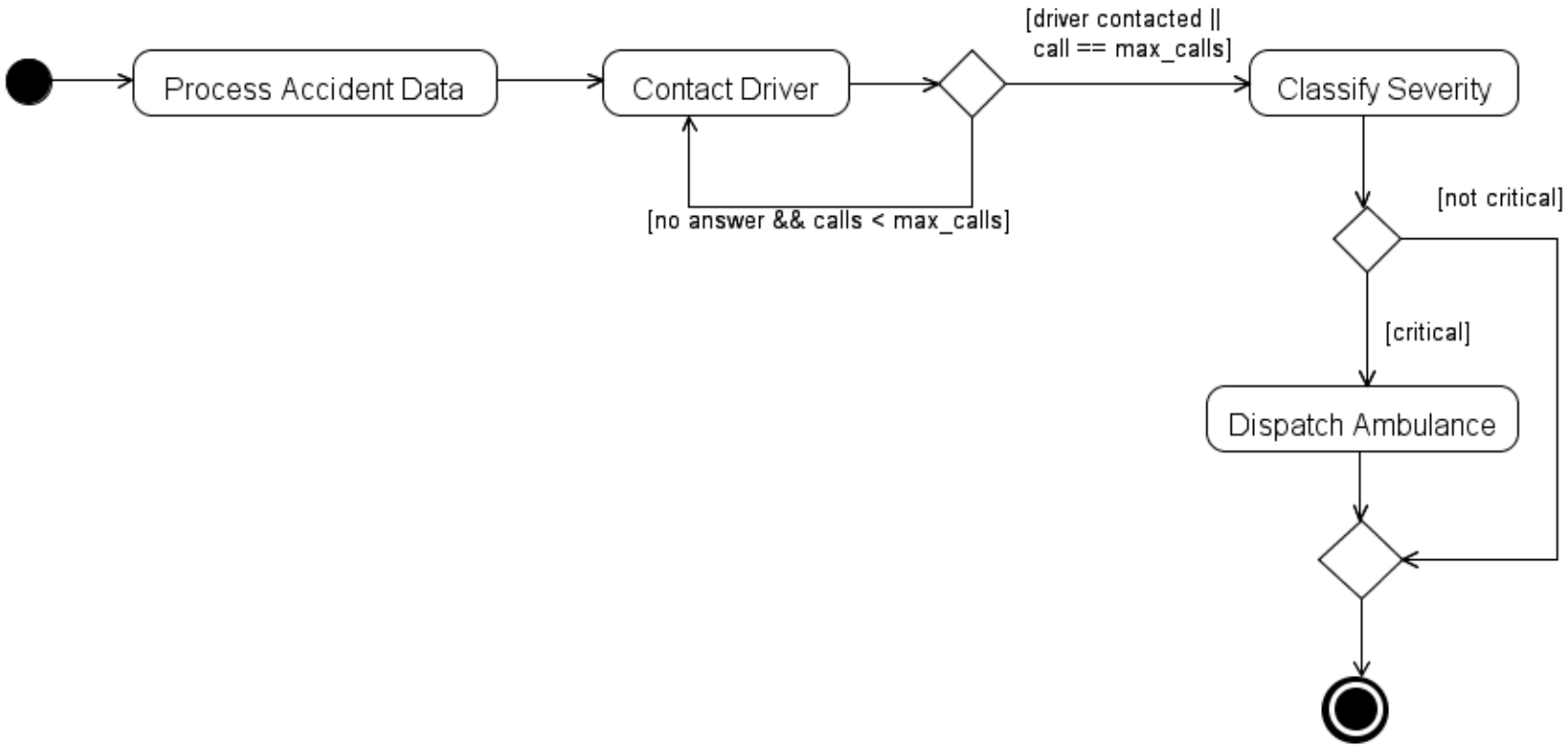
UML activity diagram of the Accident Assistance Service



UML activity diagram of the Accident Assistance Service



UML activity diagram of the Accident Assistance Service



An important case: No answer

- An important case to consider occurs when the service cannot get confirmation from the driver that they do not need assistance.

An important case: No answer

- An important case to consider occurs when the service cannot get confirmation from the driver that they do not need assistance.
- It might seem that the obvious course of action should be to consider not getting an answer to be a critical case but there is evidently a possibility that the service will send an ambulance when it is not needed.

An important case: No answer

- An important case to consider occurs when the service cannot get confirmation from the driver that they do not need assistance.
- It might seem that the obvious course of action should be to consider not getting an answer to be a critical case but there is evidently a possibility that the service will send an ambulance when it is not needed.
- That is, the driver is unhurt but did not have their mobile telephone with them, or it had no battery charge, or they had no signal from their telephone service provider, or many other similar reasons.

An important case: No answer

- Because critical services should not be deployed without good reason, the accident assistance service would like to reduce the number of occasions when an ambulance is dispatched in error. The *information on the car status* and the speed of the car at the moment of impact sent with the accident report become significant when we have no answer from the driver.

An important case: No answer

- Because critical services should not be deployed without good reason, the accident assistance service would like to reduce the number of occasions when an ambulance is dispatched in error. The *information on the car status* and the speed of the car at the moment of impact sent with the accident report become significant when we have no answer from the driver.
- In the case of no answer and car diagnostics which point to very little damage (say, the car was stationary at the time of impact, and the engine, brakes, lights and other critical functions seem to be functioning normally) then the service will decide *not to send an ambulance* to prevent sending one when it could be needed elsewhere.

What if the airbag has deployed?

- Shouldn't we *always* send the ambulance if the airbag has deployed?

What if the airbag has deployed?

- Shouldn't we *always* send the ambulance if the airbag has deployed?

What if the airbag has deployed?

- Shouldn't we *always* send the ambulance if the airbag has deployed?

Example

No, not necessarily.
(`http://www.youtube.com/watch?v=6bBVJAFrNOU`)

Scope of the modelling exercise

- Models are created with a specific purpose in mind.

Scope of the modelling exercise

- Models are created with a specific purpose in mind.
- Our model of the Accident Assistance Service details the events which are the area of responsibility of the service itself. That is, those activities which occur between an accident report being received and the service discharging its responsibility to act on the accident report.

Scope of the modelling exercise

- Models are created with a specific purpose in mind.
- Our model of the Accident Assistance Service details the events which are the area of responsibility of the service itself. That is, those activities which occur between an accident report being received and the service discharging its responsibility to act on the accident report.
- In some cases this will lead to an ambulance being sent, and in other cases not.

Scope of the modelling exercise

- Our model does not require us to know – or allow us to predict – anything about activities which happen before or after these events.

Scope of the modelling exercise

- Our model does not require us to know – or allow us to predict – anything about activities which happen before or after these events.
- For example, we do not estimate how often accidents occur and we do not calculate how long ambulances take to arrive. Both of these may be interesting to know, but our model here does not speak of them.

Outline

1. Introduction
   - Service providers
   - SENSORIA Development Environment (SDE)
2. Service description
   - Service description
   - UML activity diagram
3. Safety analysis with FSP
   - Finite State Processes (FSP)
   - Analysis using LTSA
4. Response-time analysis with PEPA
   - Response-time analysis
   - Sensitivity analysis

Safety analysis with FSP

- For the purpose of our analysis, we translate the service process workflow into the **Finite State Processes (FSP)** notation to concisely and formally model the workflow states and transitions.

Finite State Processes (FSP)

Action prefix	`(x->P)` describes a process which first engages in the action `x` and then behaves as described by the auxiliary process `P`;
Choice	`(x->P|y->Q)` describes a process which initially engages in either `x` or `y`, and then becomes `P` or `Q`, respectively;
Recursion	The behaviour of a process may be defined in terms of itself, in order to express repetition;
Sequential composition	`(P;Q)` behaves as `P` and when it reaches the `END` state of `P` behaves as `Q`;
Parallel composition	`(P||Q)` describes the parallel composition of processes `P` and `Q`.

FSP model

ProcessAccidentData FSP Process Composition

```
VEHDIAGCHOICE = (
    vehicle.emergsrv.diags_normal->
        emergsrv.diag.write[normal]->END |
    vehicle.emergsrv.diags_critical->
        emergsrv.diag.write[critical]->END).
VEHDIAGSEQ = VEHDIAGCHOICE; END.
||PROCESSACCIDENTDATA = (VEHDIAGSEQ).
```

FSP model

Call Attempts Composition

```
// no of calls before automatic dispatch
const Max = 3

CALLATTEMPT(N=0) = CALL[N],
CALL[v:Int] = (emergsrv.driver.callphone->ANSWER[v]),
ANSWER[v:Int] = (driver.emergsrv.noanswer->CALL[v+1]|
   driver.emergsrv.answer->ANSWEREDACTION),
ANSWEREDACTION = (emergsrv.phone.write[normal]->END|
   emergsrv.phone.write[critical]->END),
CALL[Max] = (emergsrv.phone.write[critical]->END).
```

FSP model

Call Attempts Composition

```
// no of calls before automatic dispatch
const Max = 3

CALLATTEMPT(N=0) = CALL[N],
CALL[v:Int] = (emergsrv.driver.callphone->ANSWER[v]),
ANSWER[v:Int] = (driver.emergsrv.noanswer->CALL[v+1]|
   driver.emergsrv.answer->ANSWEREDACTION),
ANSWEREDACTION = (emergsrv.phone.write[normal]->END|
   emergsrv.phone.write[critical]->END),
CALL[Max] = (emergsrv.phone.write[critical]->END).
```

FSP model

Call Attempts Composition

```
// no of calls before automatic dispatch
const Max = 3

CALLATTEMPT(N=0) = CALL[N],
CALL[v:Int] = (emergsrv.driver.callphone->ANSWER[v]),
ANSWER[v:Int] = (driver.emergsrv.noanswer->CALL[v+1]|
   driver.emergsrv.answer->ANSWEREDACTION),
ANSWEREDACTION = (emergsrv.phone.write[normal]->END|
   emergsrv.phone.write[critical]->END),
CALL[Max] = (emergsrv.phone.write[critical]->END).
```

FSP model

Call Attempts Composition

```
// no of calls before automatic dispatch
const Max = 3

CALLATTEMPT(N=0) = CALL[N],
CALL[v:Int] = (emergsrv.driver.callphone->ANSWER[v]),
ANSWER[v:Int] = (driver.emergsrv.noanswer->CALL[v+1]|
   driver.emergsrv.answer->ANSWEREDACTION),
ANSWEREDACTION = (emergsrv.phone.write[normal]->END|
   emergsrv.phone.write[critical]->END),
CALL[Max] = (emergsrv.phone.write[critical]->END).
```

FSP model

Call Attempts Composition

```
// no of calls before automatic dispatch
const Max = 3

CALLATTEMPT(N=0) = CALL[N],
CALL[v:Int] = (emergsrv.driver.callphone->ANSWER[v]),
ANSWER[v:Int] = (driver.emergsrv.noanswer->CALL[v+1]|
   driver.emergsrv.answer->ANSWEREDACTION),
ANSWEREDACTION = (emergsrv.phone.write[normal]->END|
   emergsrv.phone.write[critical]->END),
CALL[Max] = (emergsrv.phone.write[critical]->END).
```

FSP model

Call Attempts Composition

```
// no of calls before automatic dispatch
const Max = 3

CALLATTEMPT(N=0) = CALL[N],
CALL[v:Int] = (emergsrv.driver.callphone->ANSWER[v]),
ANSWER[v:Int] = (driver.emergsrv.noanswer->CALL[v+1]|
   driver.emergsrv.answer->ANSWEREDACTION),
ANSWEREDACTION = (emergsrv.phone.write[normal]->END|
   emergsrv.phone.write[critical]->END),
CALL[Max] = (emergsrv.phone.write[critical]->END).
```

FSP model

Check diagnostic information received

```
QUERYDIAGSTATUS = (emergsrv.diag.read[i:Int]->
    QUERYDIAGSTATUS[i]),
QUERYDIAGSTATUS[i:Int] = if (i==critical)
    then DISPATCH; END else END.
CLASSIFYSEQ = QUERYDIAGSTATUS; END.
||CLASSIFYSEVERITY = (CLASSIFYSEQ).
```

FSP model

Dispatch Ambulance

```
SENDAMBULANCE =
    (emergsrv.station.send_ambulance->END).
||DISPATCH = (SENDAMBULANCE).
```

Assistance Log (Final Action)

```
LOG = (emergsrv.log.result->END).
||LOGREPORT = (LOG).
```

Service Main sequence

```
MAINSEQ = PROCESSVEHICLEDATA; CONTACTDRIVER;
    CLASSIFYSEVERITY; LOG; END.
```

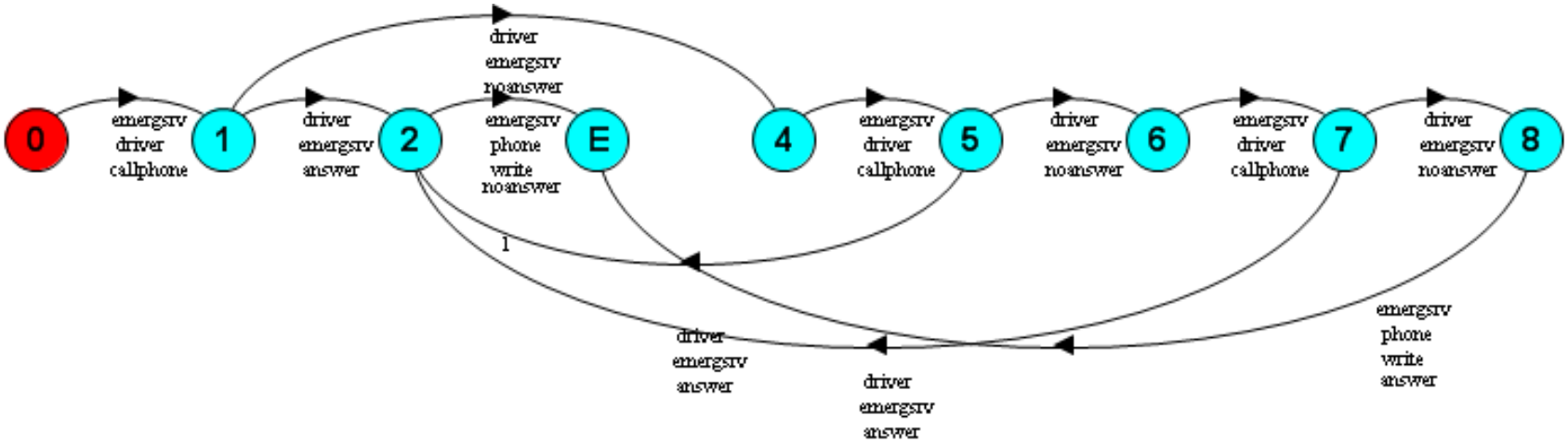
Labelled Transition System Analyzer (LTSA)

- The constructed FSP can be used to model the exact transition of workflow processes through a modelling tool such as the **Labelled Transition System Analyzer (LTSA)**, which provides a compilation of an FSP into a state machine and provides a resulting Labelled Transition System (LTS).

Labelled Transition System Analyzer (LTSA)

- The constructed FSP can be used to model the exact transition of workflow processes through a modelling tool such as the **Labelled Transition System Analyzer (LTSA)**, which provides a compilation of an FSP into a state machine and provides a resulting Labelled Transition System (LTS).
- A default deadlock check of the service process results in no violations being found (i.e. that there are no deadlock states in the model).

Labelled transition system for the driver call process



Property checking with LTSA

- The LTSA tool allows us to check logical properties against our FSP model.

Property checking with LTSA

- The LTSA tool allows us to check logical properties against our FSP model.

Property checking with LTSA

- The LTSA tool allows us to check logical properties against our FSP model.

Model-checking uncovered a flaw in an earlier version of the model which (erroneously) omitted the check on the severity reported by the driver in the case that they answer the phone.

Property checking with LTSA

- The LTSA tool allows us to check logical properties against our FSP model.

Model-checking uncovered a flaw in an earlier version of the model which (erroneously) omitted the check on the severity reported by the driver in the case that they answer the phone.

- This led to the possibility of an ambulance being sent in error.

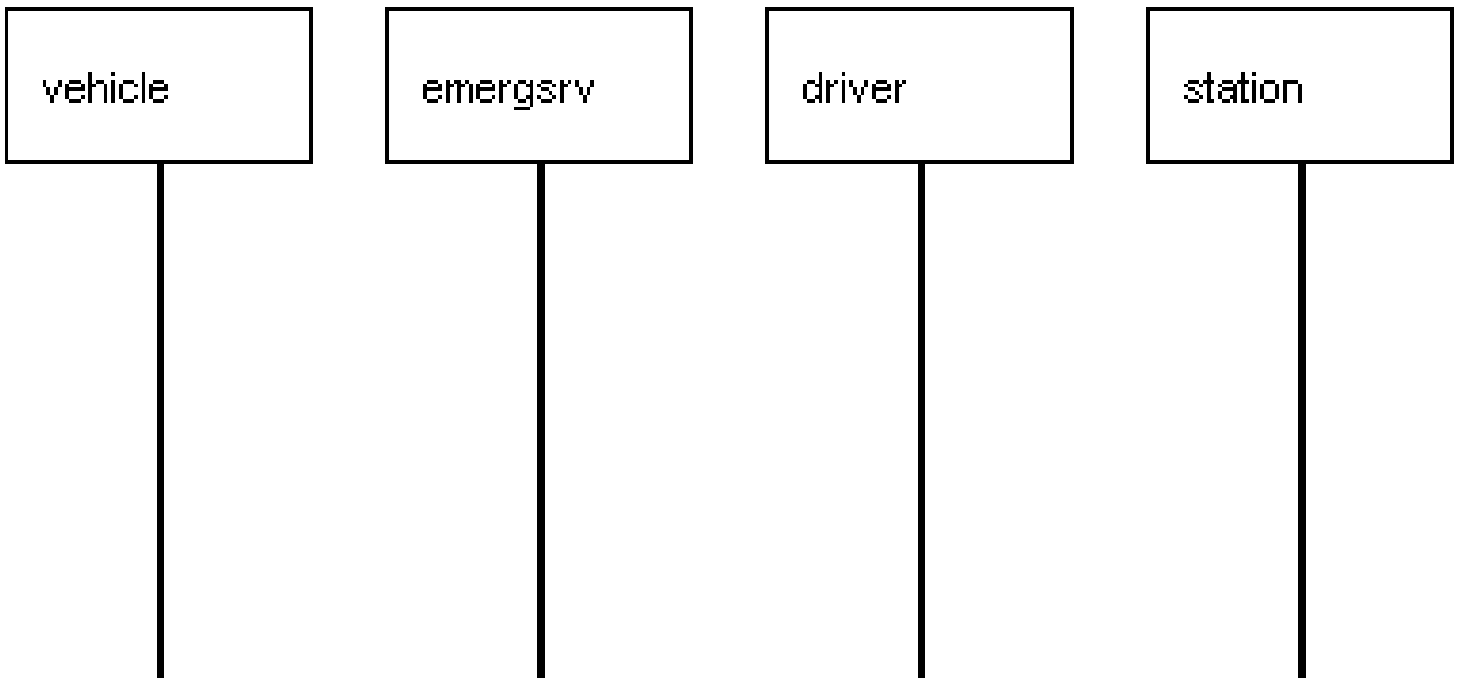
Property checking with LTSA

- The LTSA tool allows us to check logical properties against our FSP model.

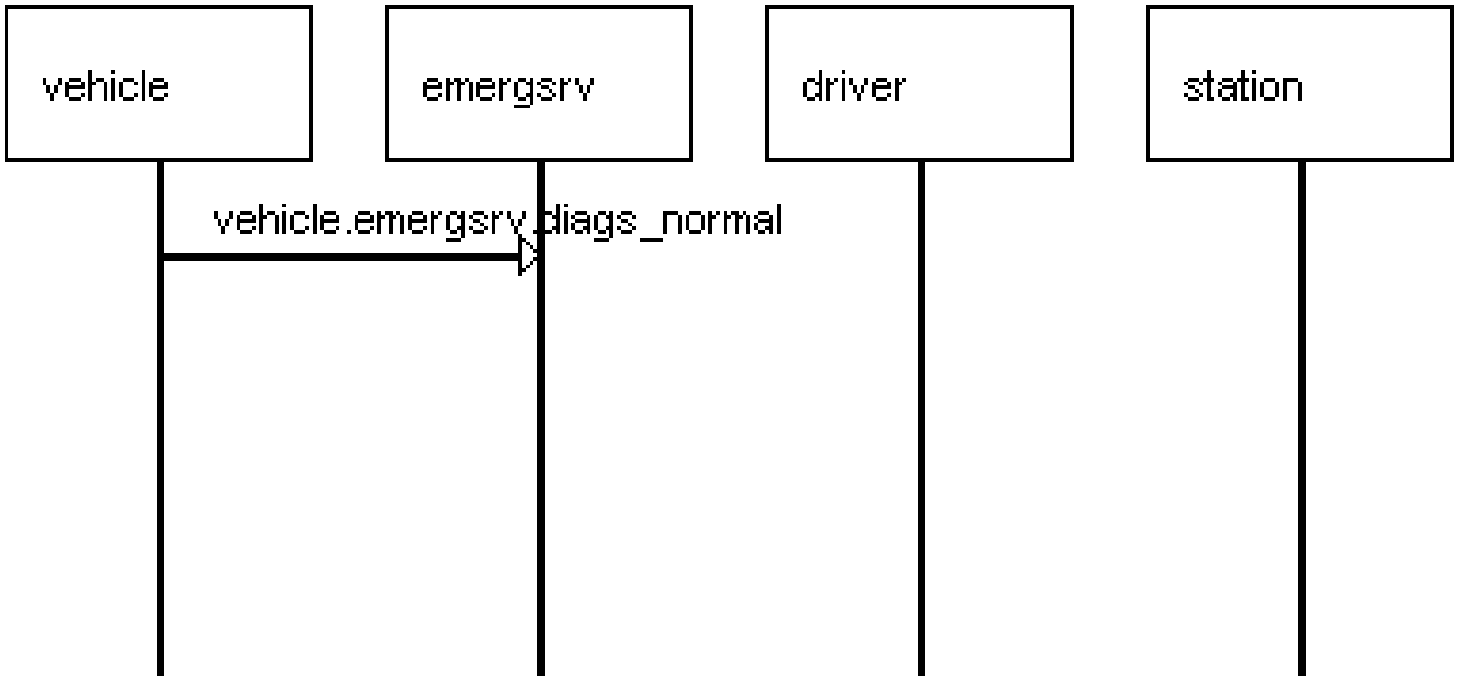
Model-checking uncovered a flaw in an earlier version of the model which (erroneously) omitted the check on the severity reported by the driver in the case that they answer the phone.

- This led to the possibility of an ambulance being sent in error.
- The violation of the property is reported by LTSA in the form of a message sequence chart.

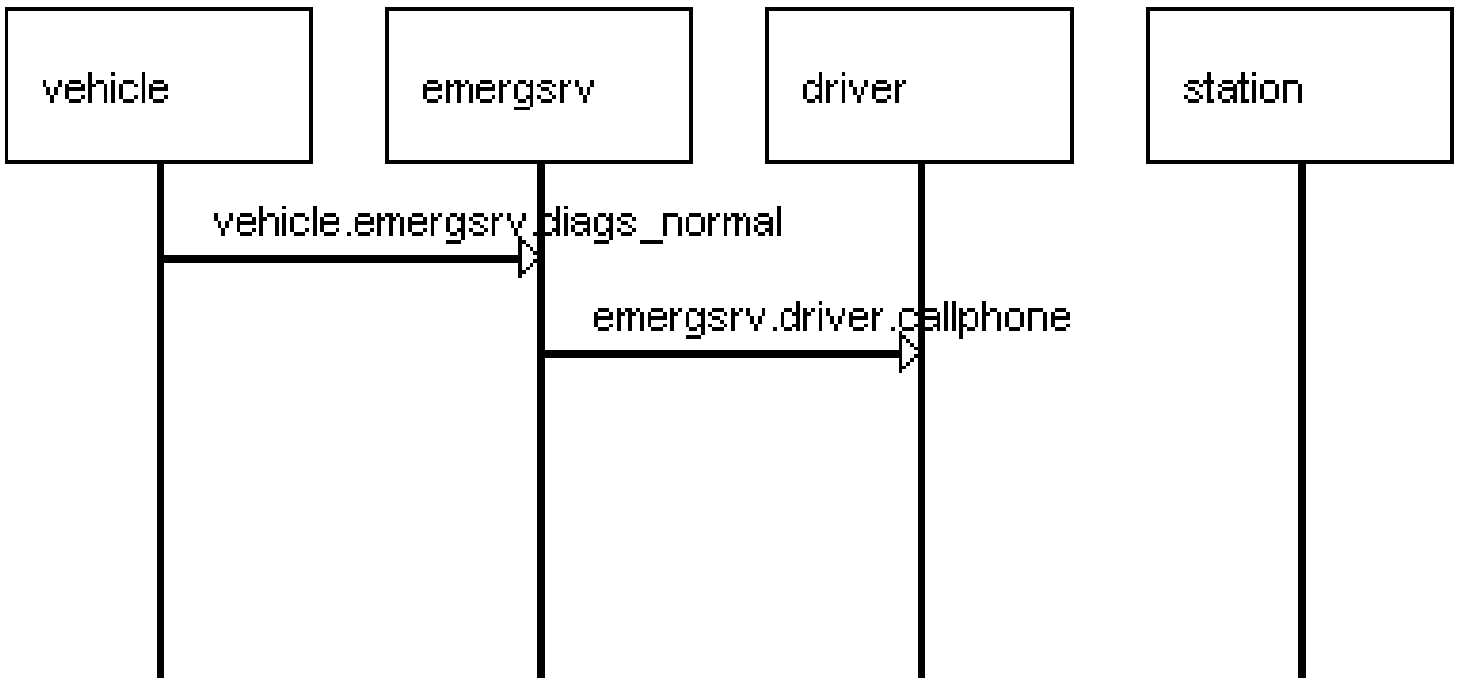
LTSA output of trace leading to violation



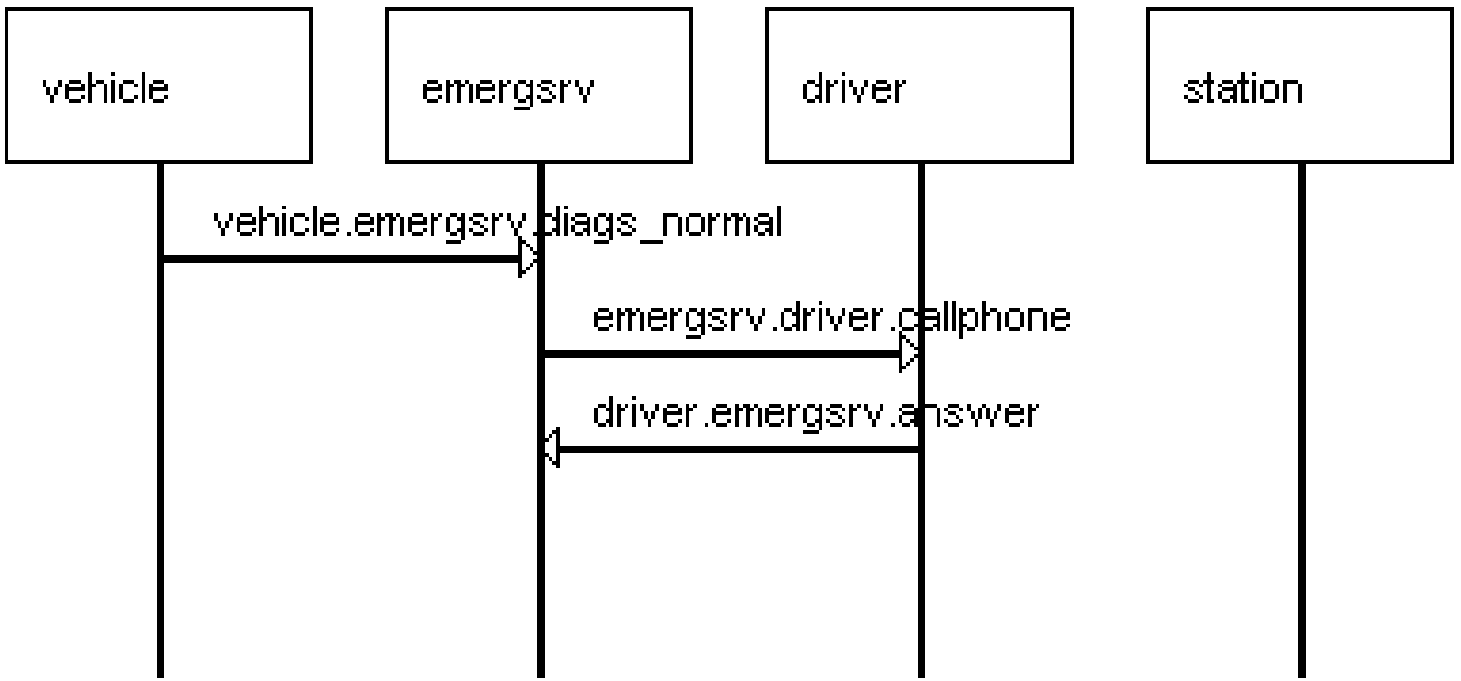
LTSA output of trace leading to violation



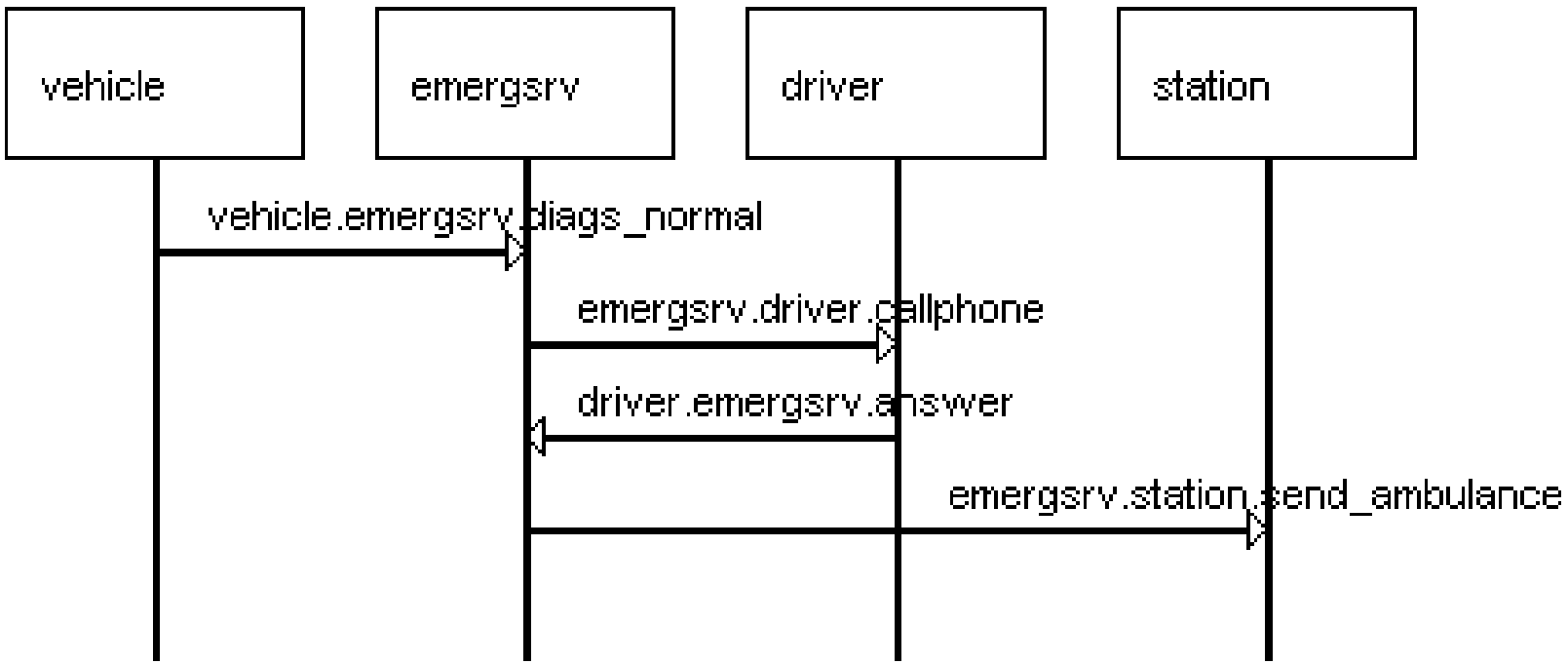
LTSA output of trace leading to violation



LTSA output of trace leading to violation



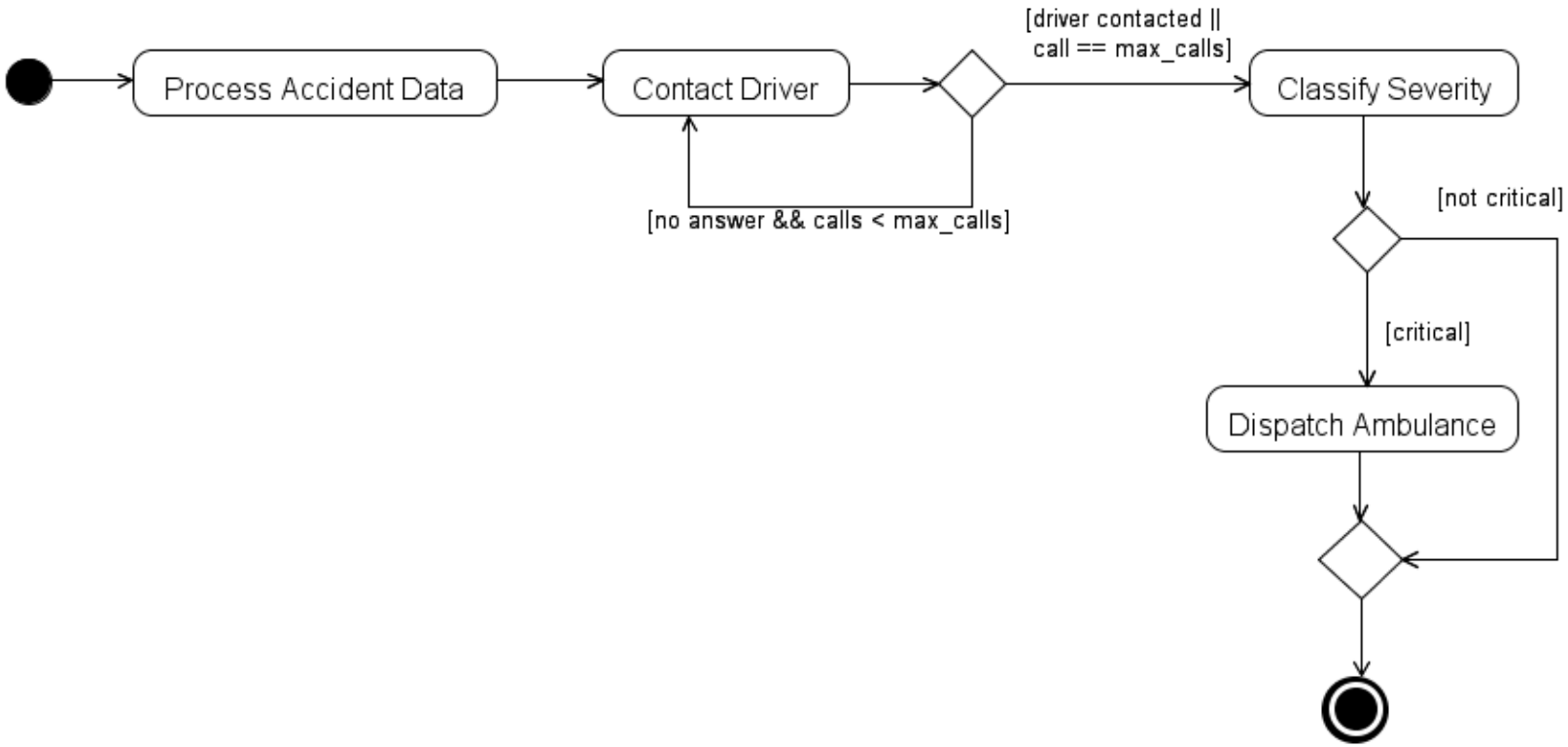
LTSA output of trace leading to violation



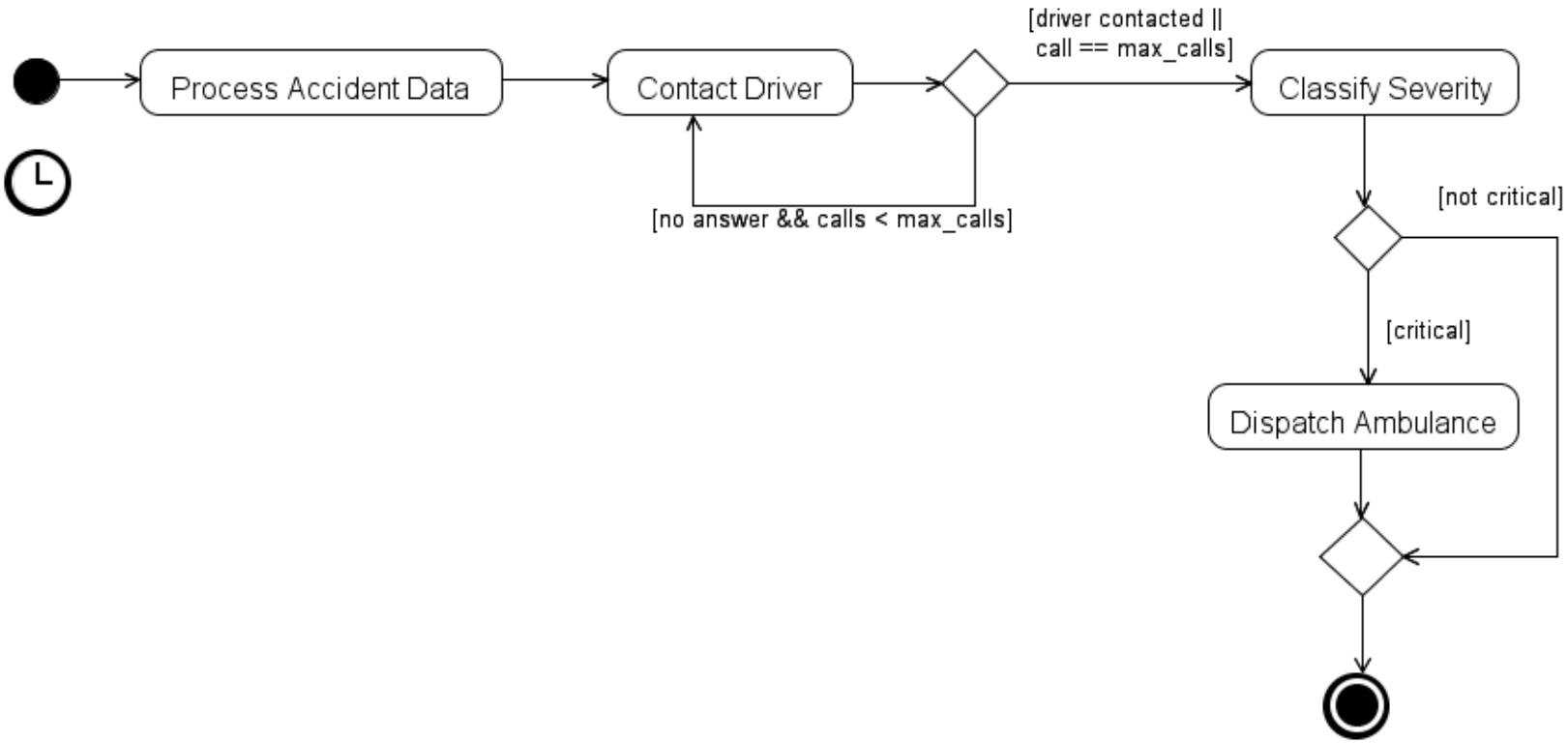
Outline

1. Introduction
   - Service providers
   - SENSORIA Development Environment (SDE)
2. Service description
   - Service description
   - UML activity diagram
3. Safety analysis with FSP
   - Finite State Processes (FSP)
   - Analysis using LTSA
4. Response-time analysis with PEPA
   - Response-time analysis
   - Sensitivity analysis

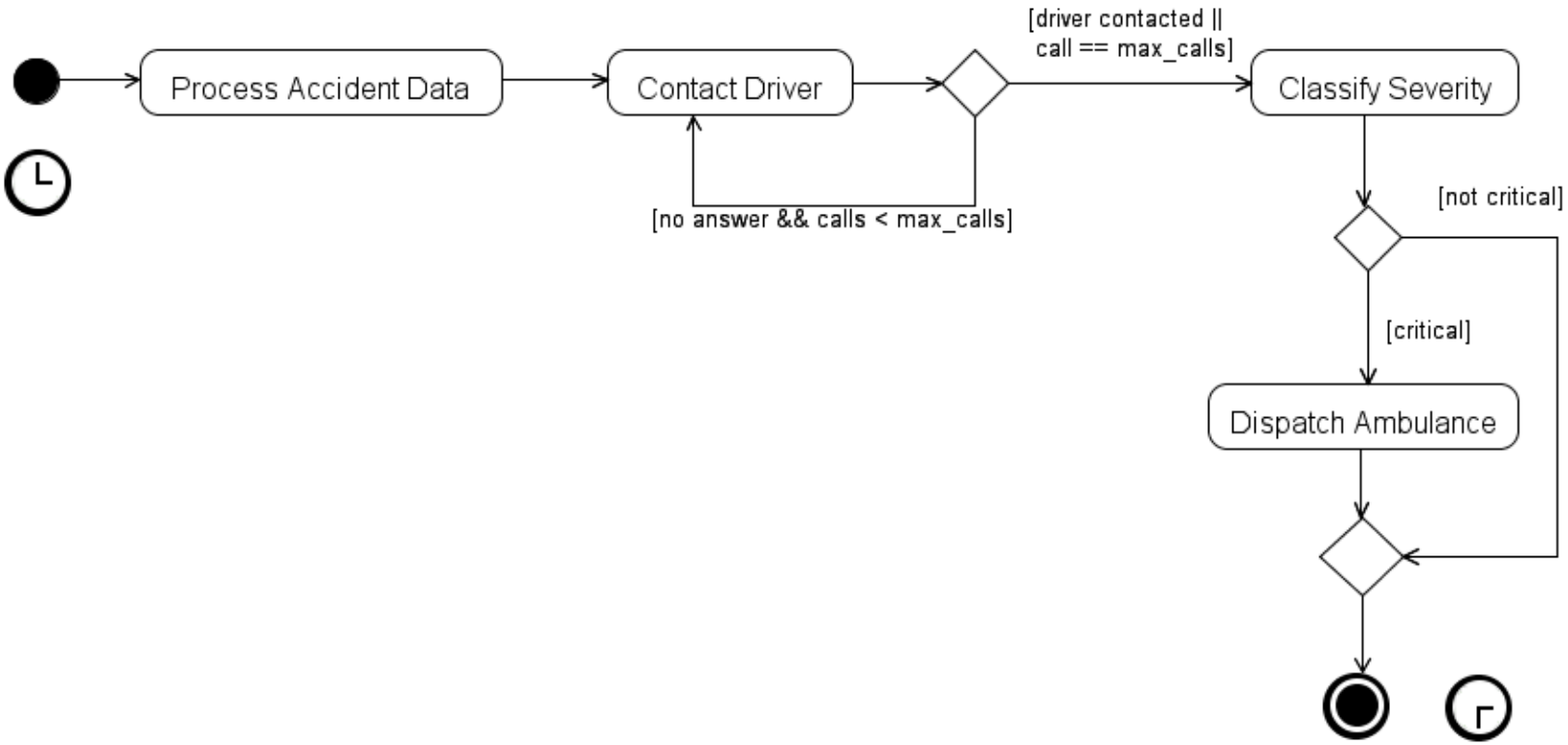
Response-time analysis of the Accident Assistance Service



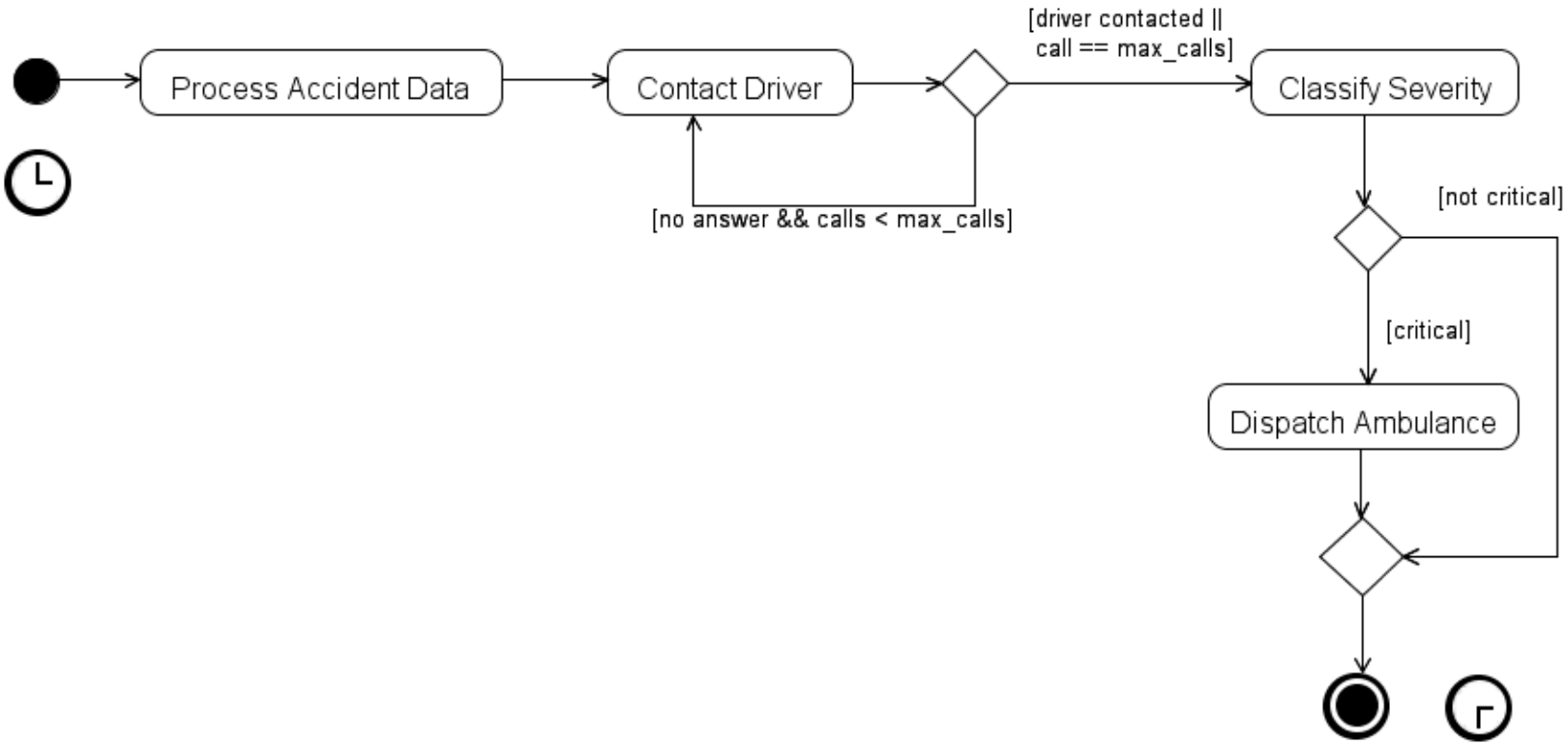
Response-time analysis of the Accident Assistance Service



Response-time analysis of the Accident Assistance Service

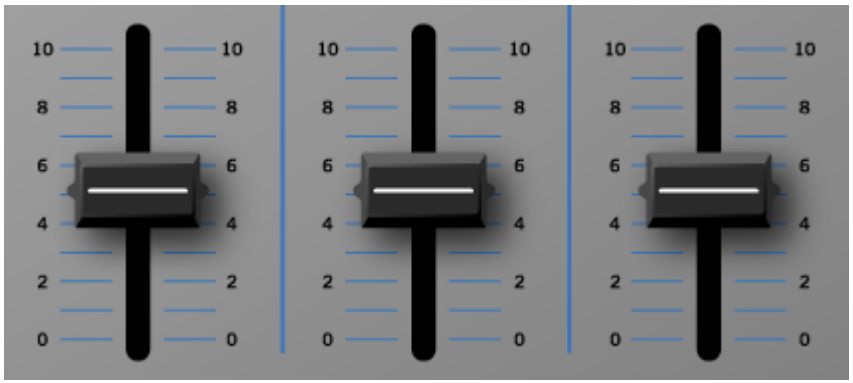


Response-time analysis of the Accident Assistance Service



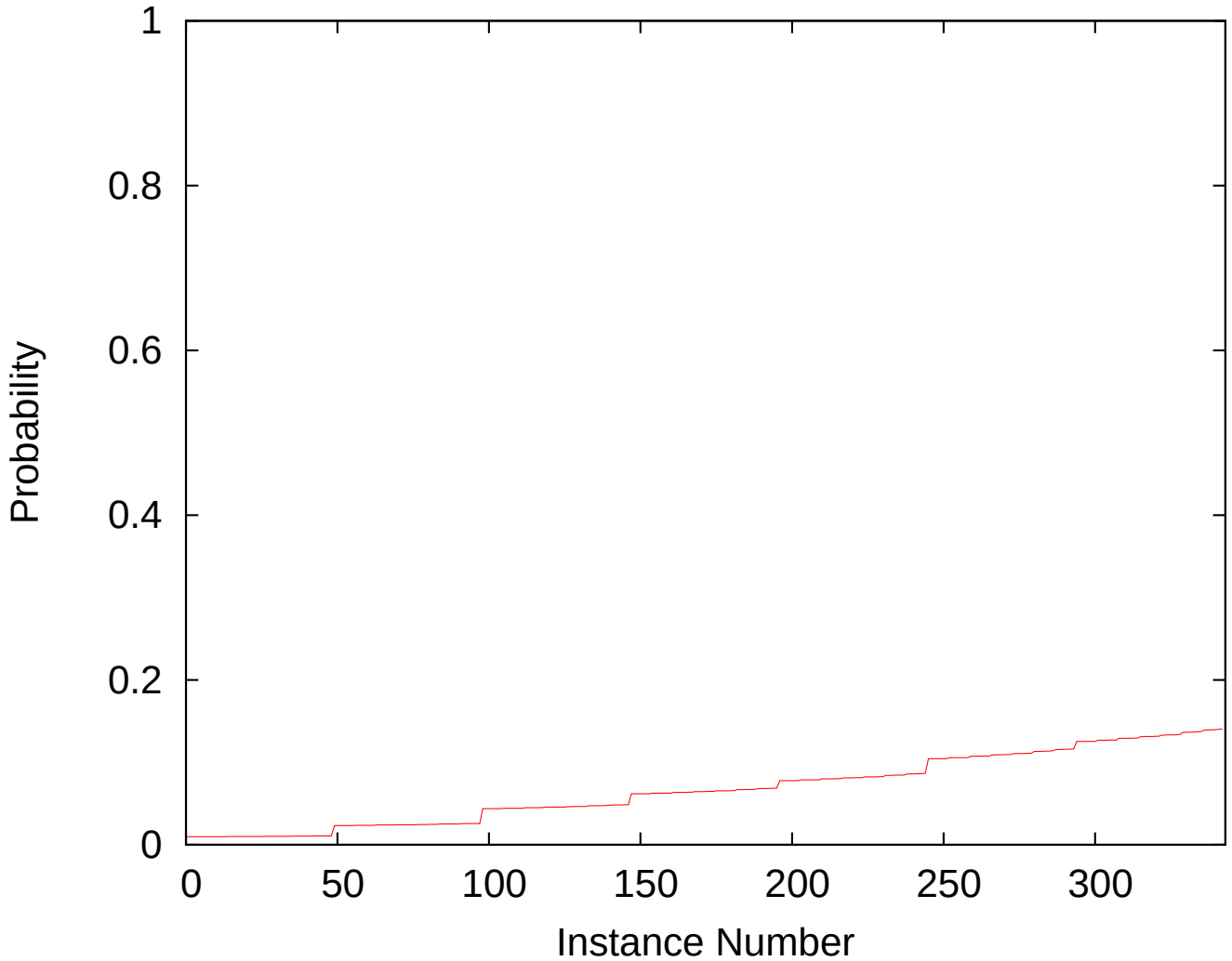
Sensitivity analysis

We vary the rates at which the three calls to the driver are performed. In the PEPA model these are rates *r_wait_answer_1*, *r_wait_answer_2* and *r_wait_answer_3*.

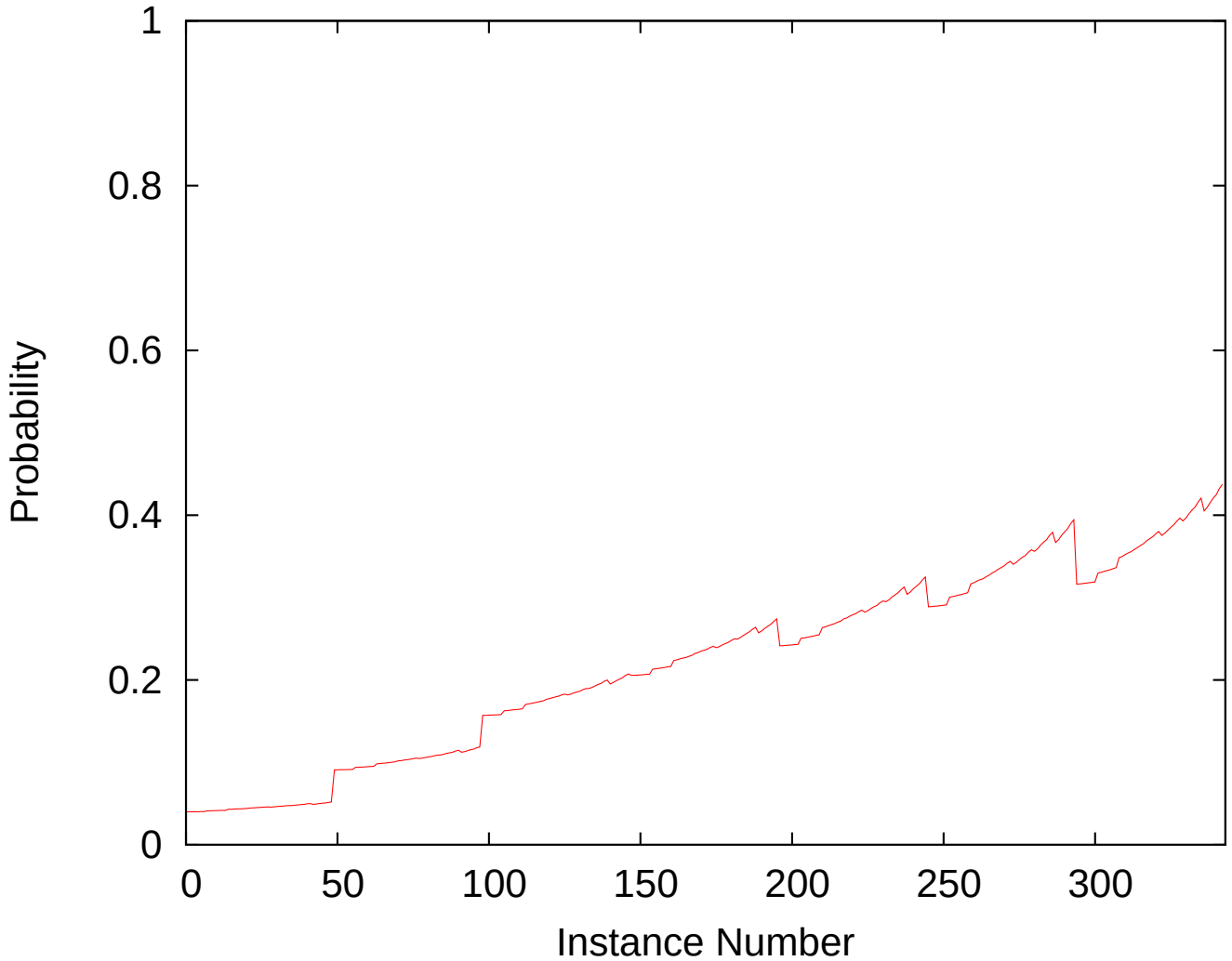


These three variables are varied across the seven values $\{0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2\}$. We do $7 \times 7 \times 7 = 343$ experiments.

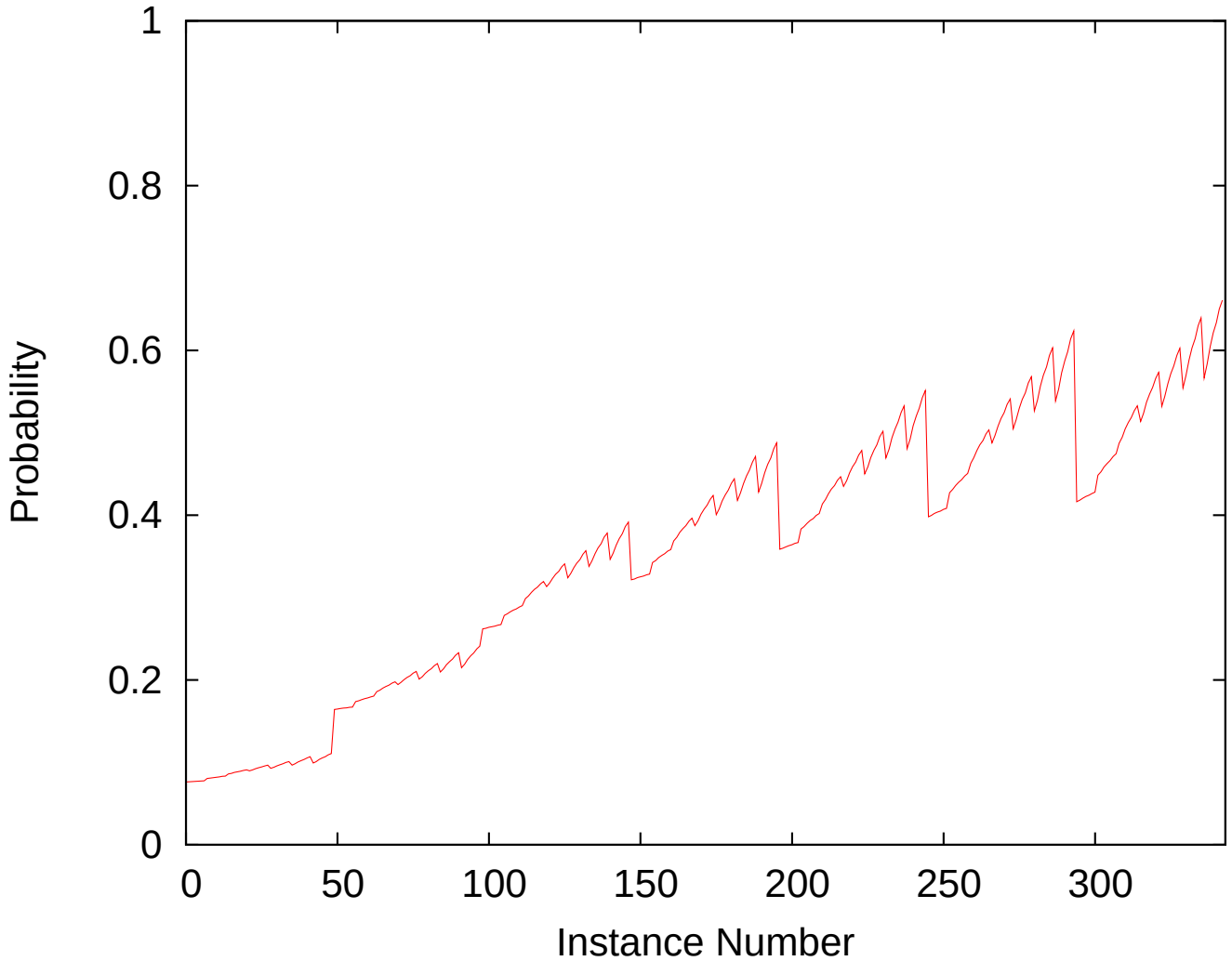
Response-time analysis at time 5



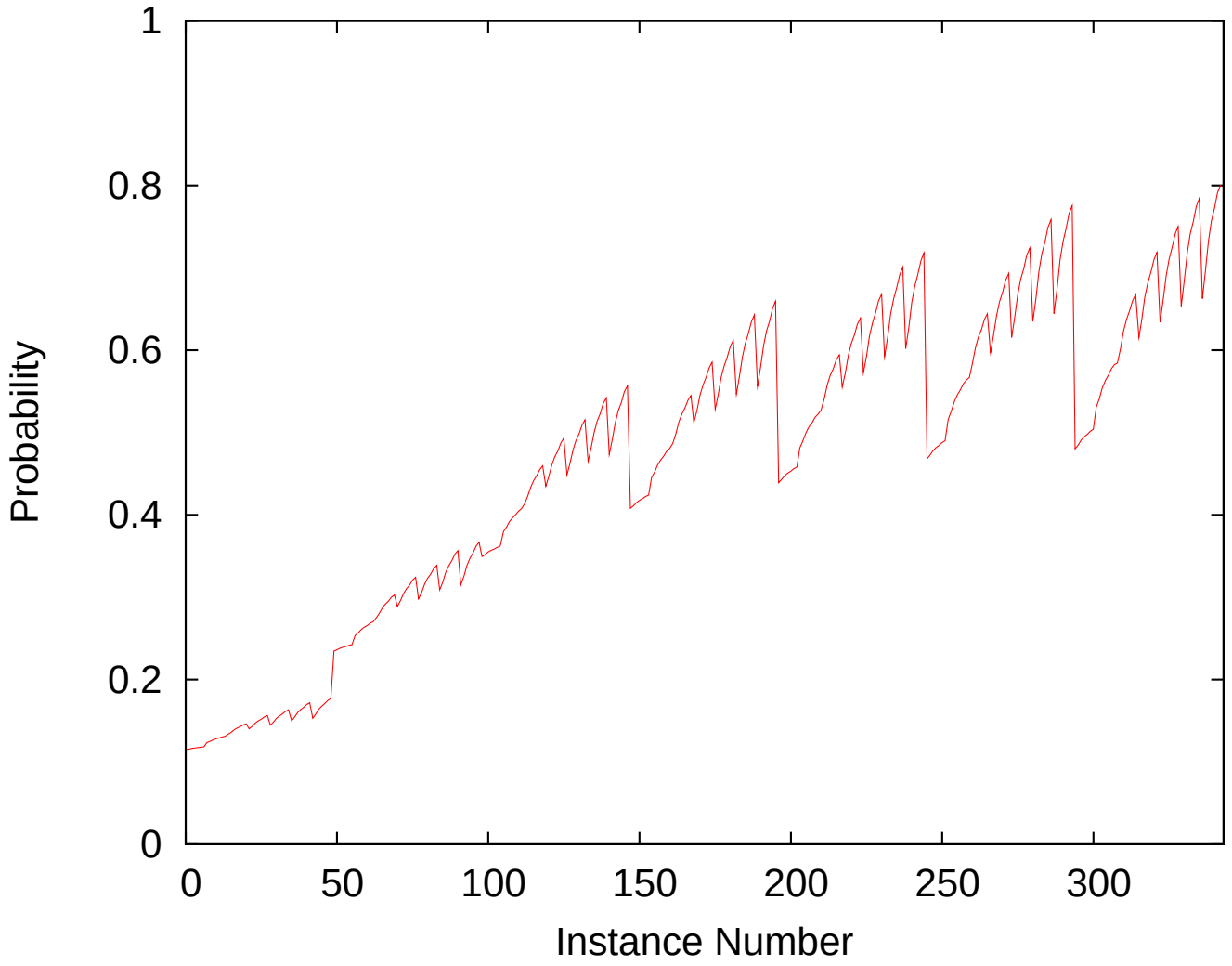
Response-time analysis at time 10



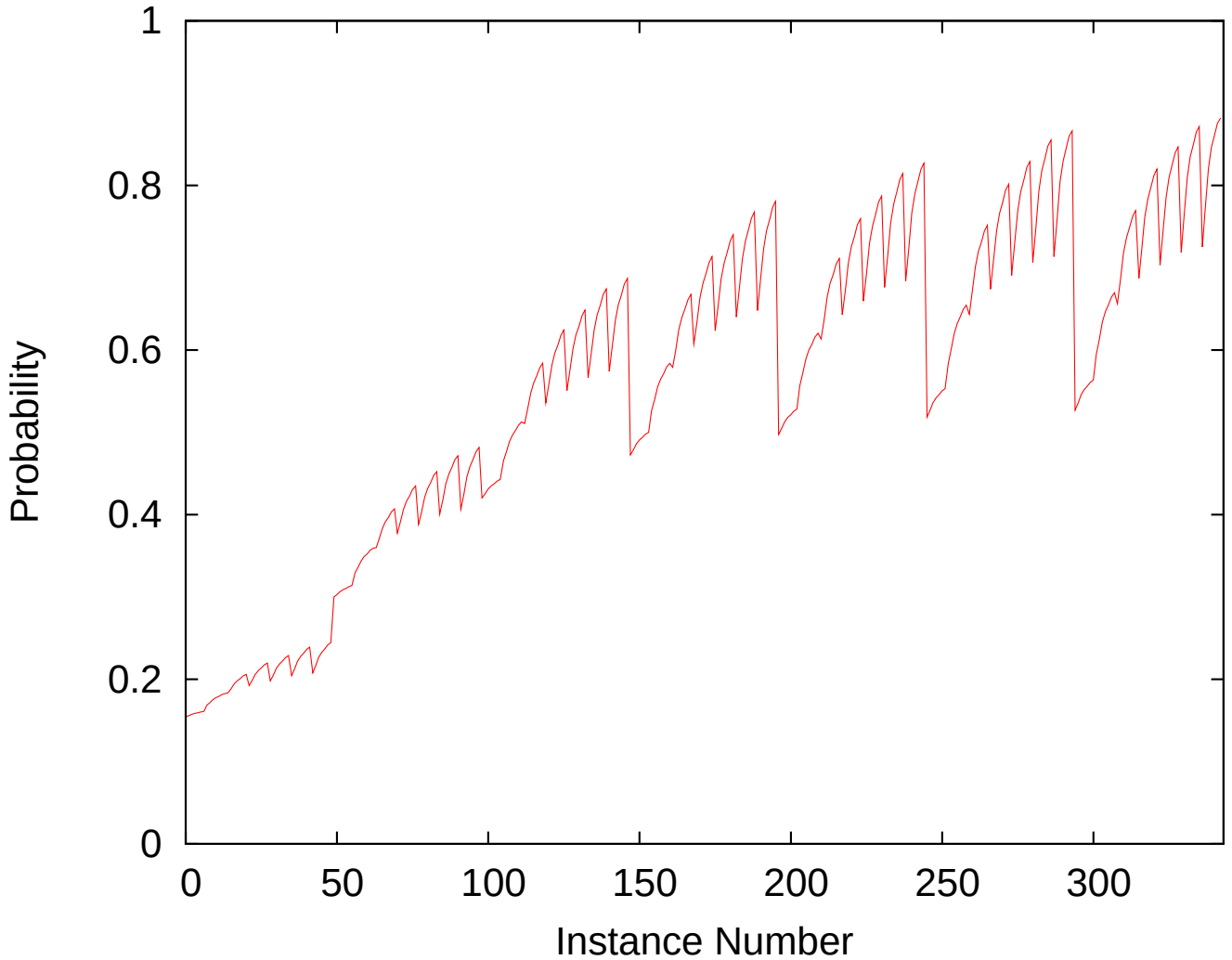
Response-time analysis at time 15



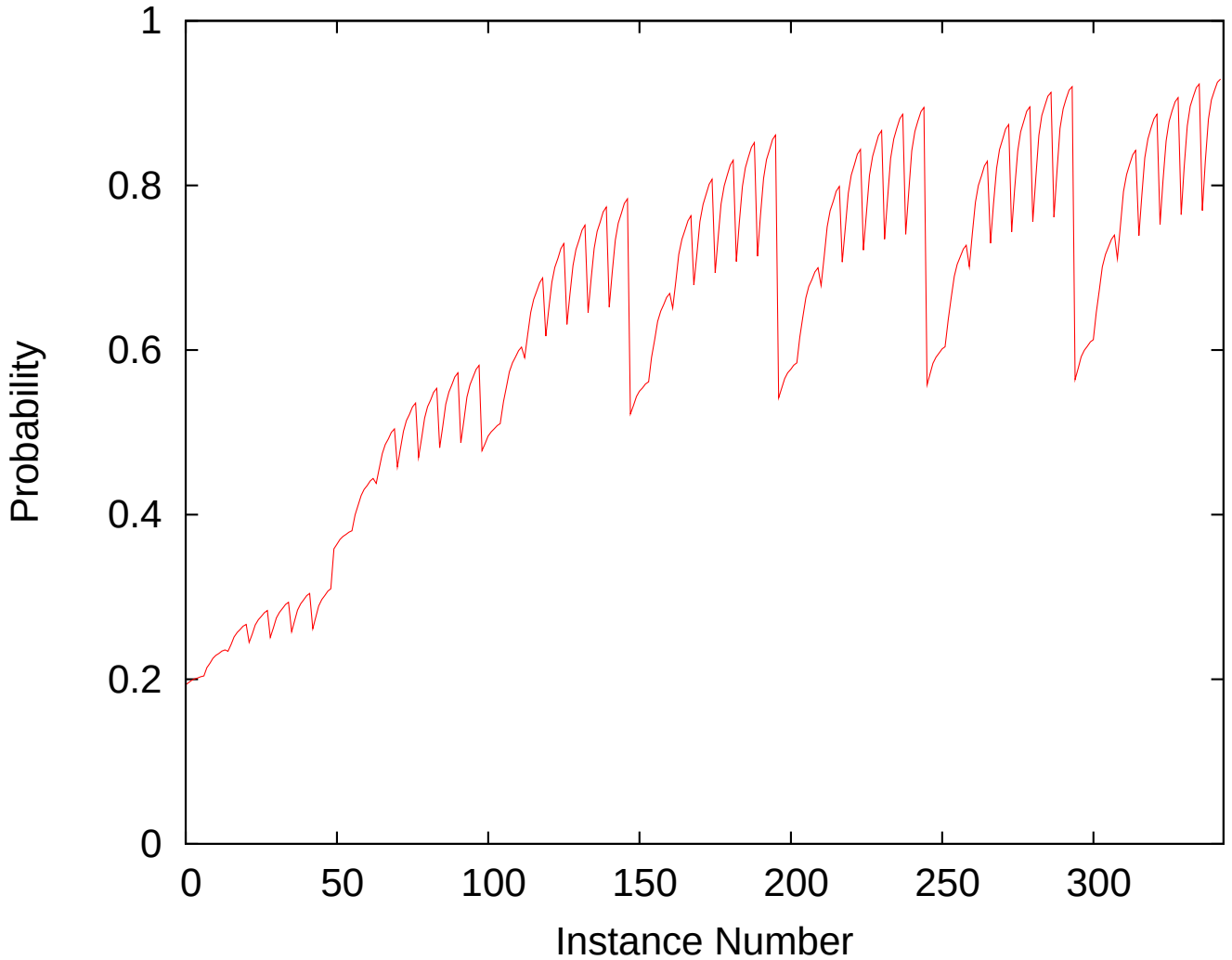
Response-time analysis at time 20



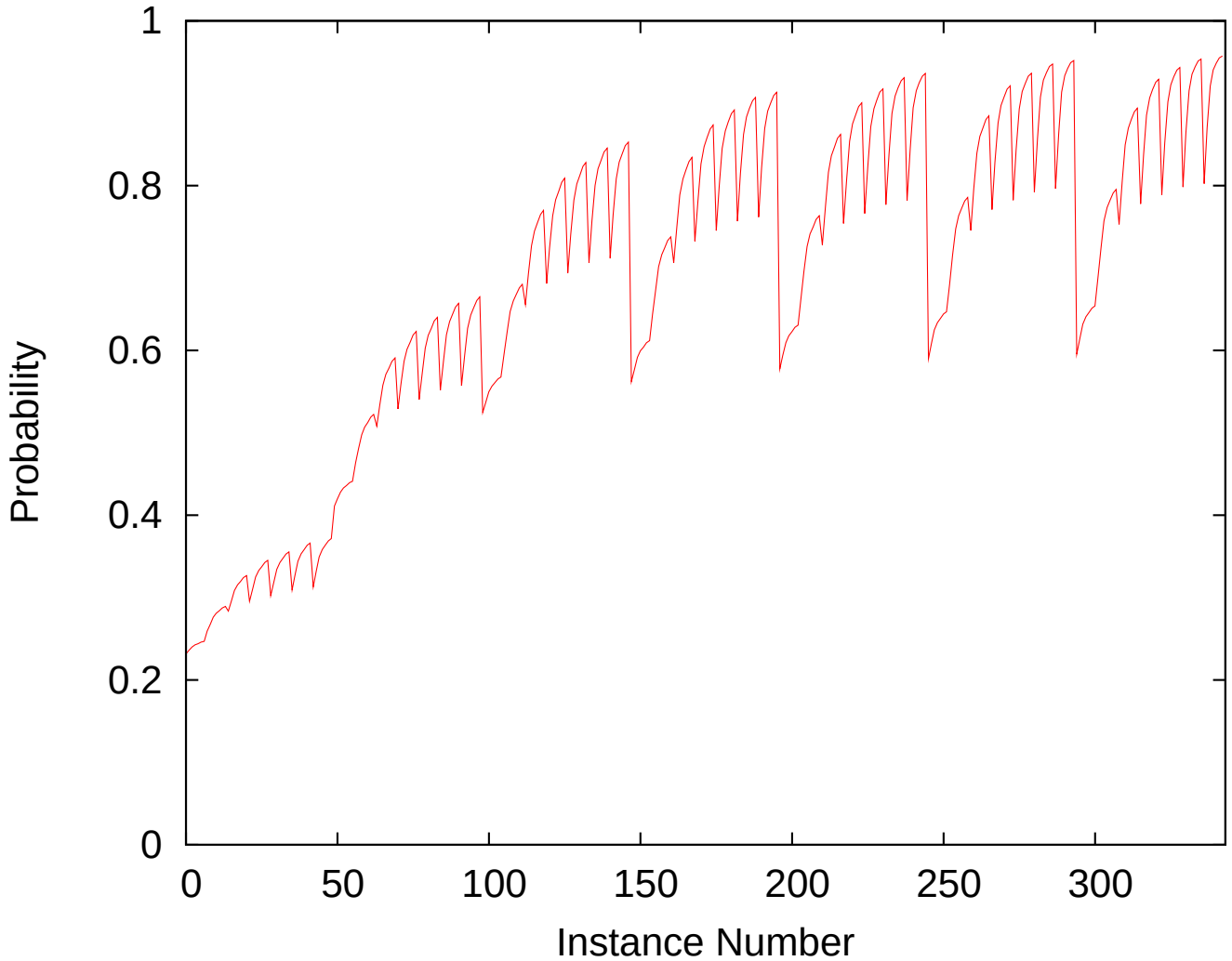
Response-time analysis at time 25



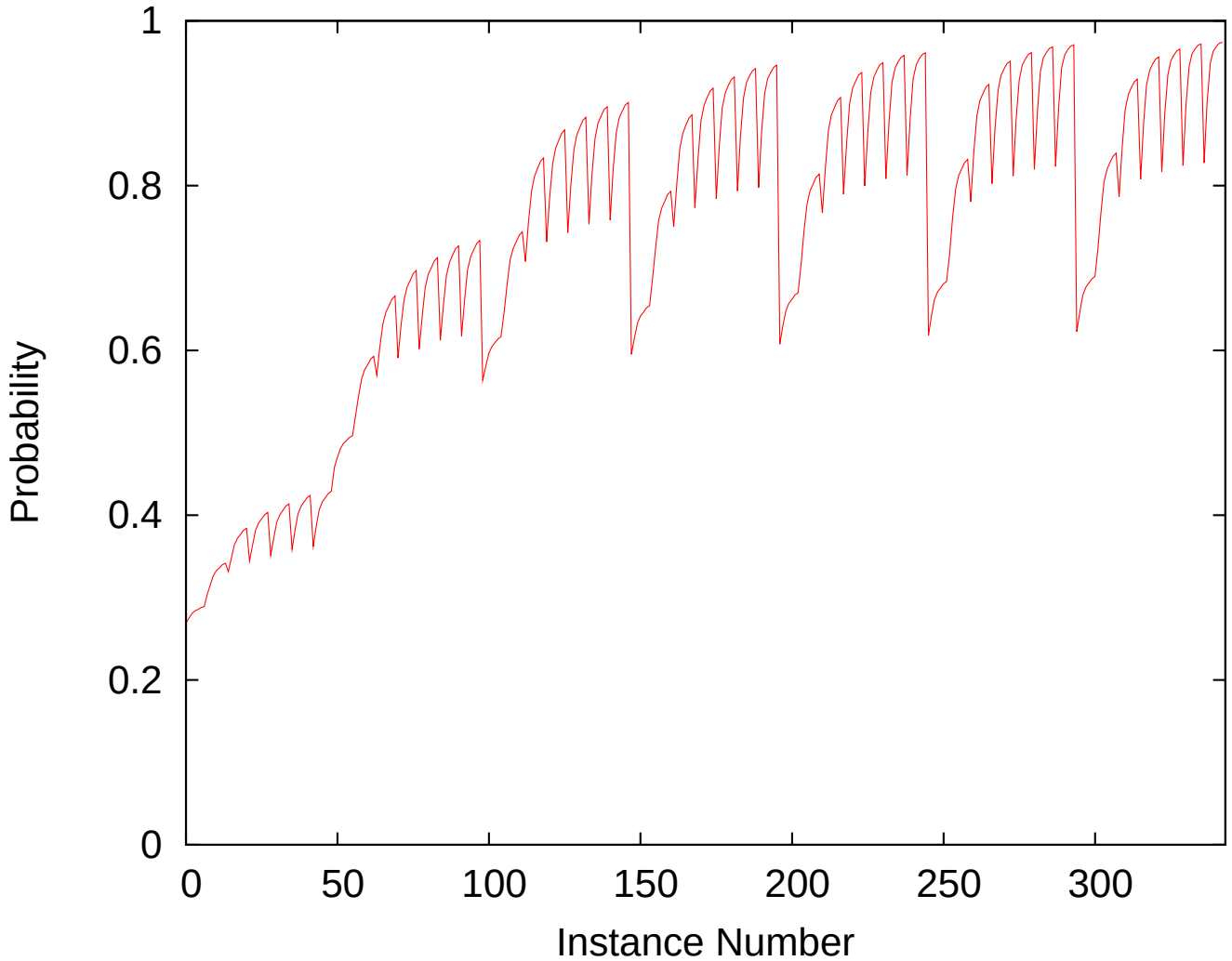
Response-time analysis at time 30



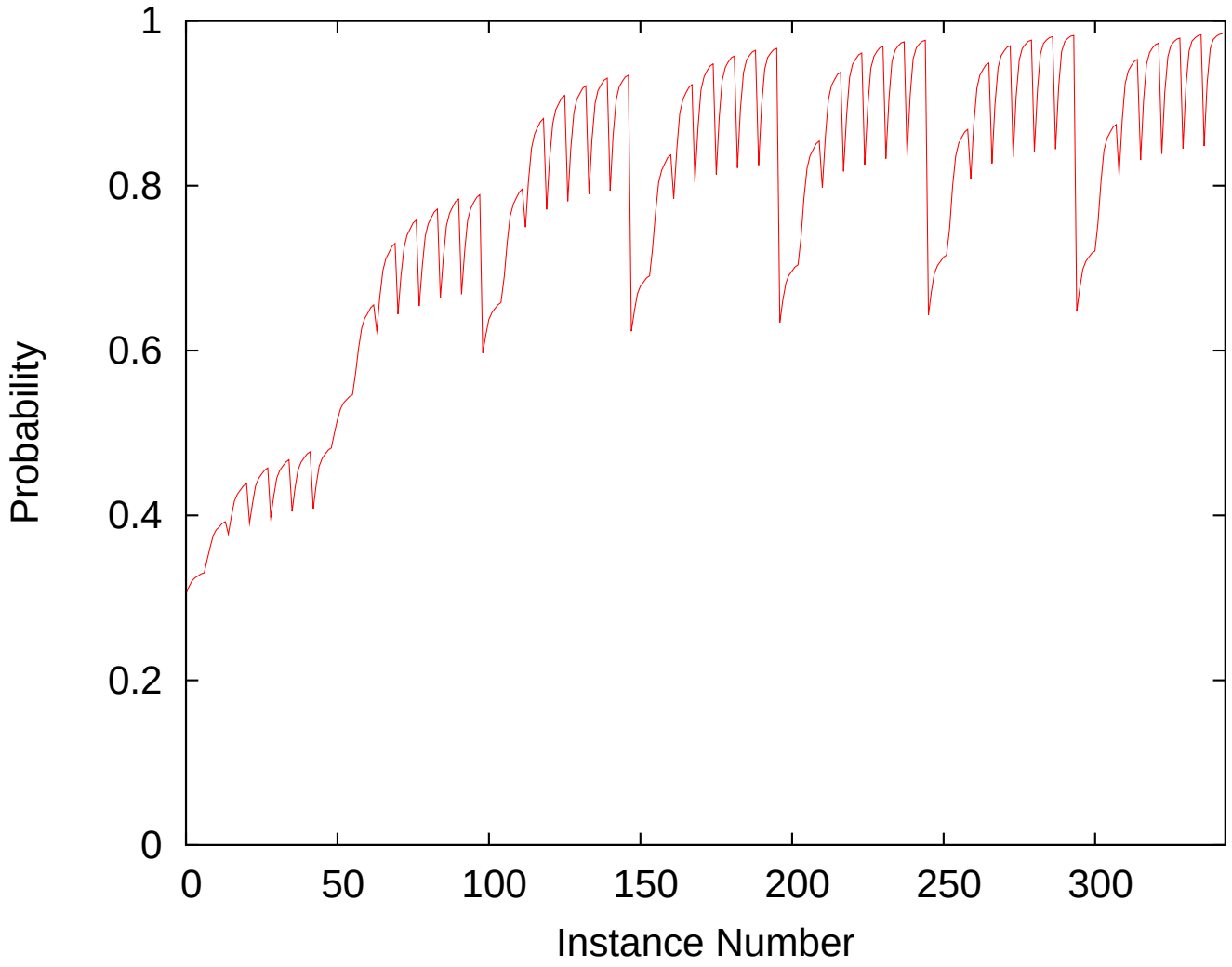
Response-time analysis at time 35



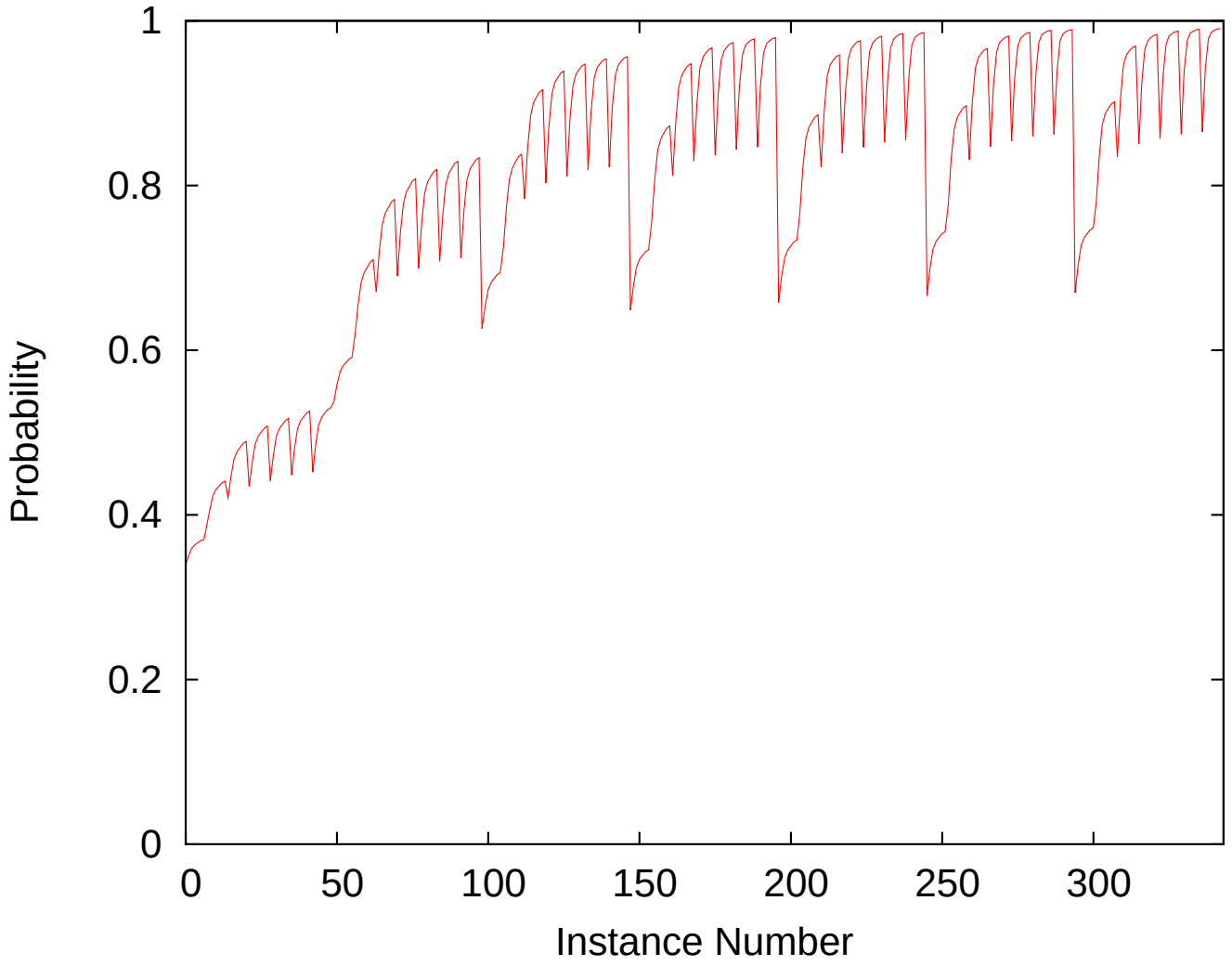
Response-time analysis at time 40



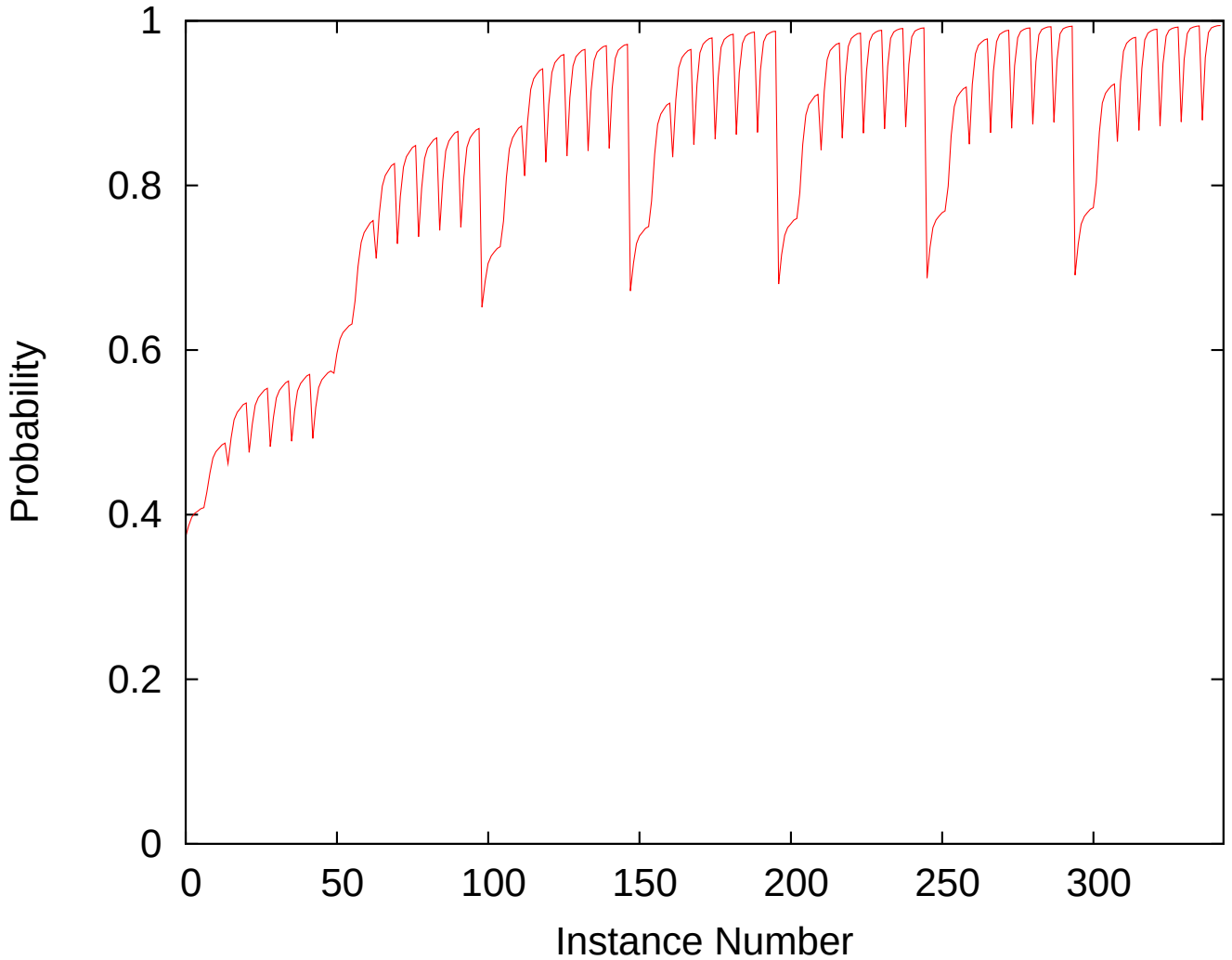
Response-time analysis at time 45



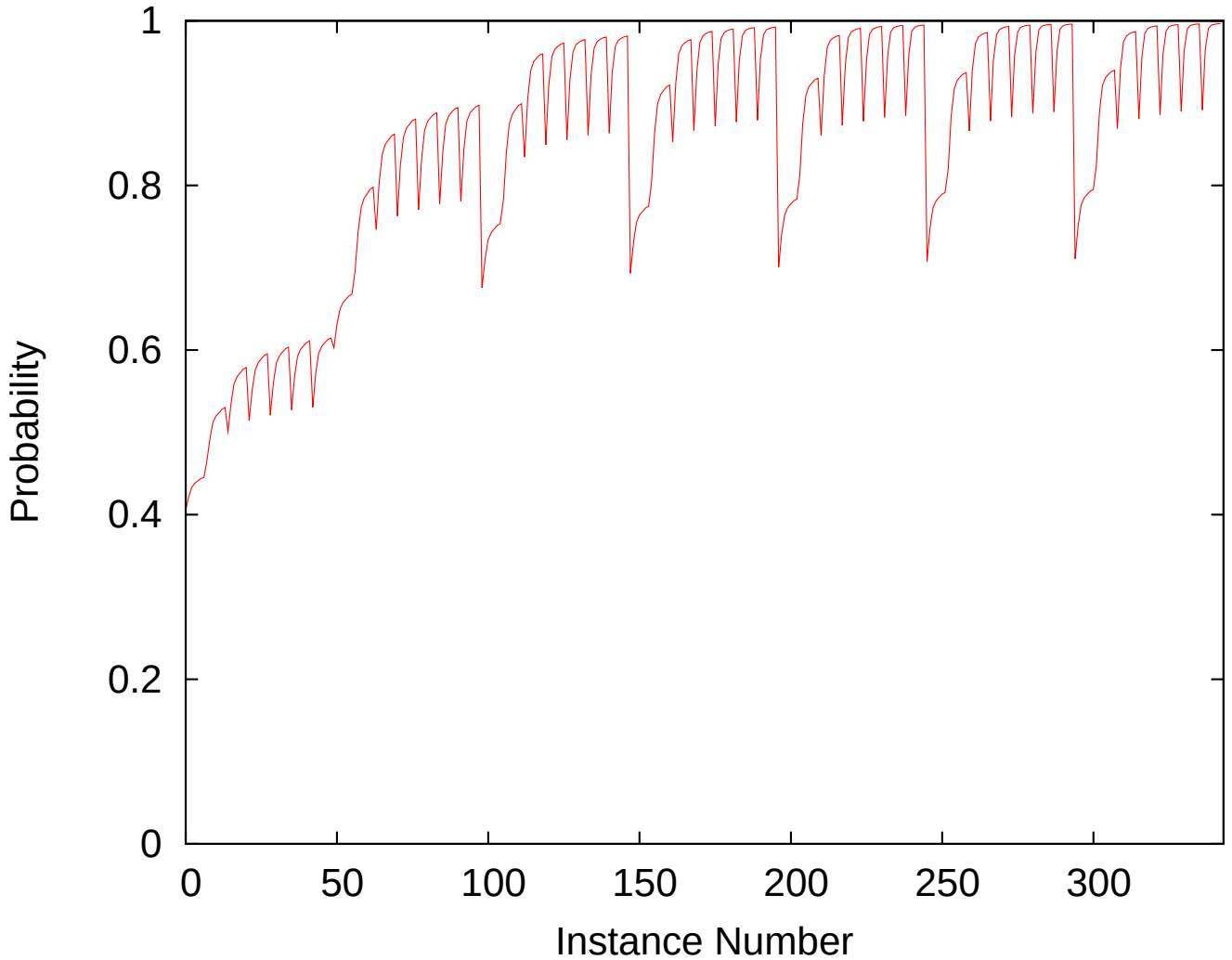
Response-time analysis at time 50



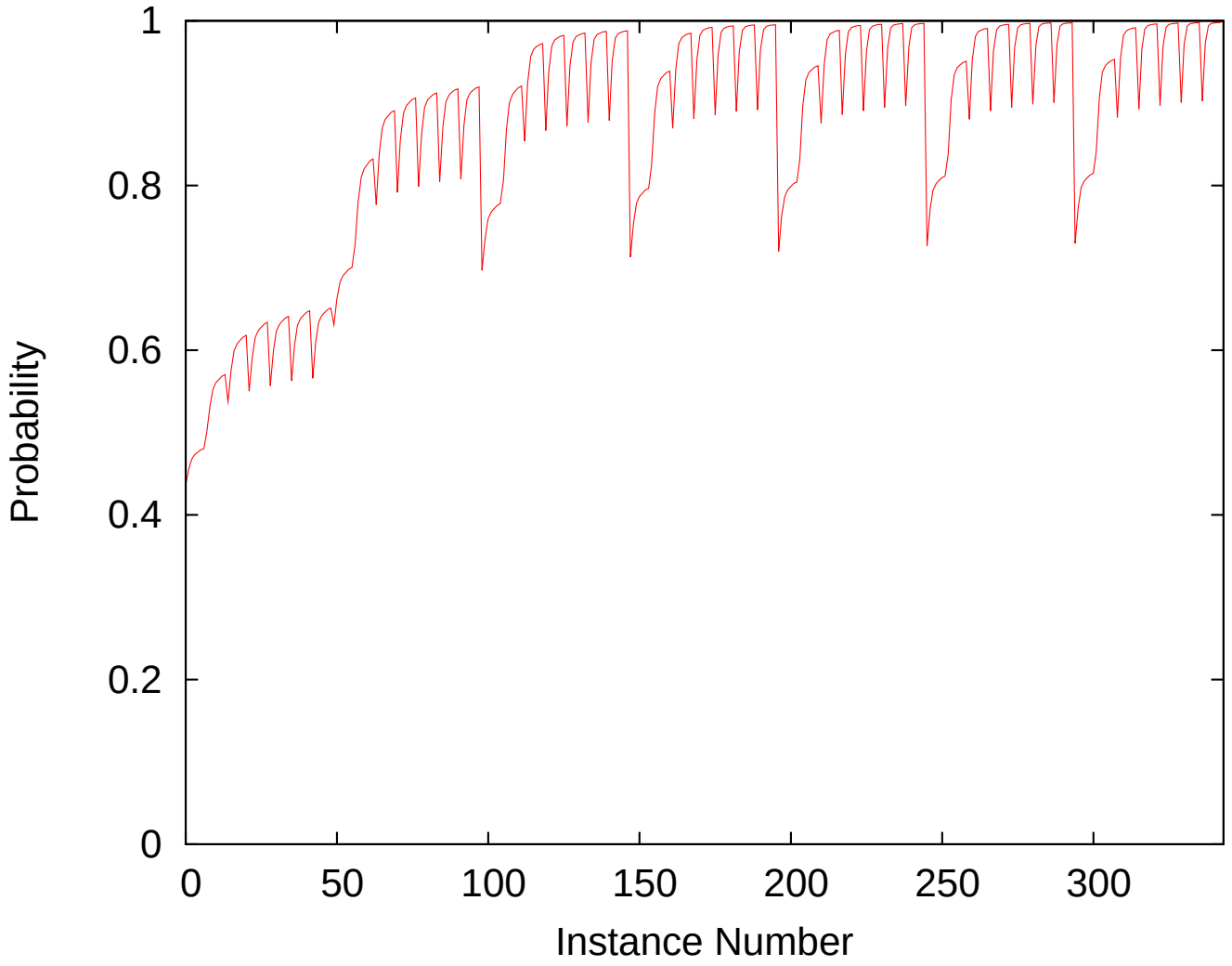
Response-time analysis at time 55



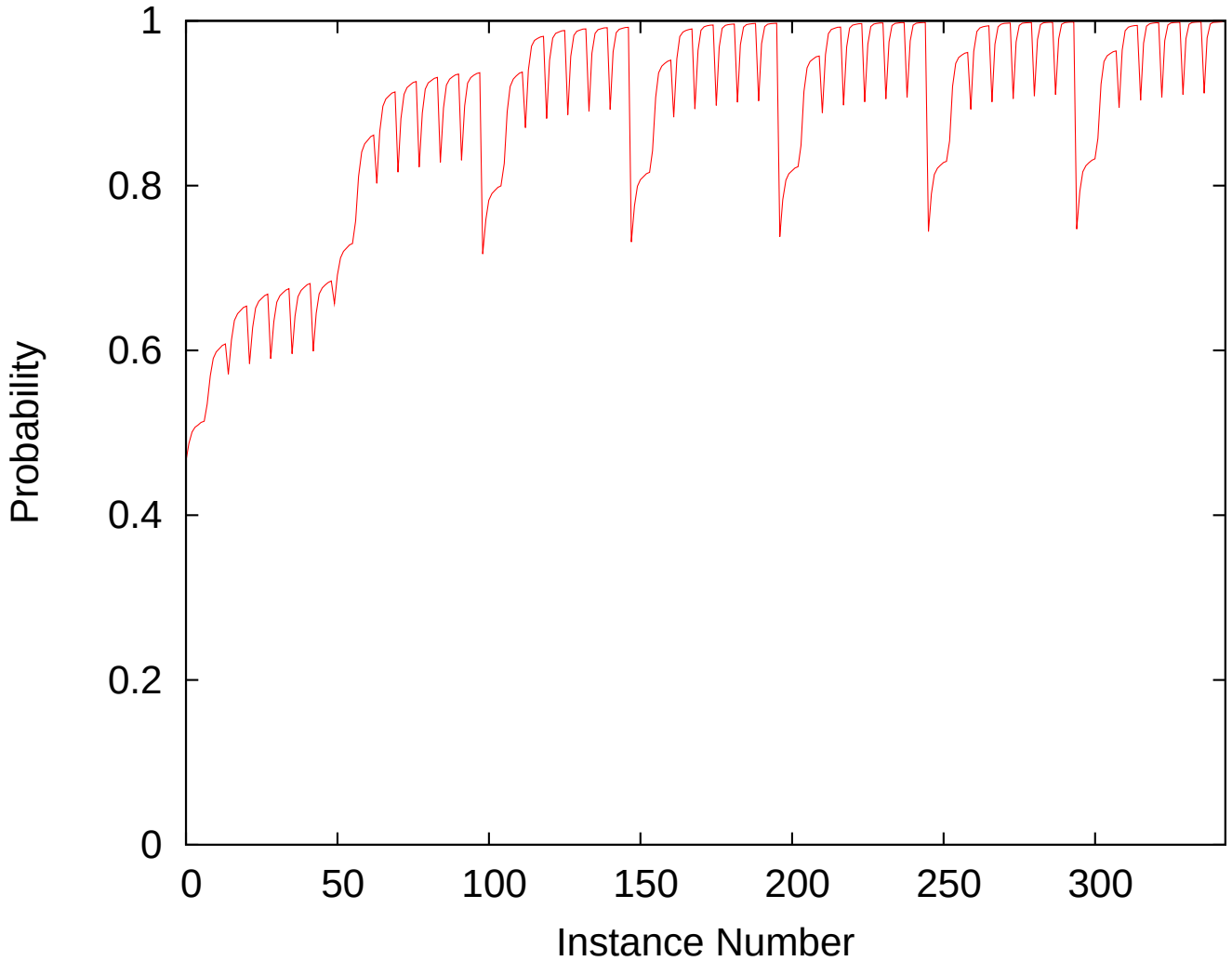
Response-time analysis at time 60



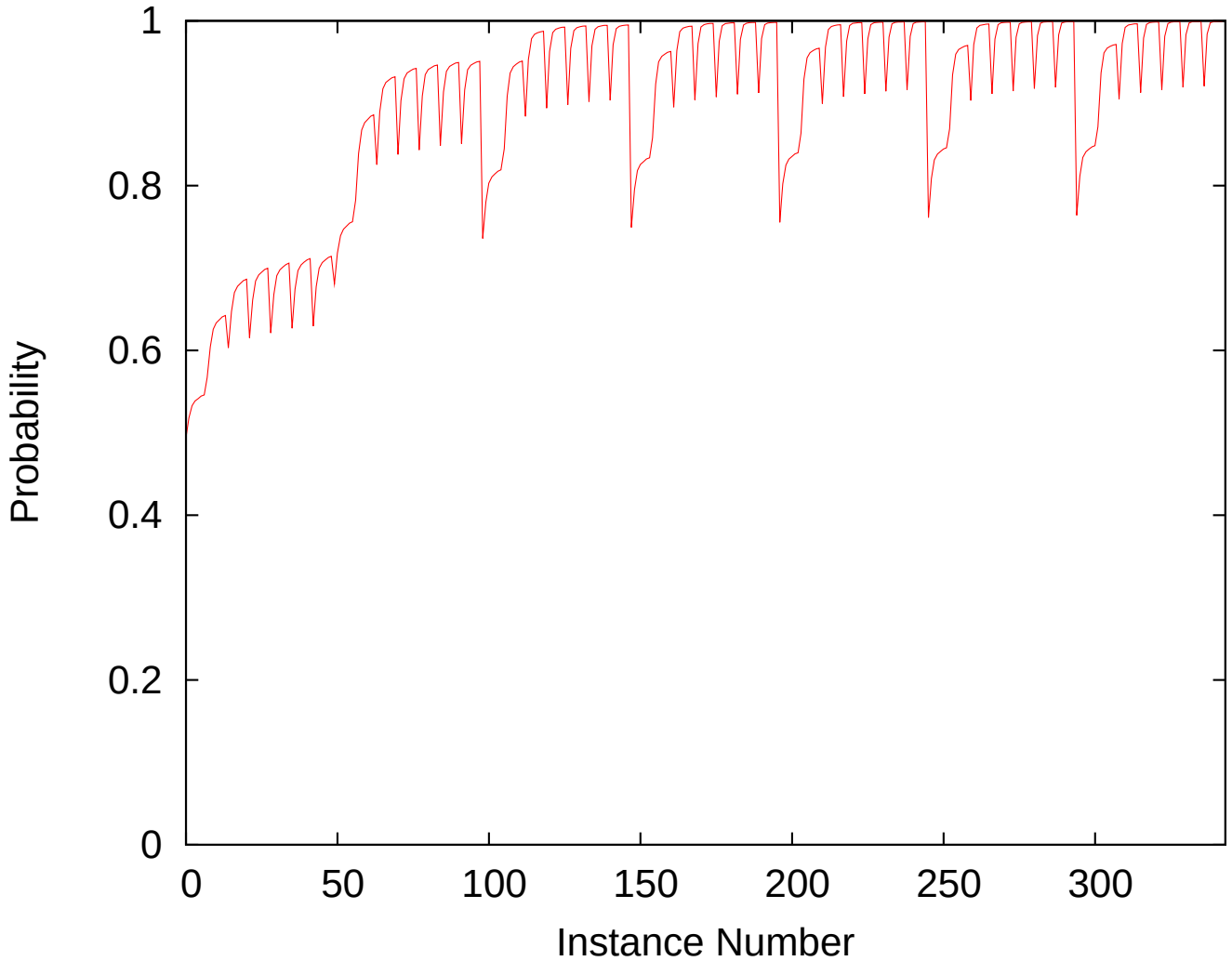
Response-time analysis at time 65



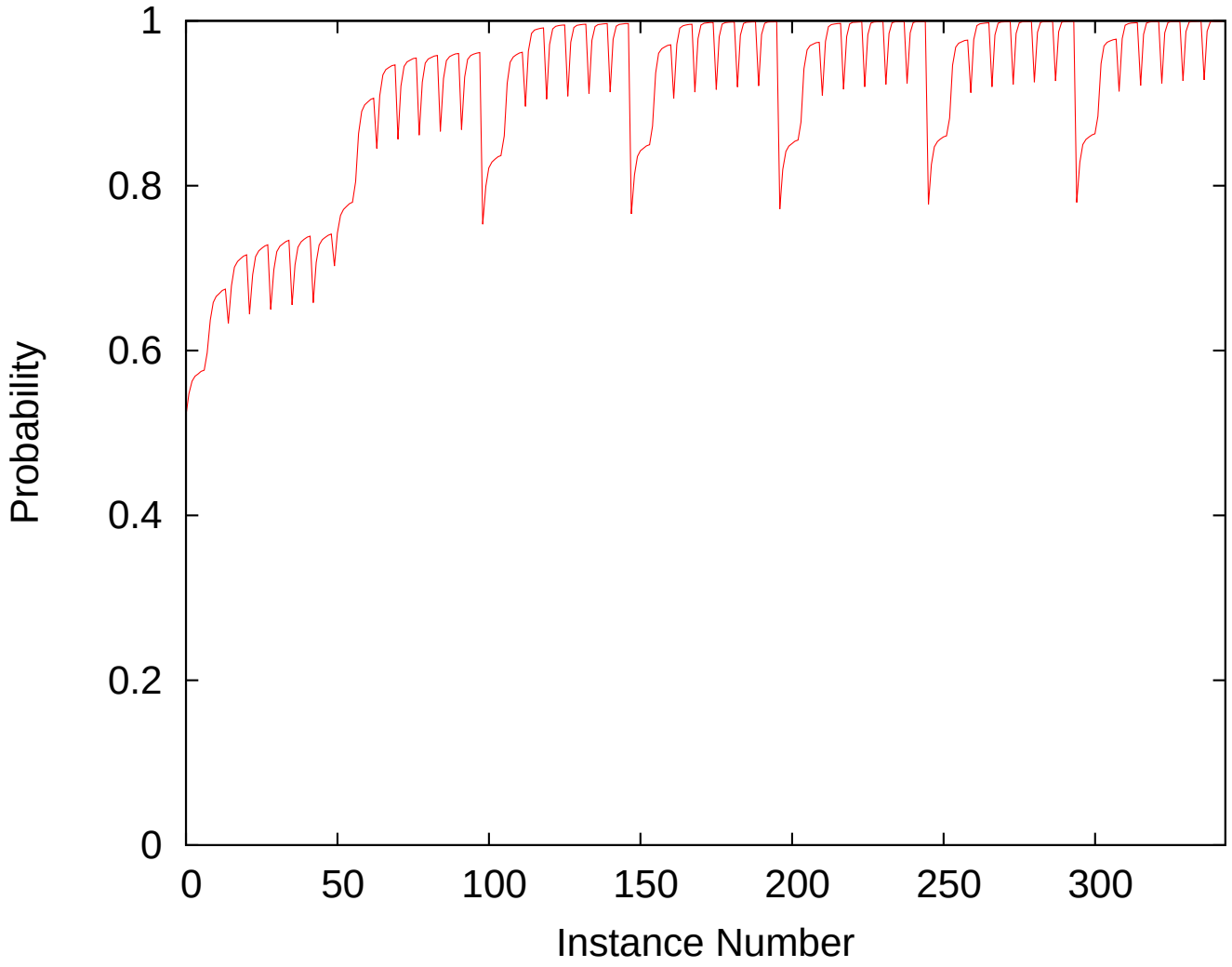
Response-time analysis at time 70



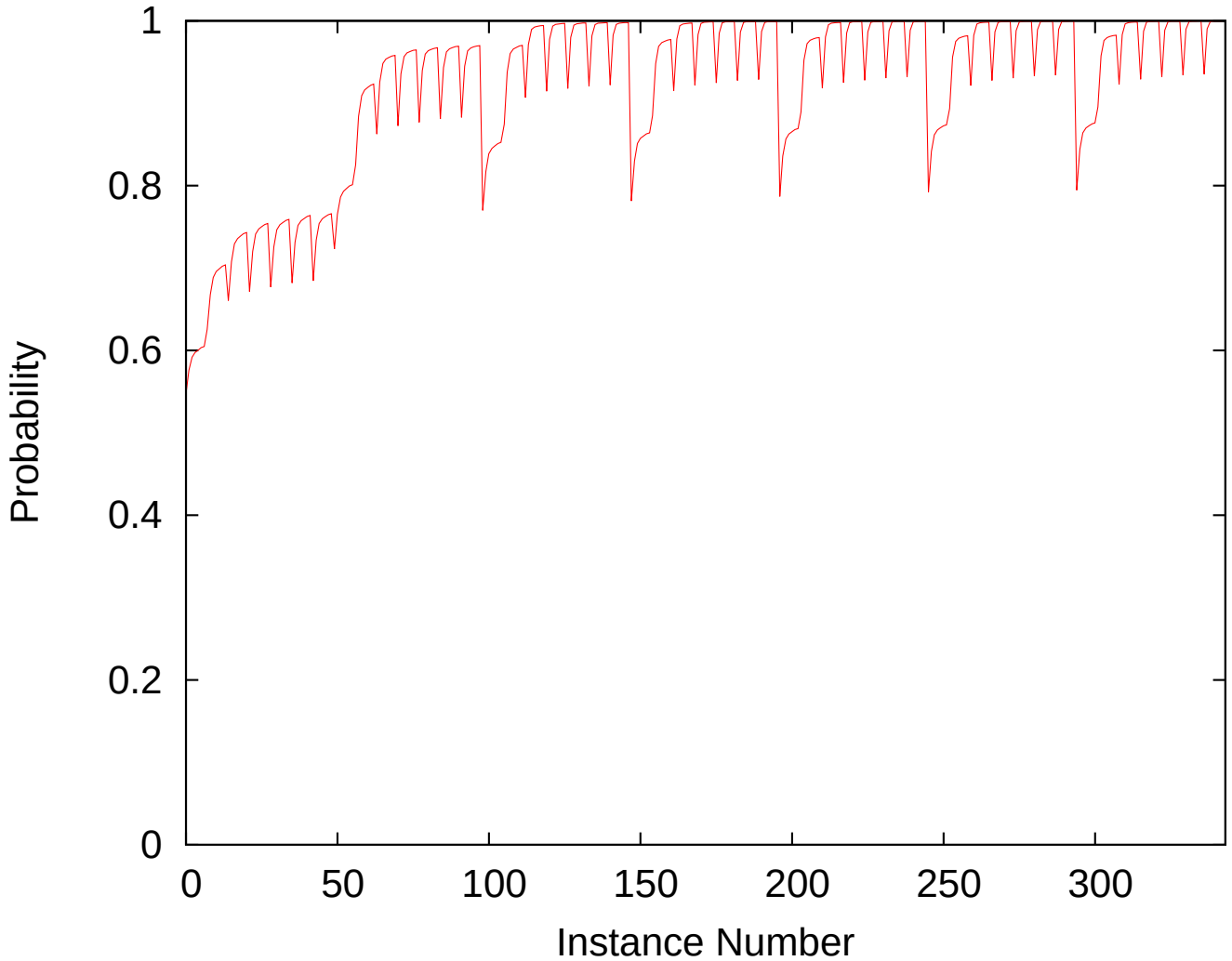
Response-time analysis at time 75



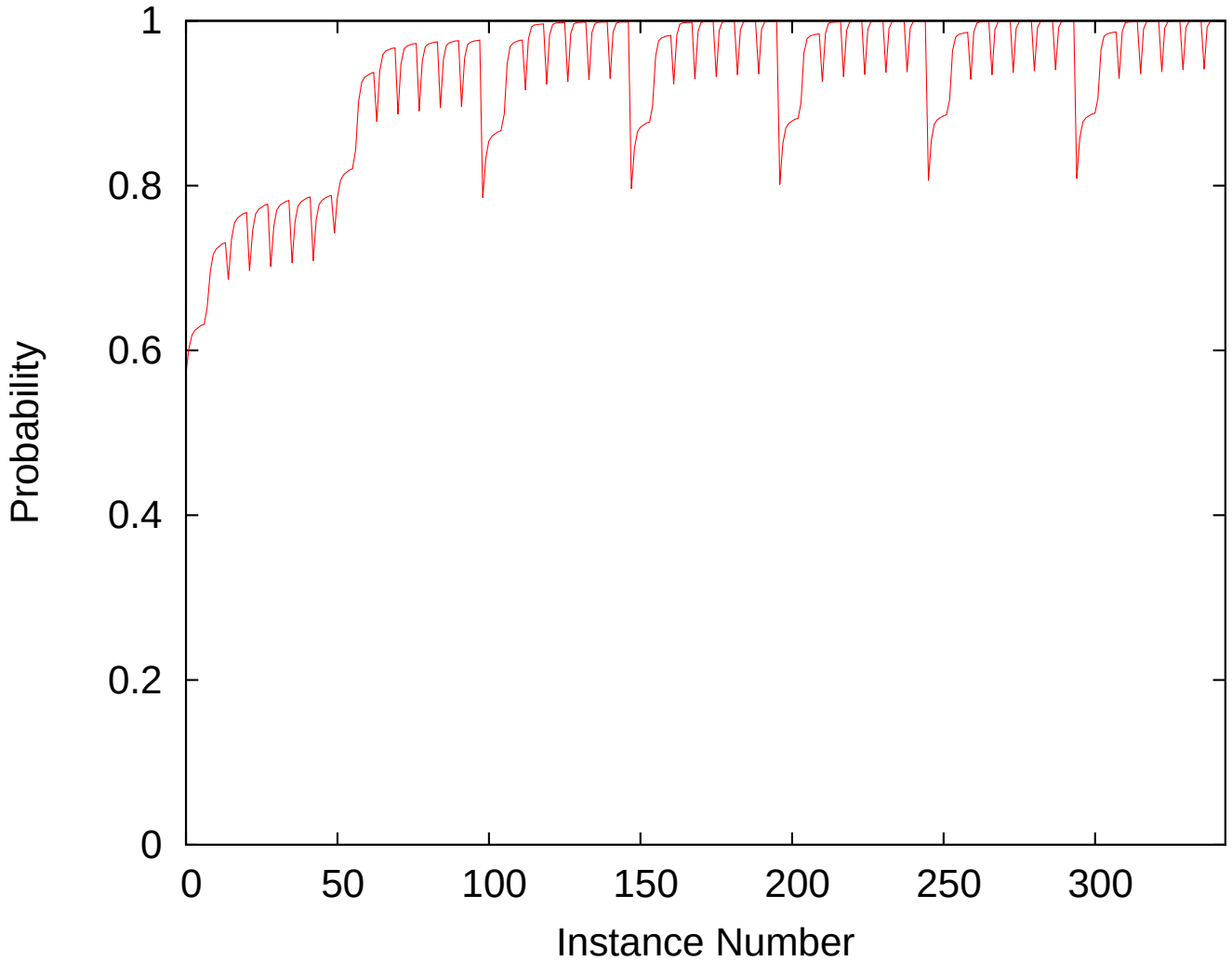
Response-time analysis at time 80



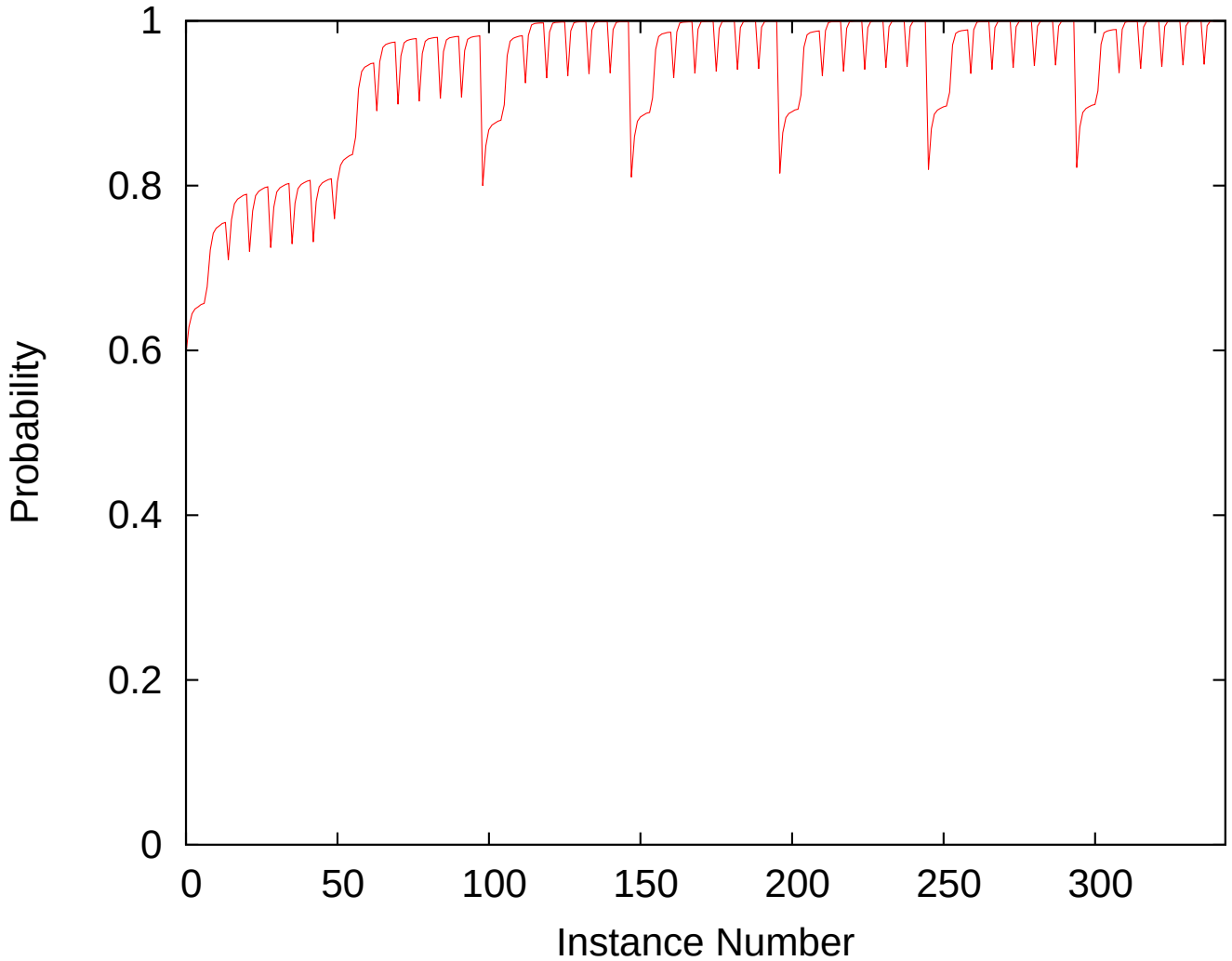
Response-time analysis at time 85



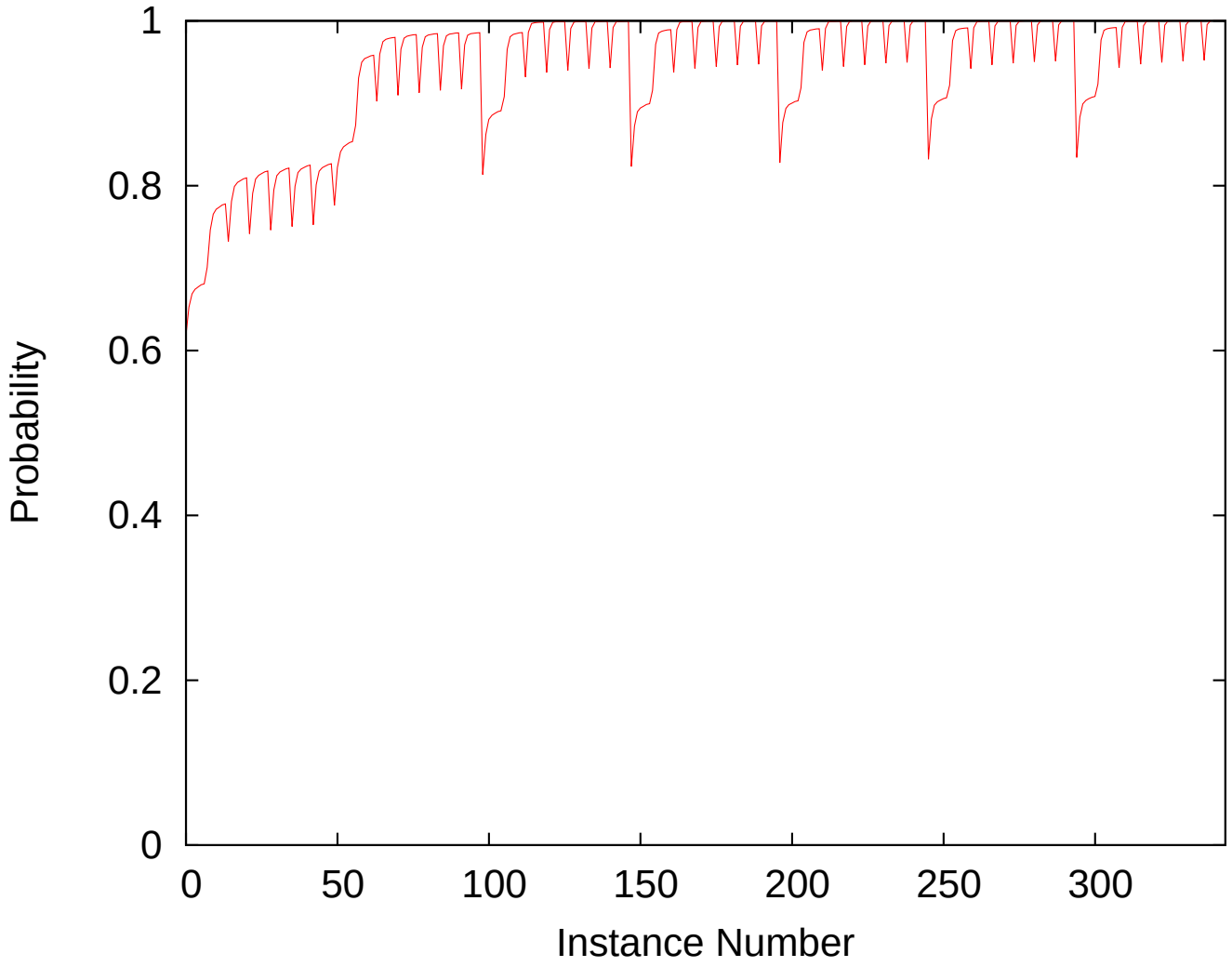
Response-time analysis at time 90



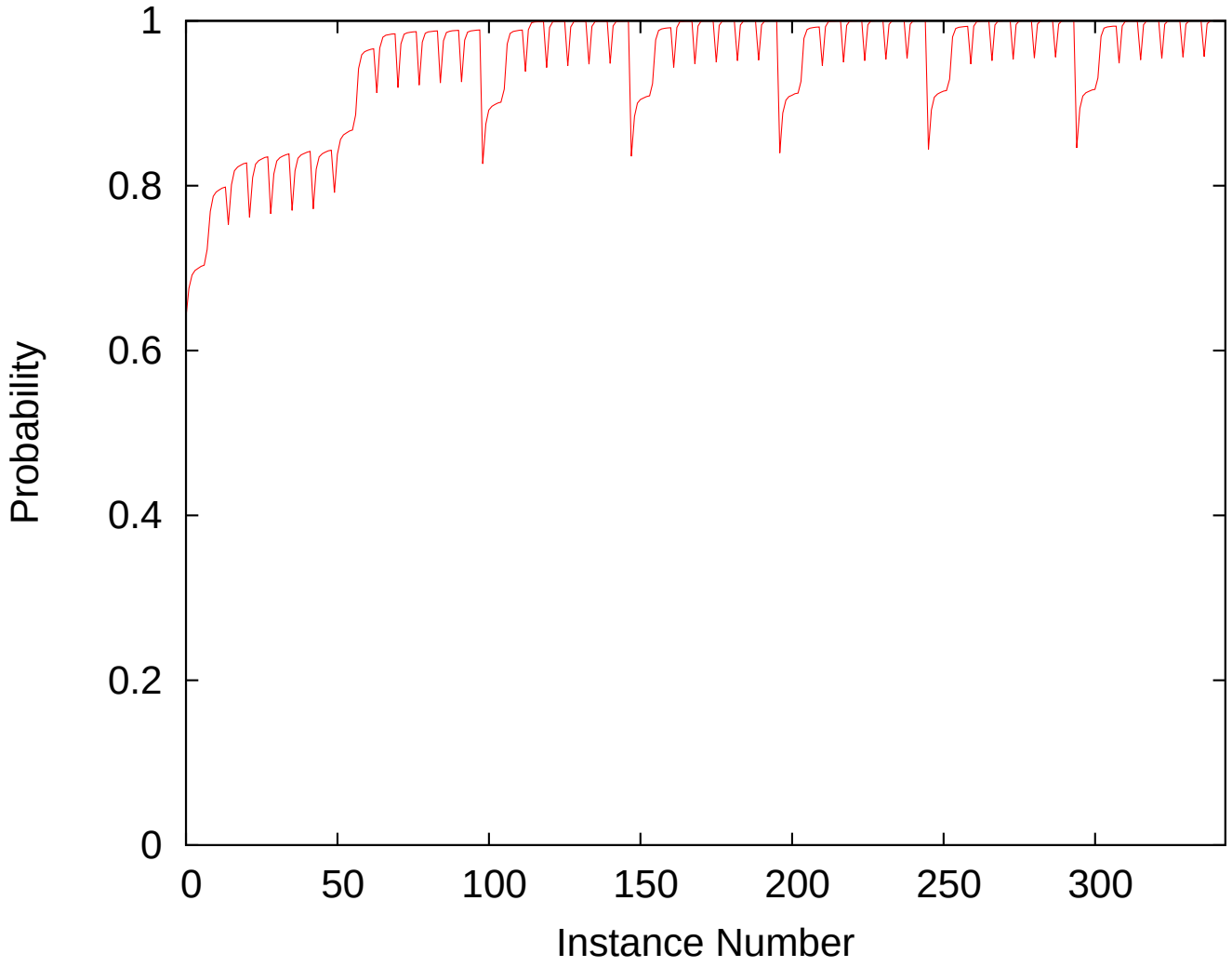
Response-time analysis at time 95



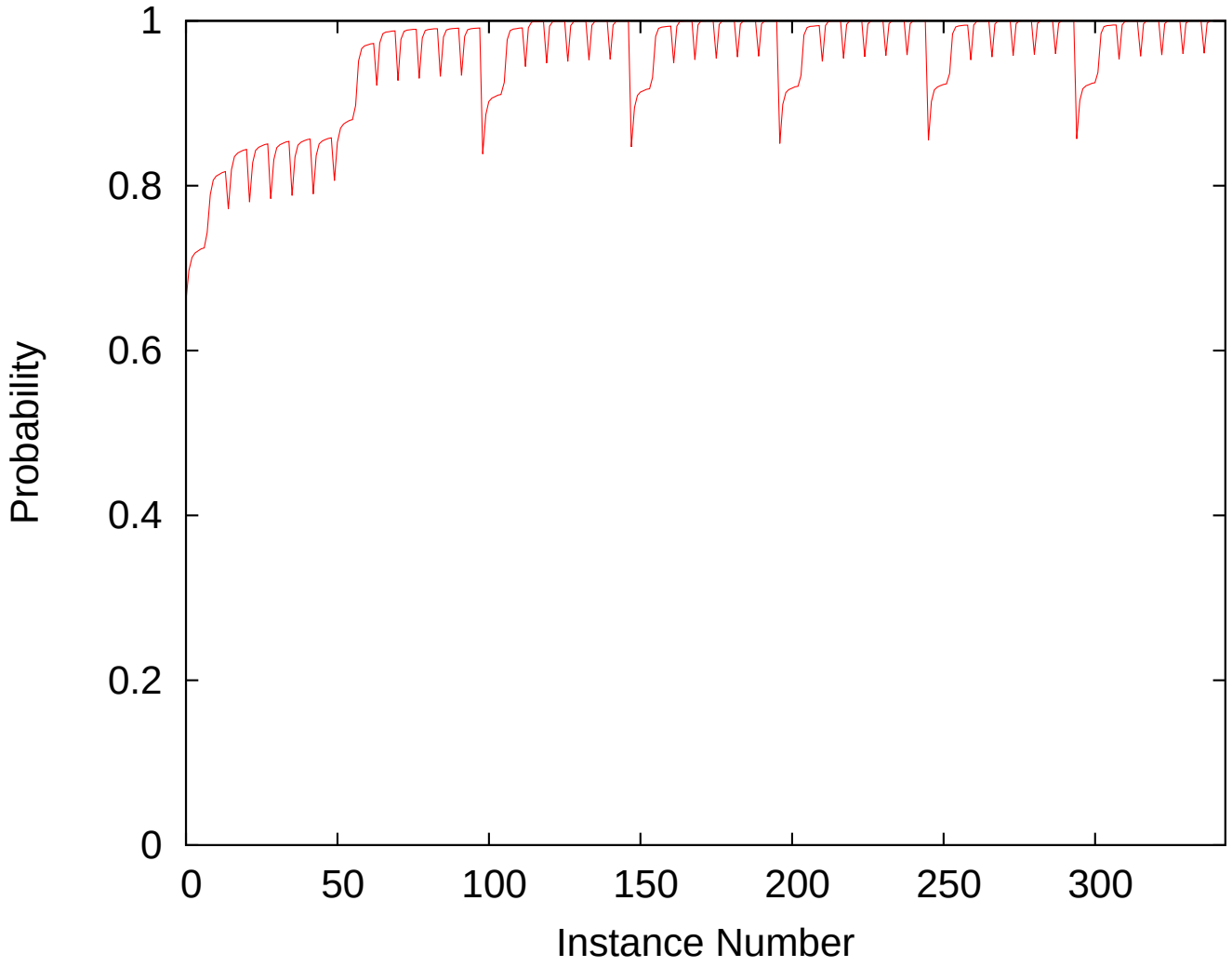
Response-time analysis at time 100



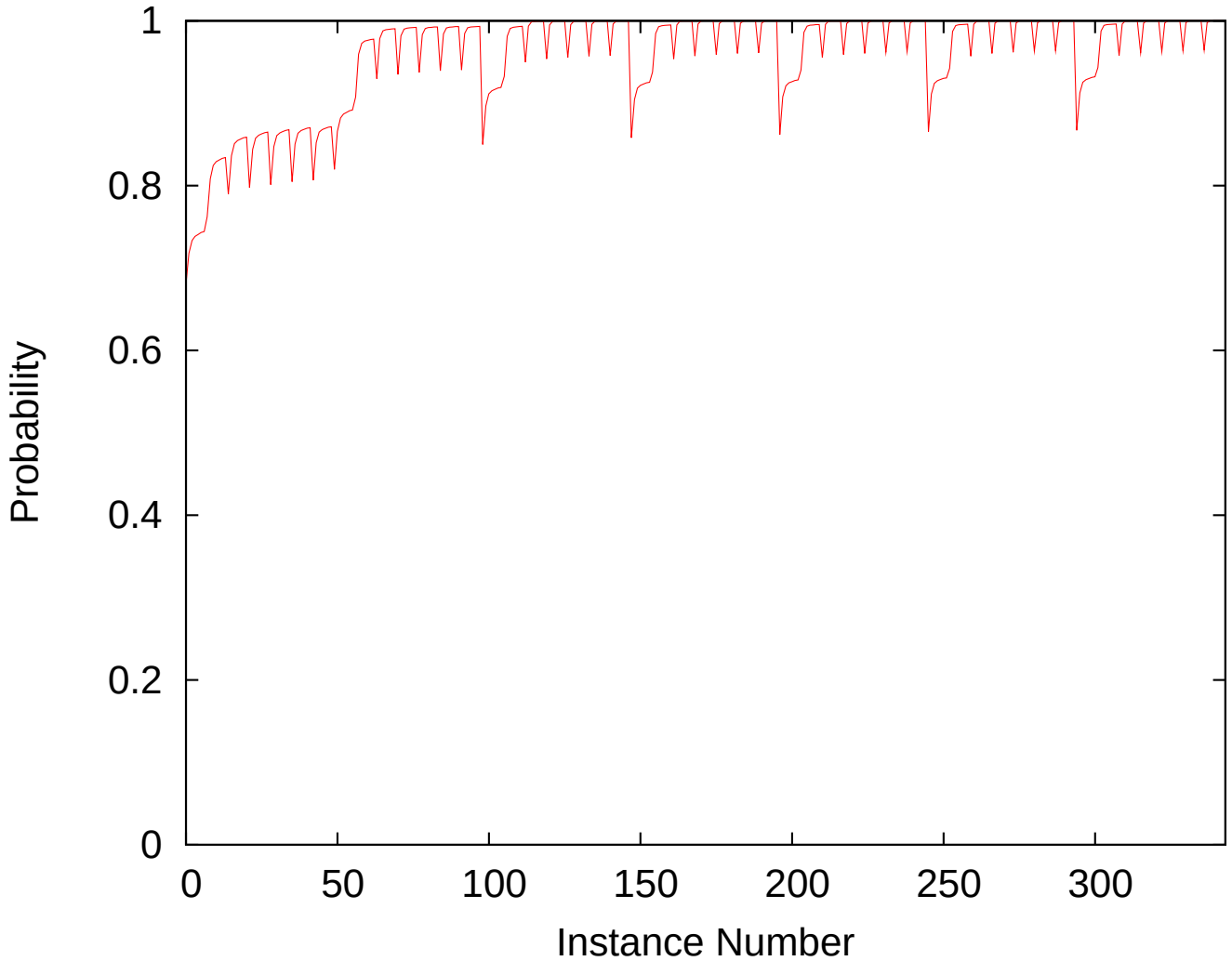
Response-time analysis at time 105



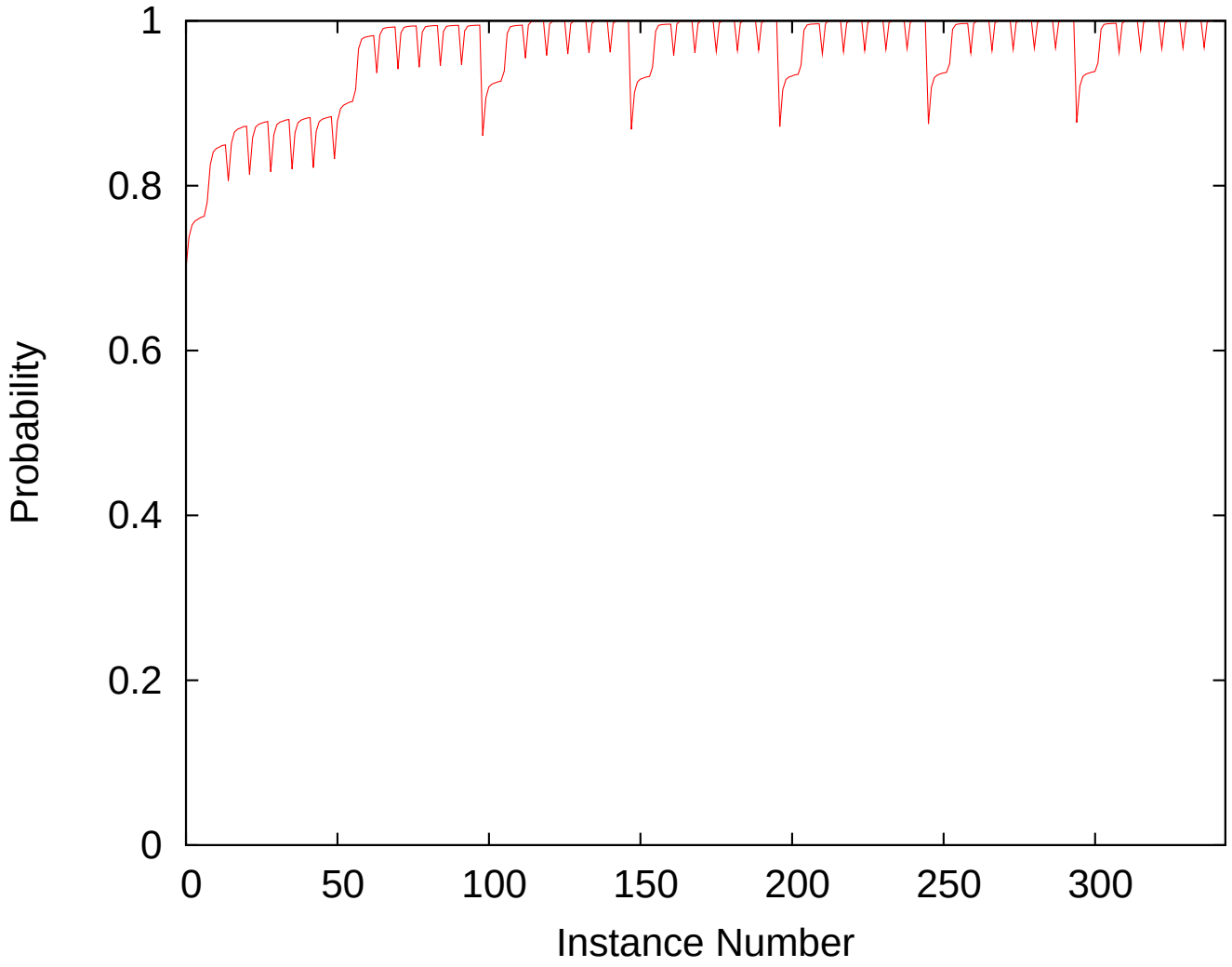
Response-time analysis at time 110



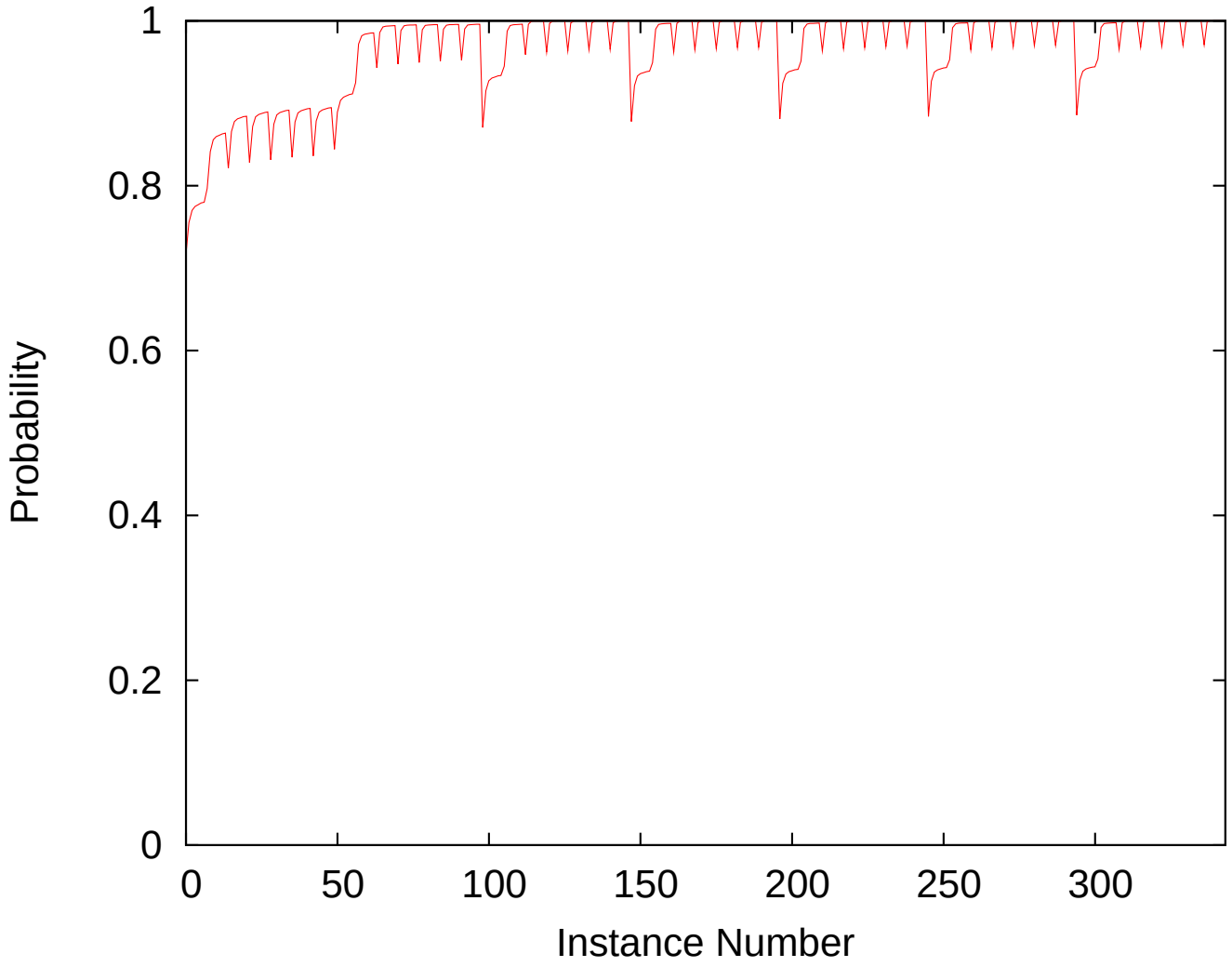
Response-time analysis at time 115



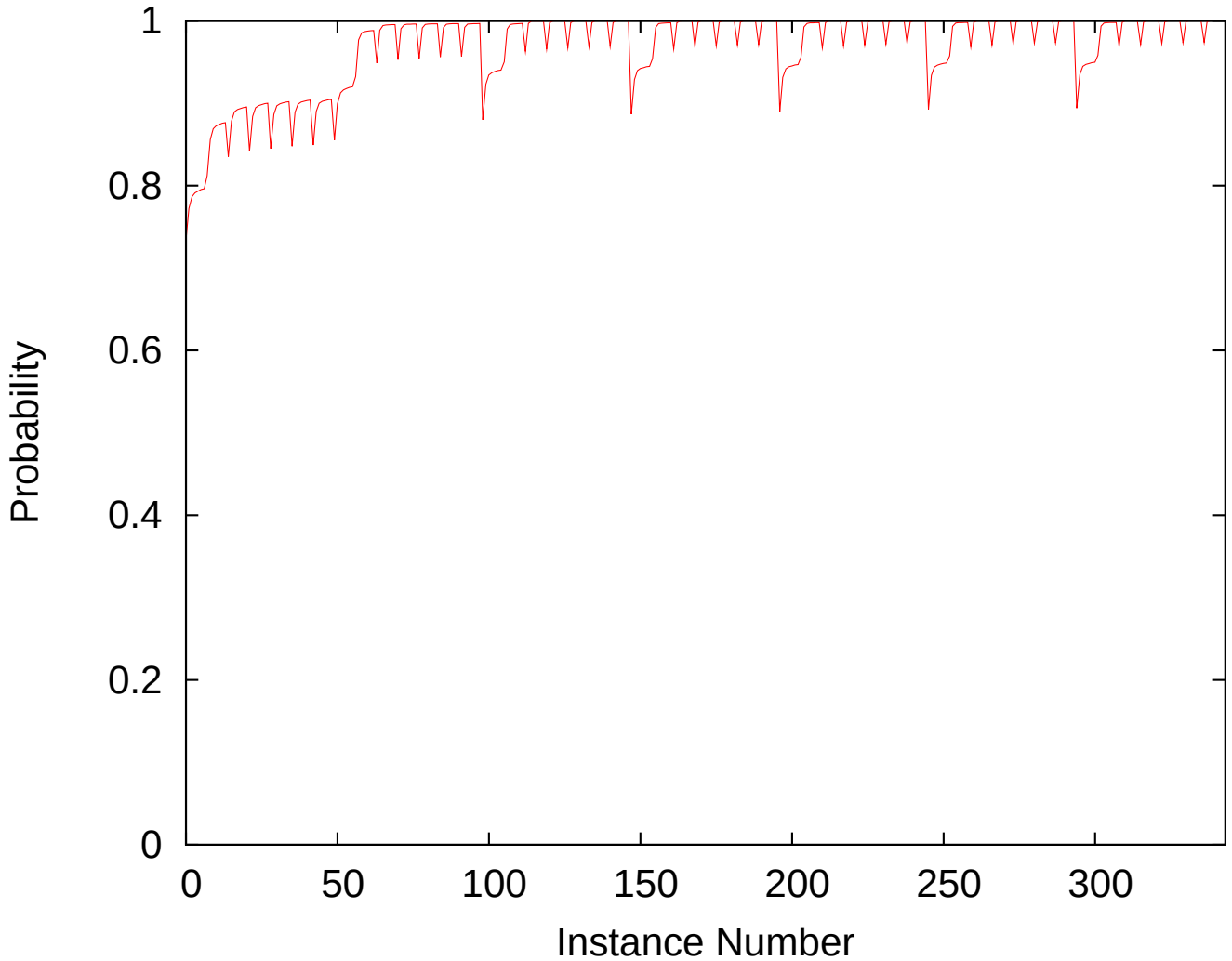
Response-time analysis at time 120



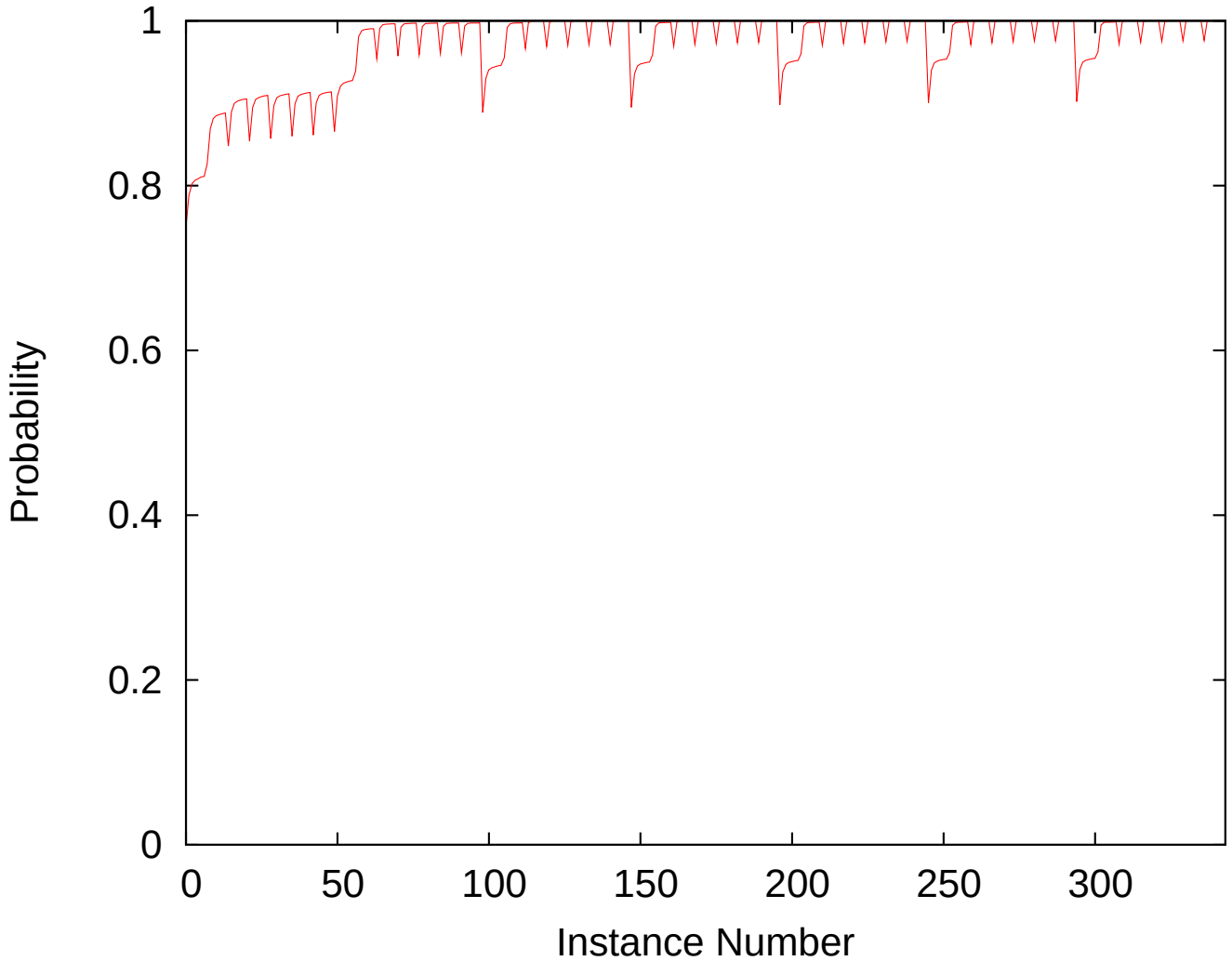
Response-time analysis at time 125



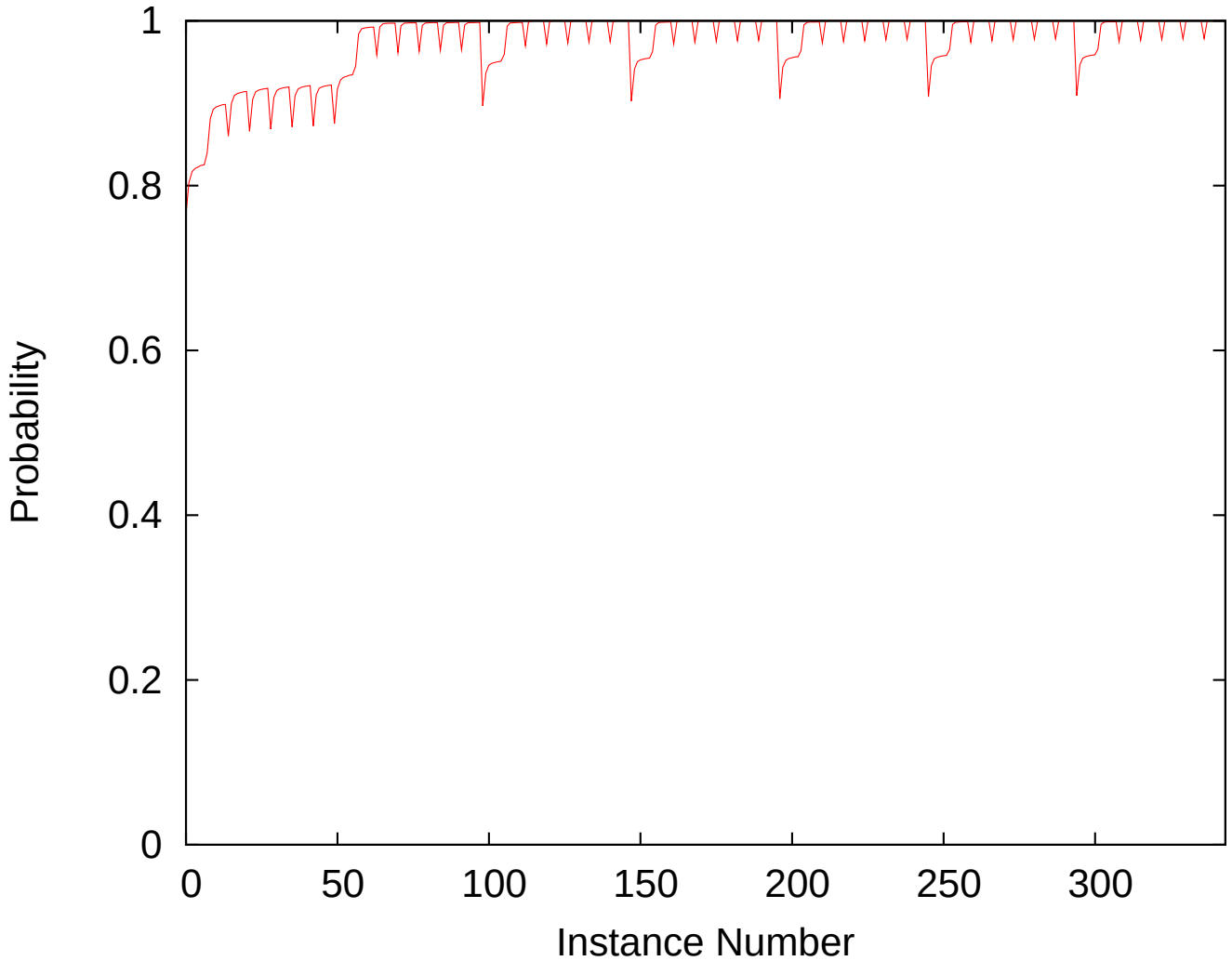
Response-time analysis at time 130



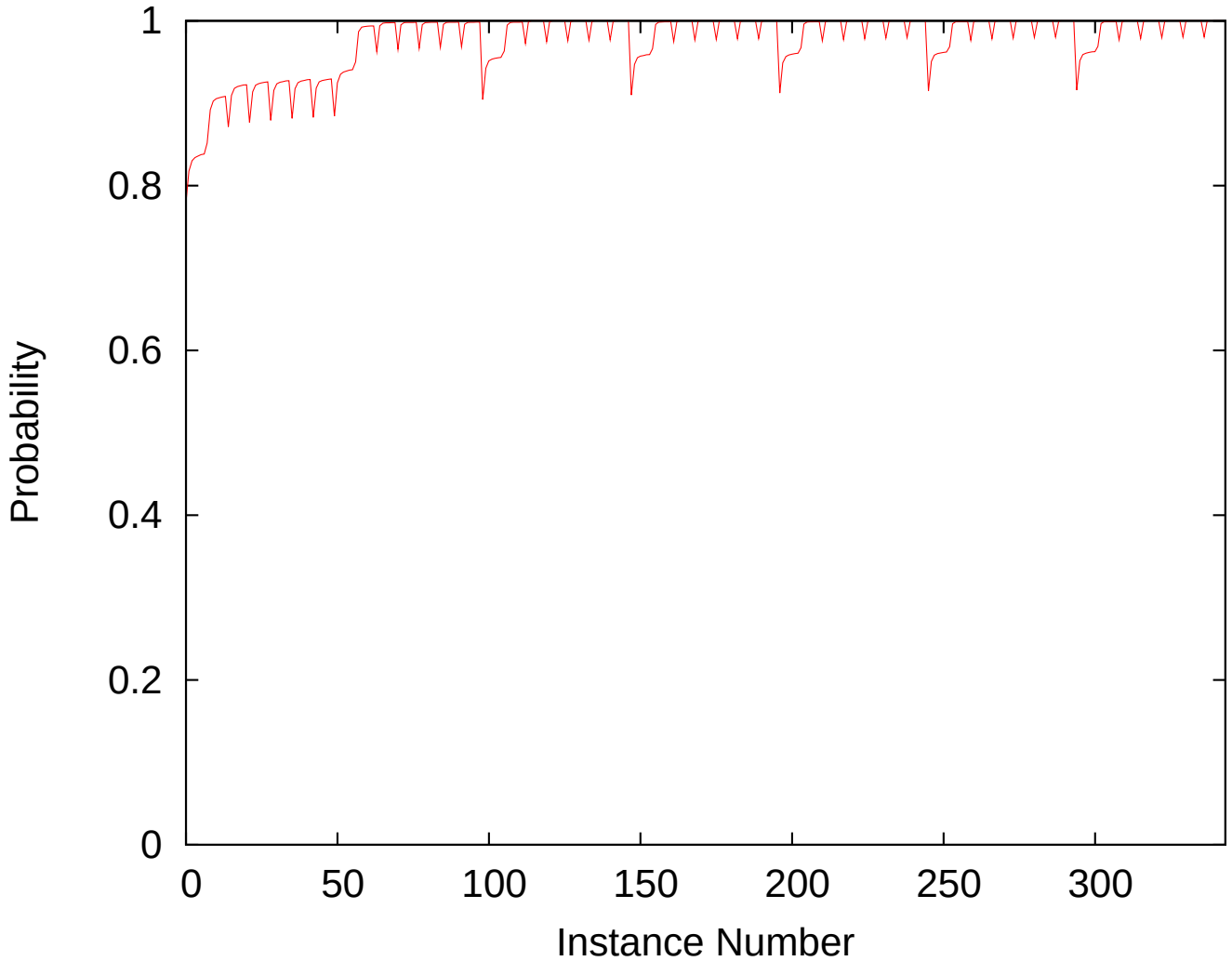
Response-time analysis at time 135



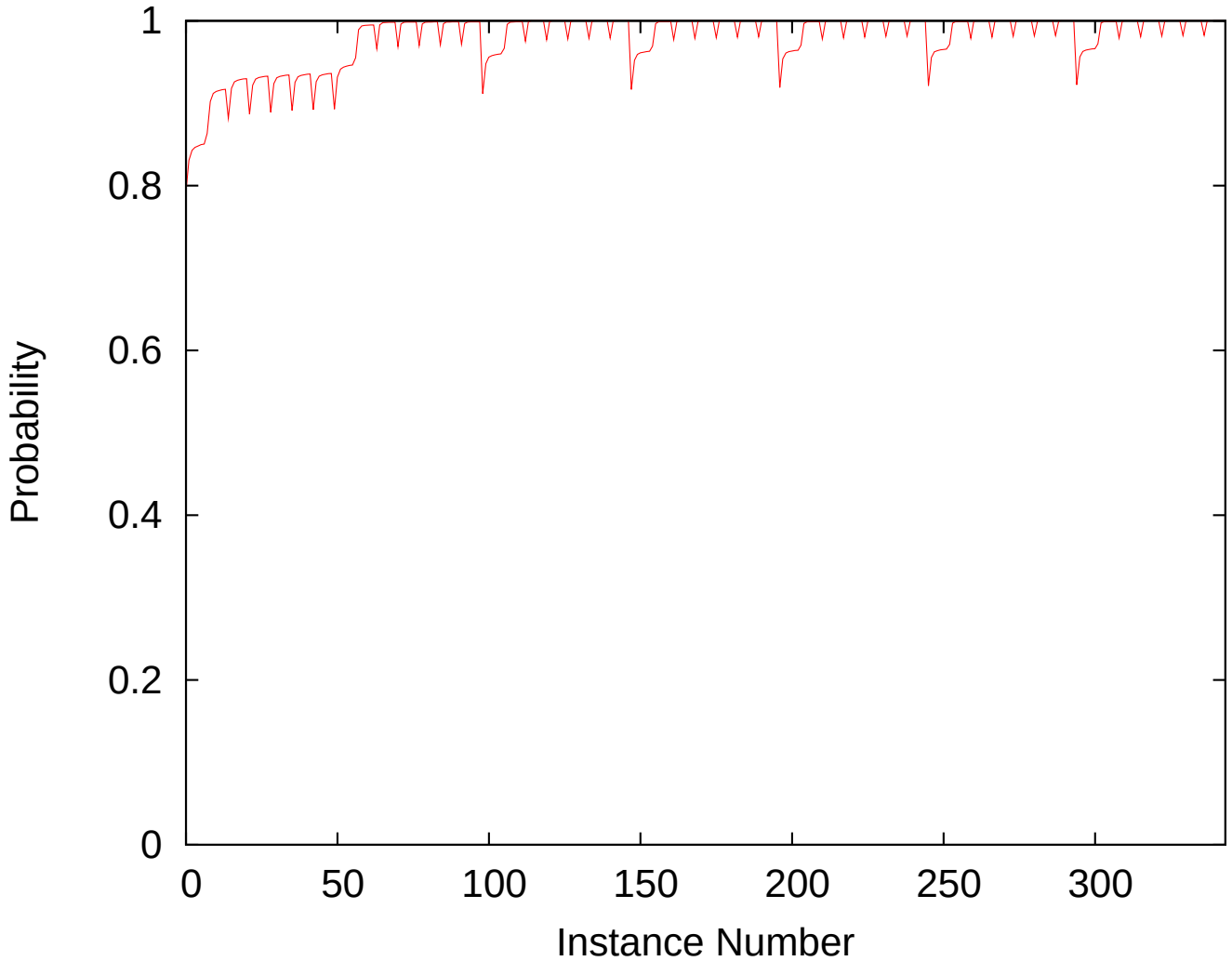
Response-time analysis at time 140



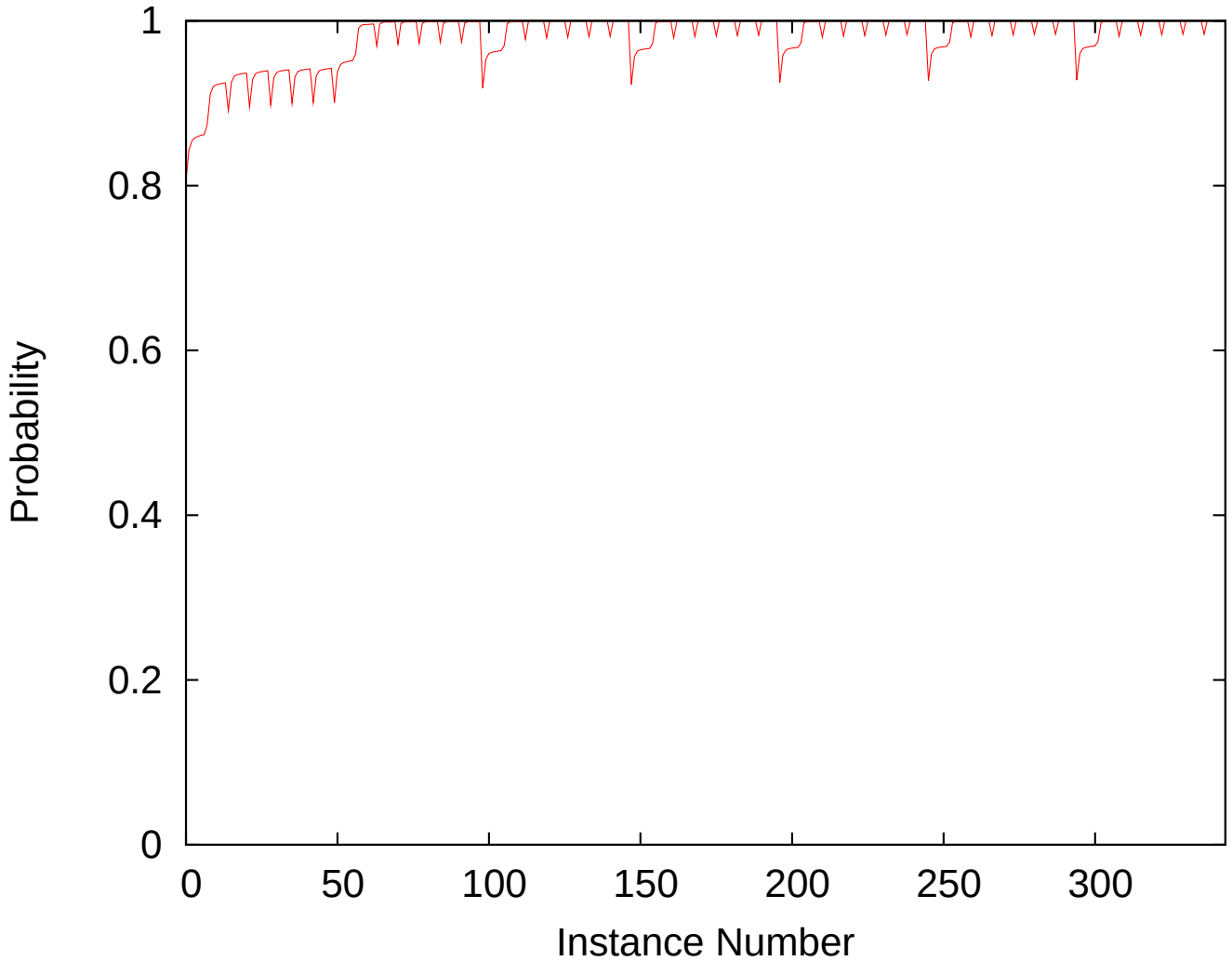
Response-time analysis at time 145



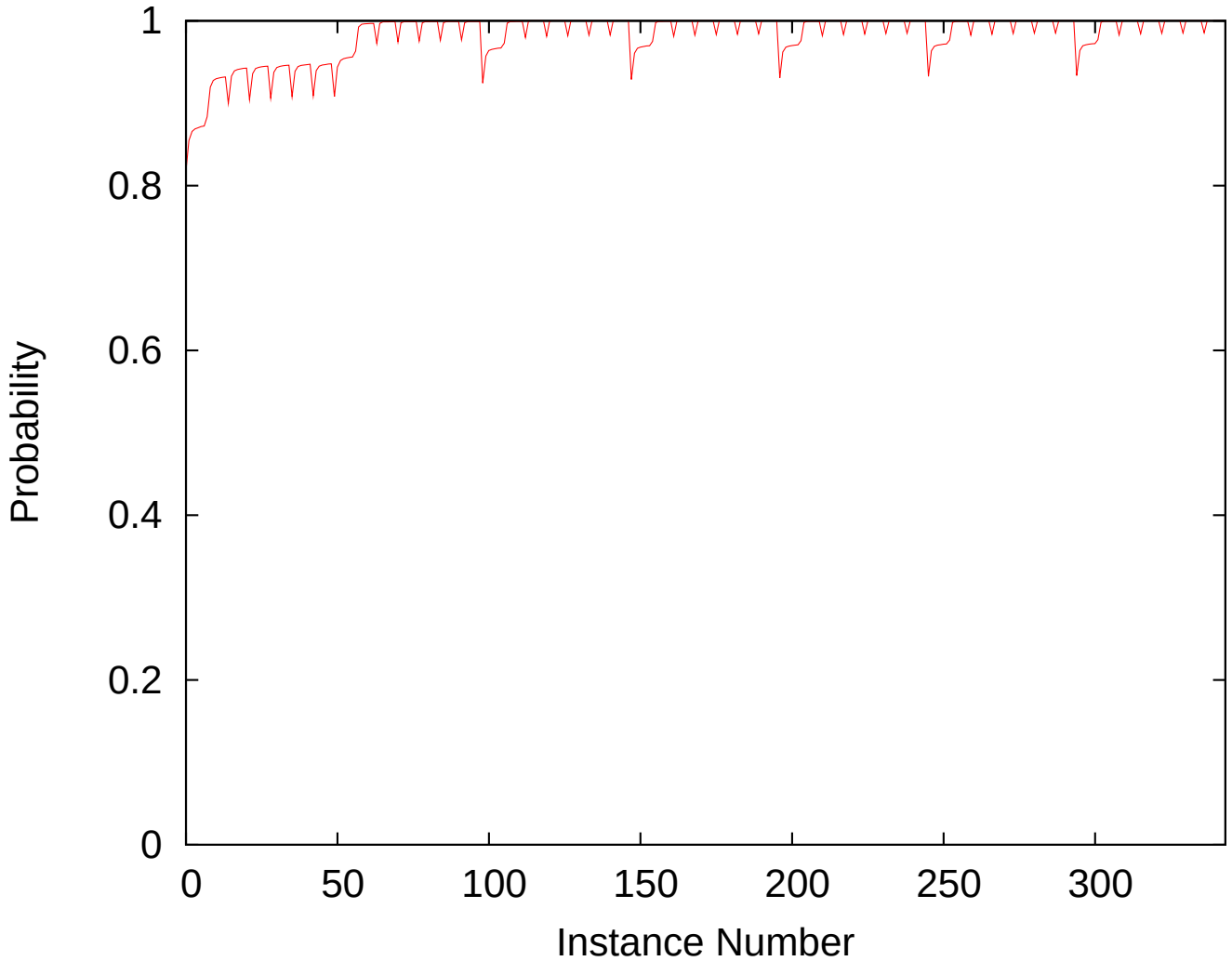
Response-time analysis at time 150



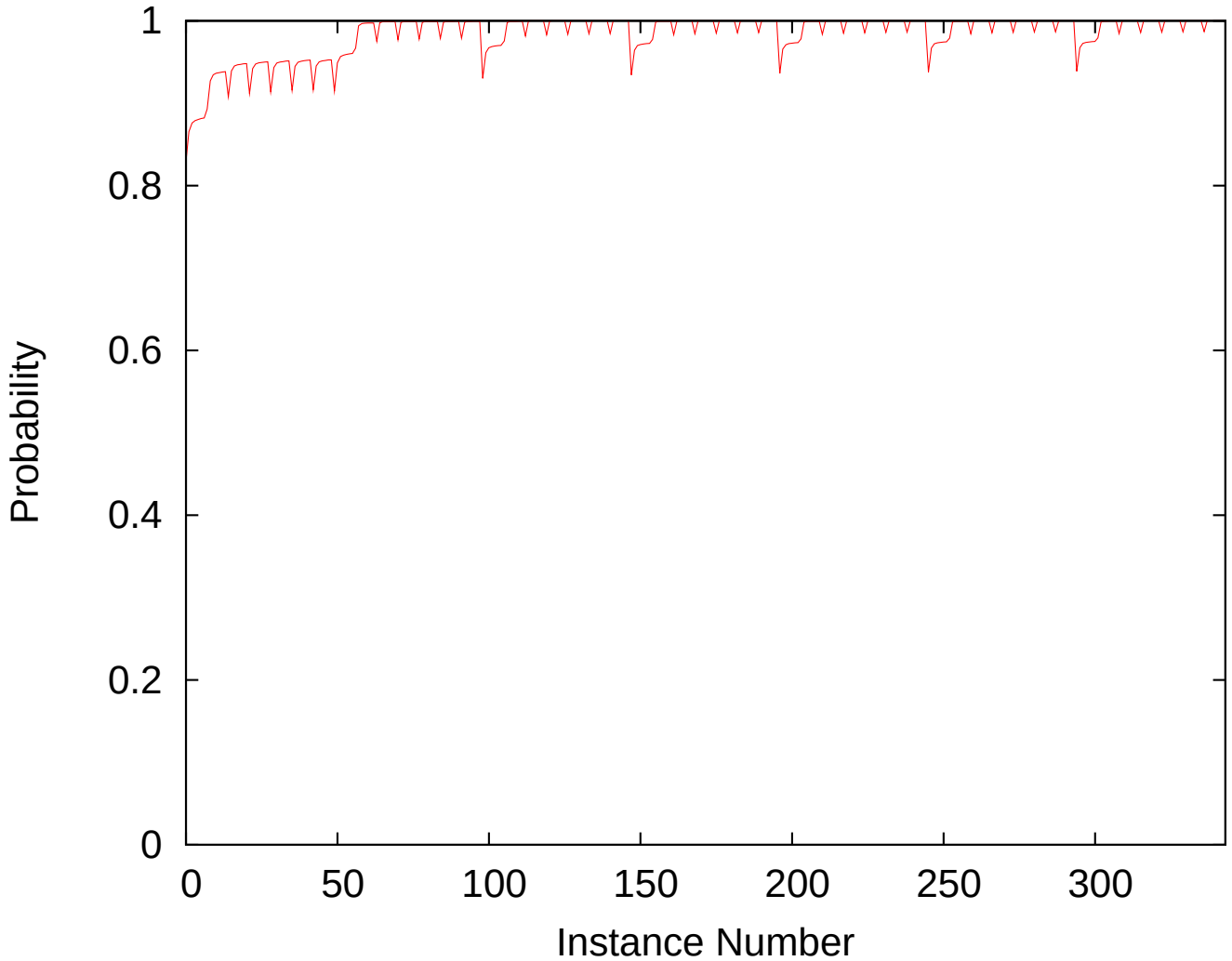
Response-time analysis at time 155



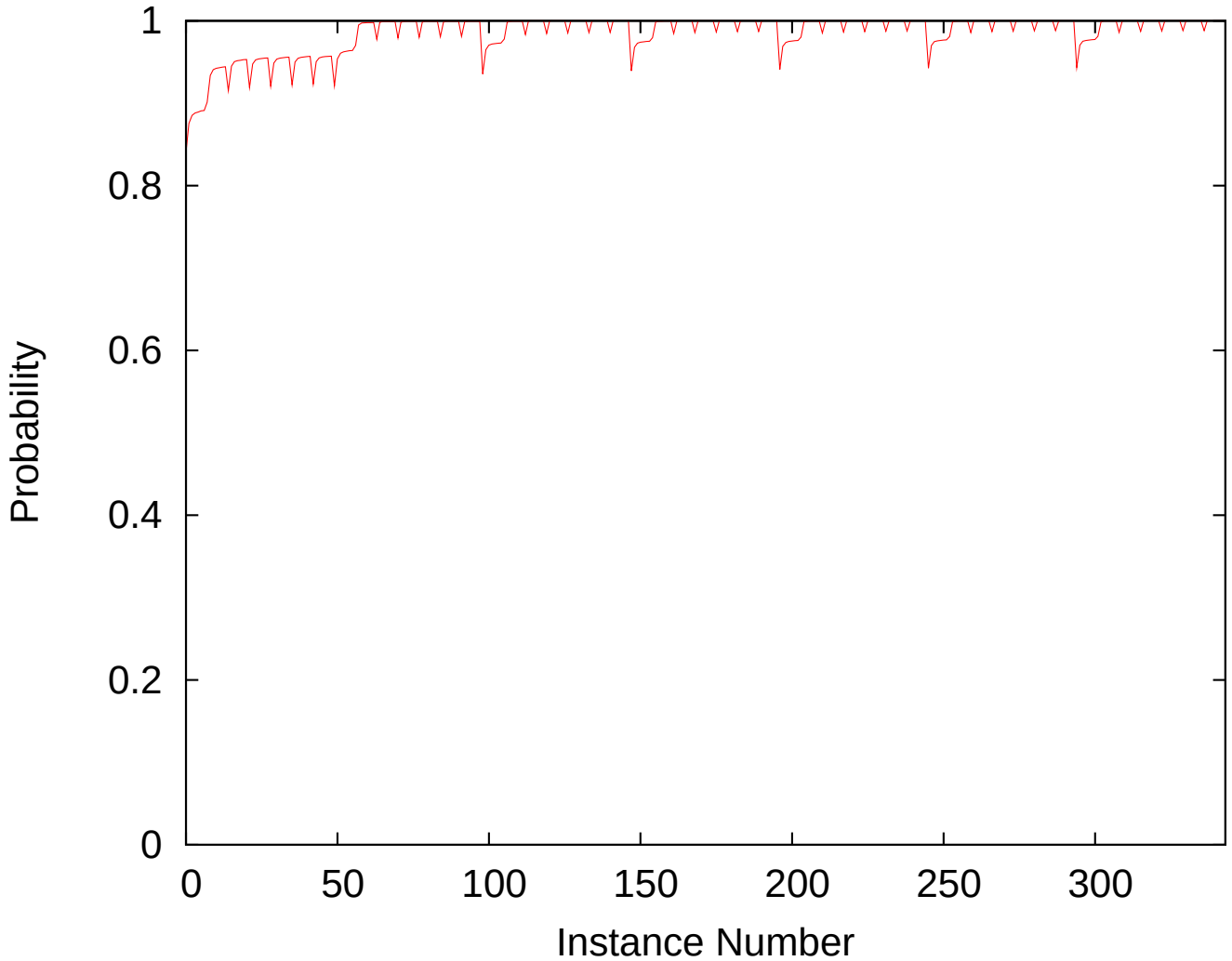
Response-time analysis at time 160



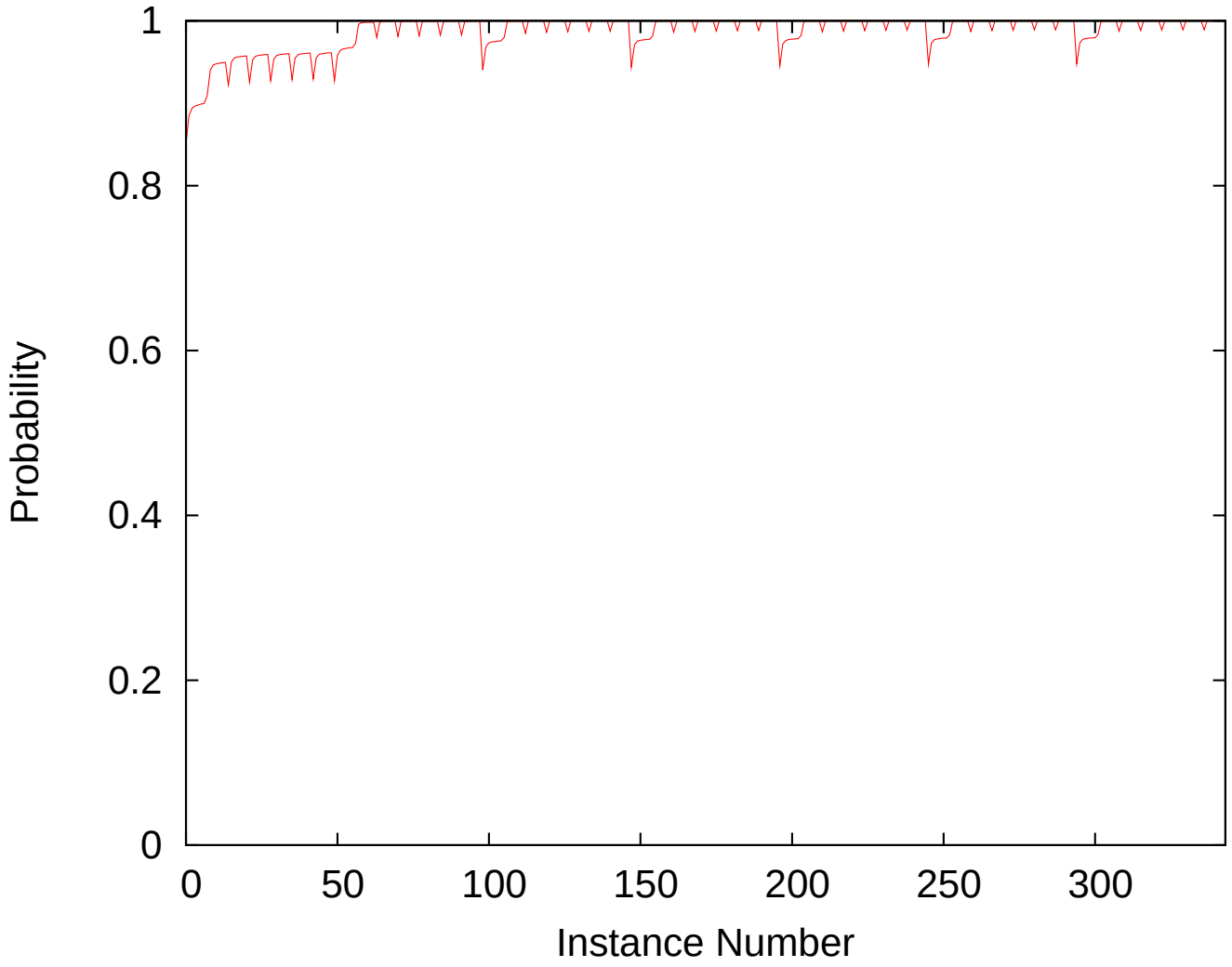
Response-time analysis at time 165



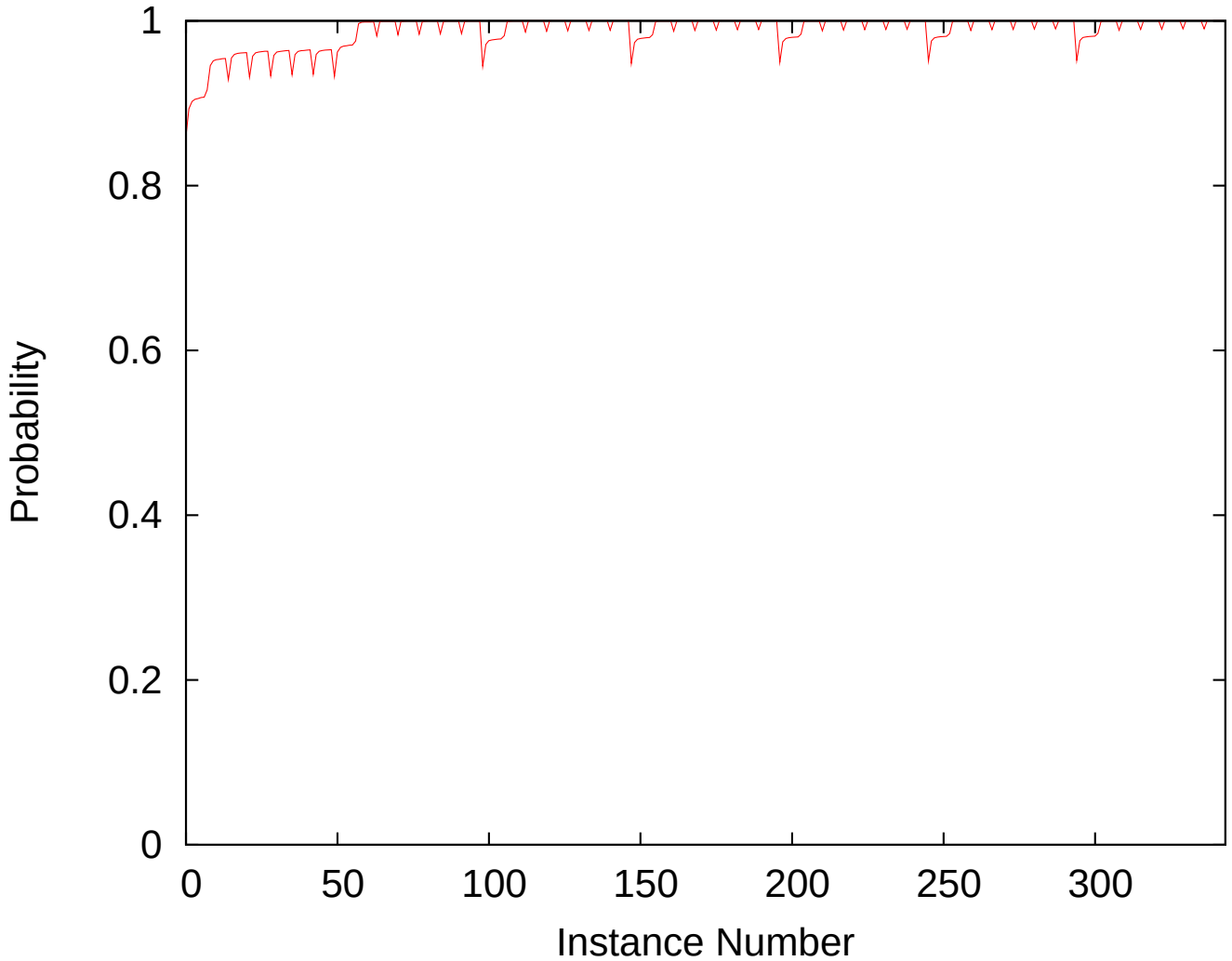
Response-time analysis at time 170



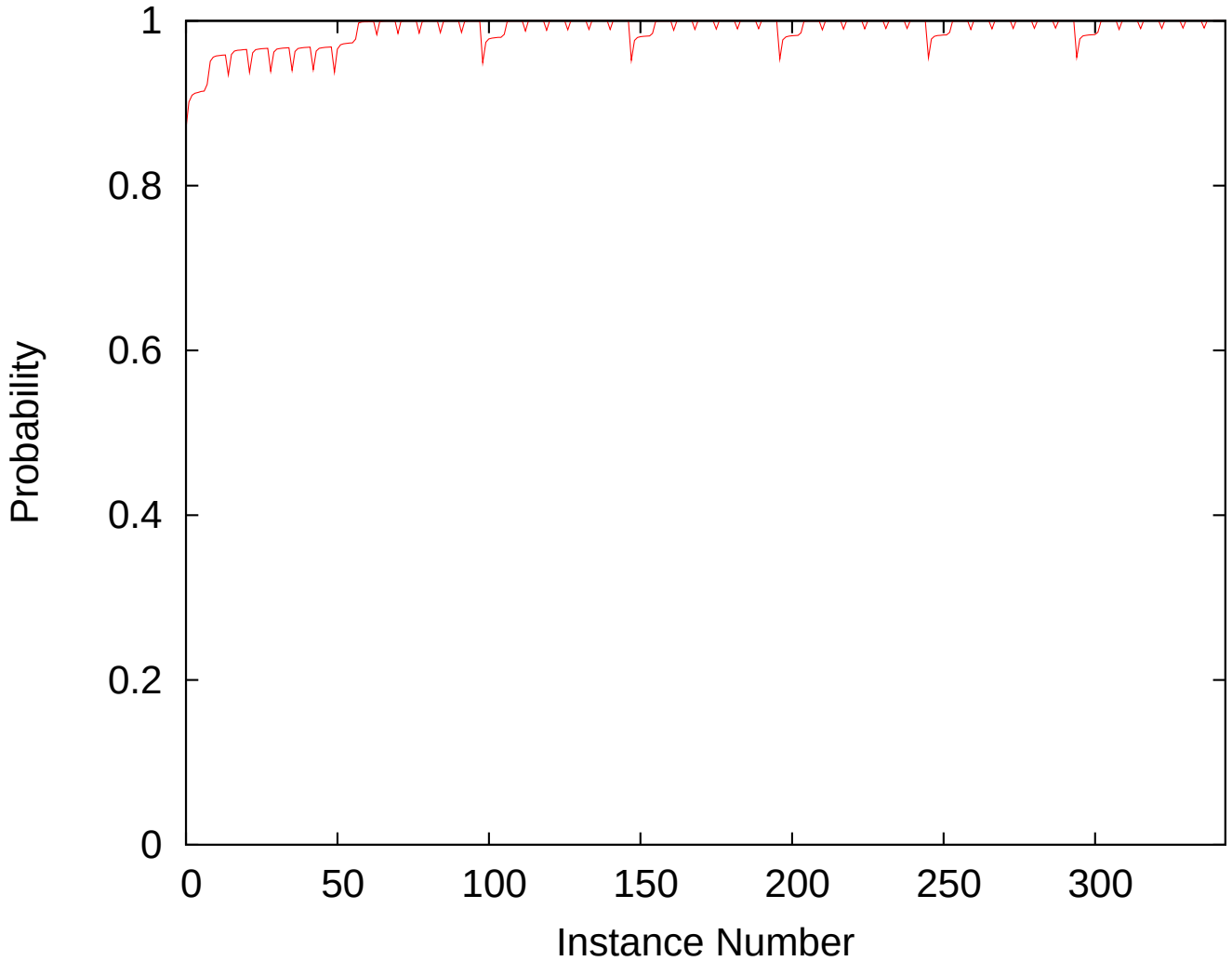
Response-time analysis at time 175



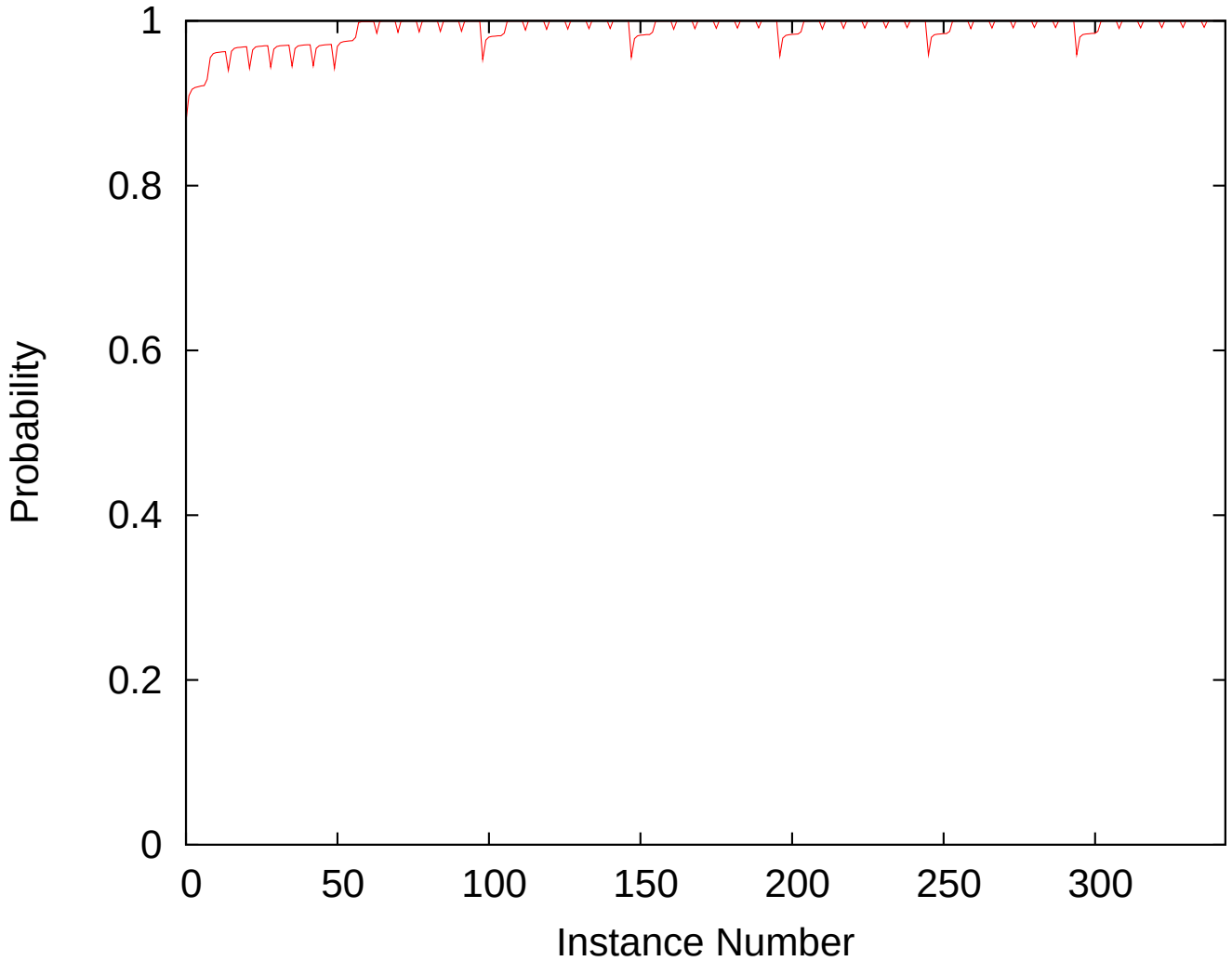
Response-time analysis at time 180



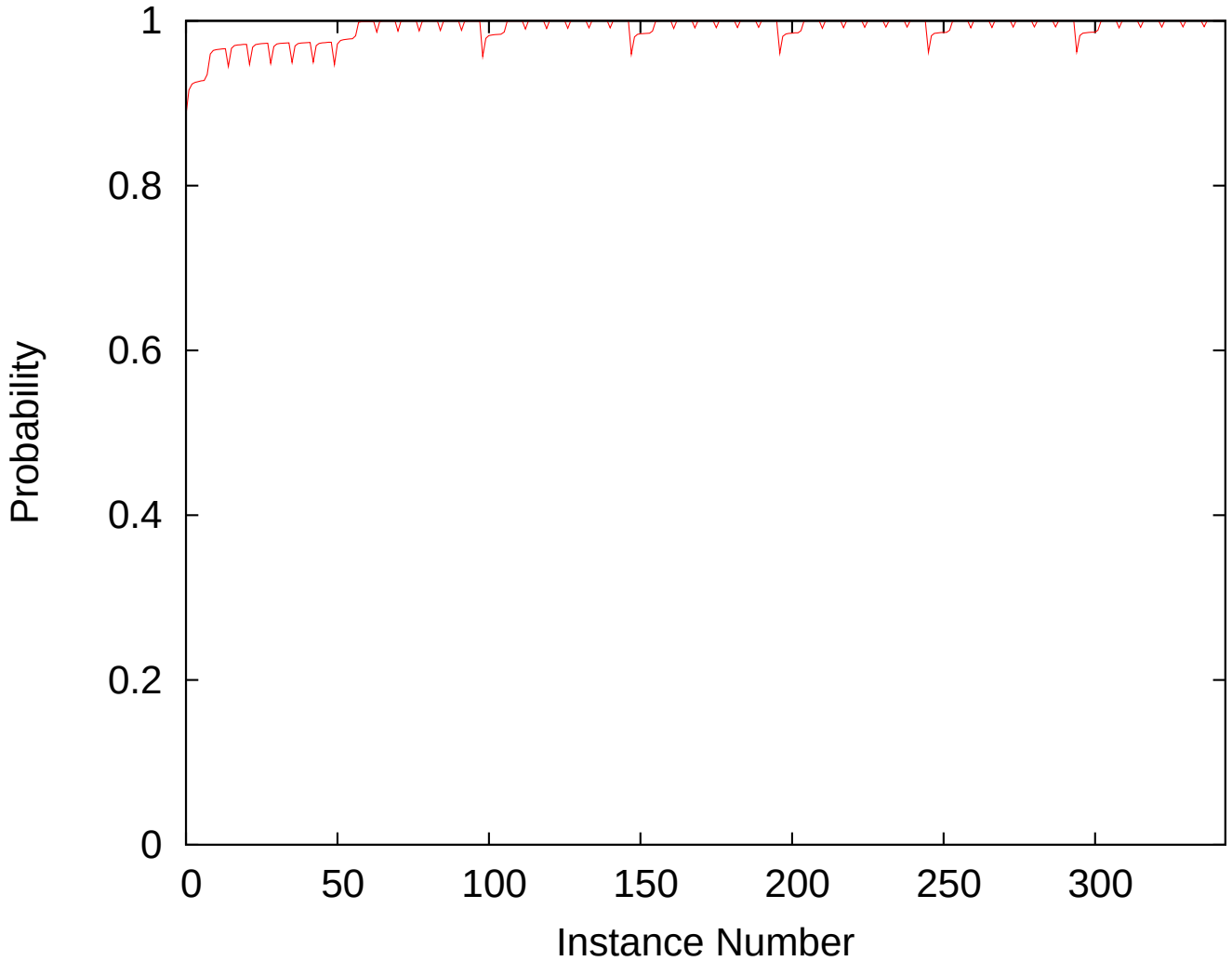
Response-time analysis at time 185



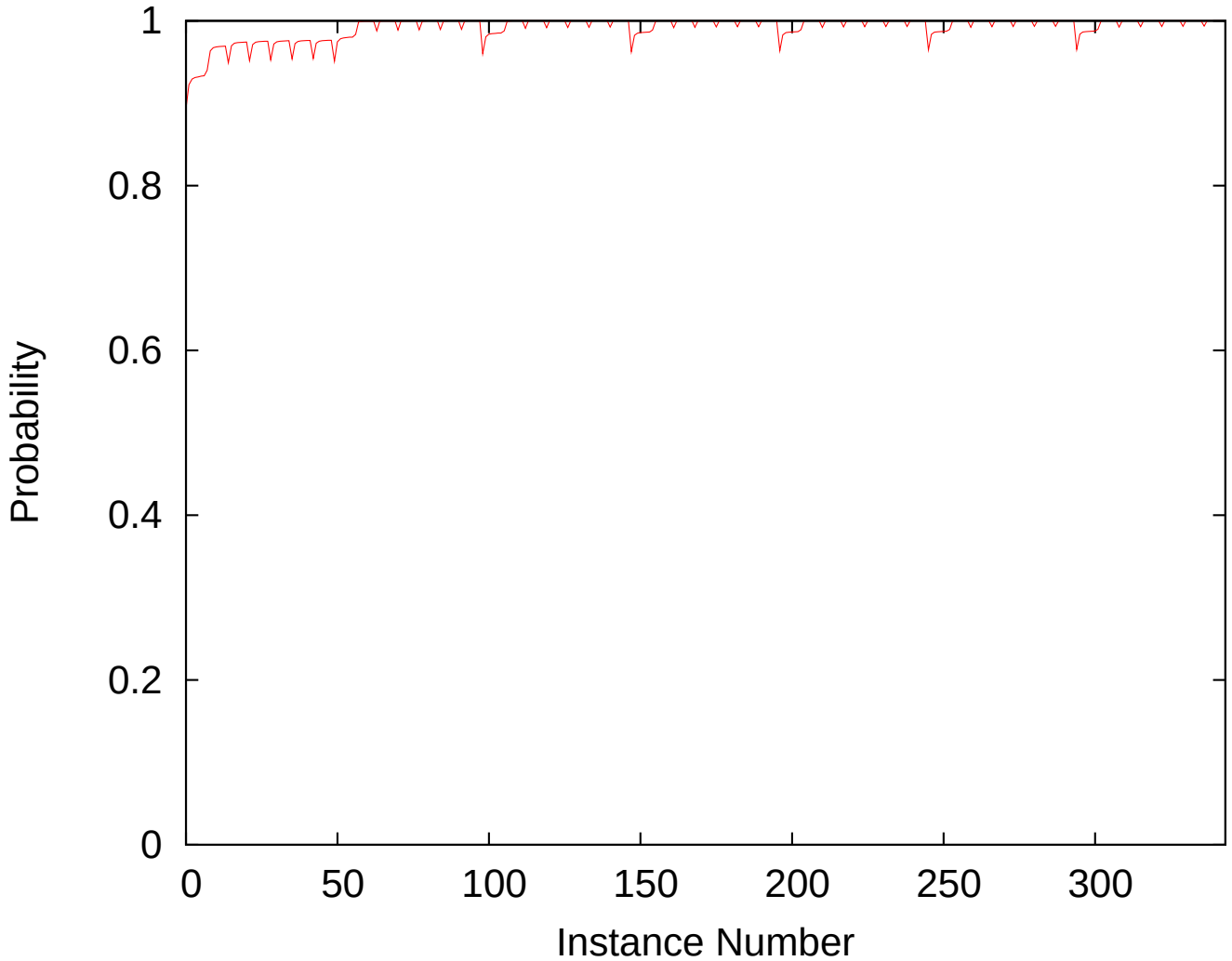
Response-time analysis at time 190



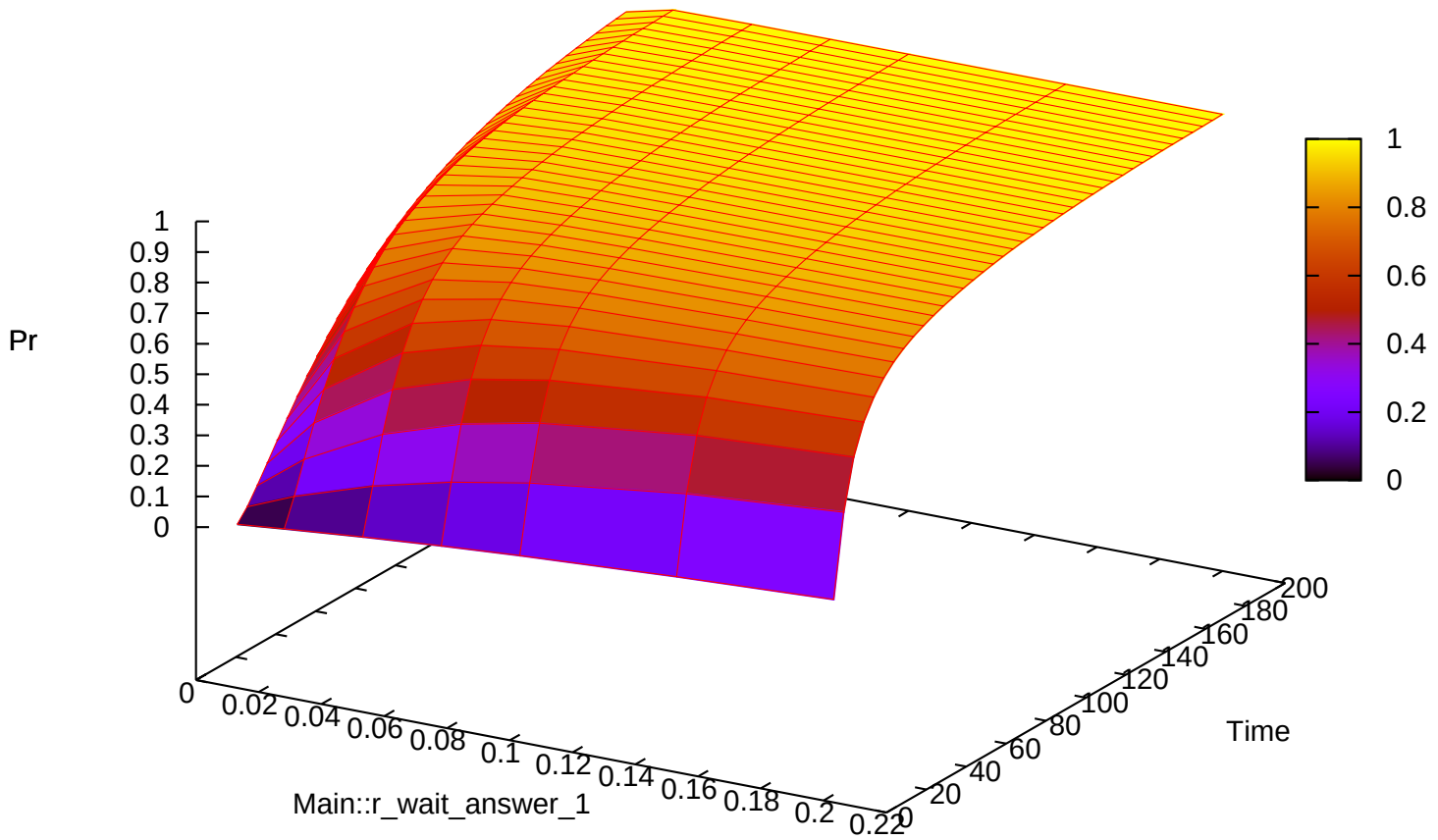
Response-time analysis at time 195



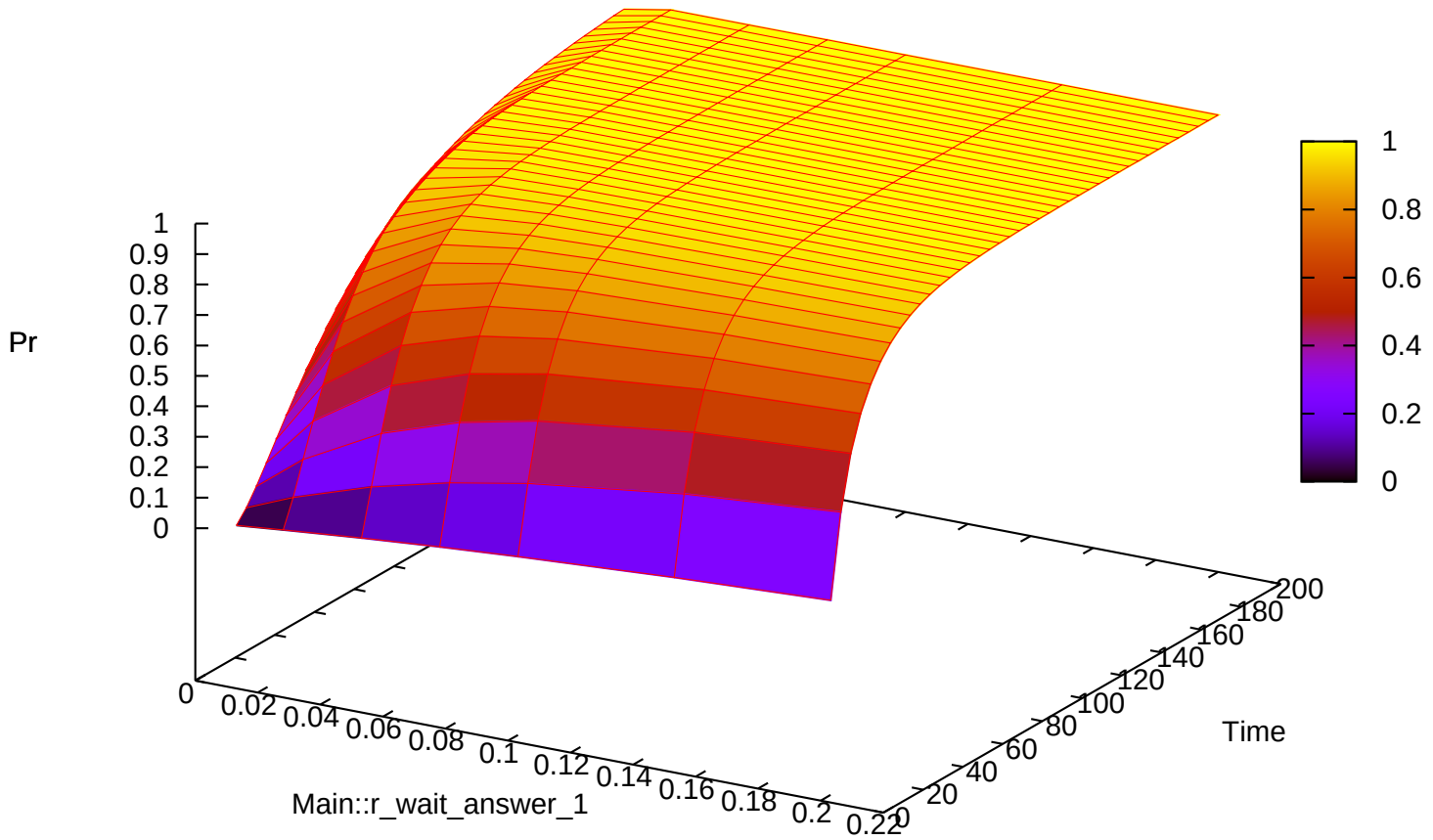
Response-time analysis at time 200



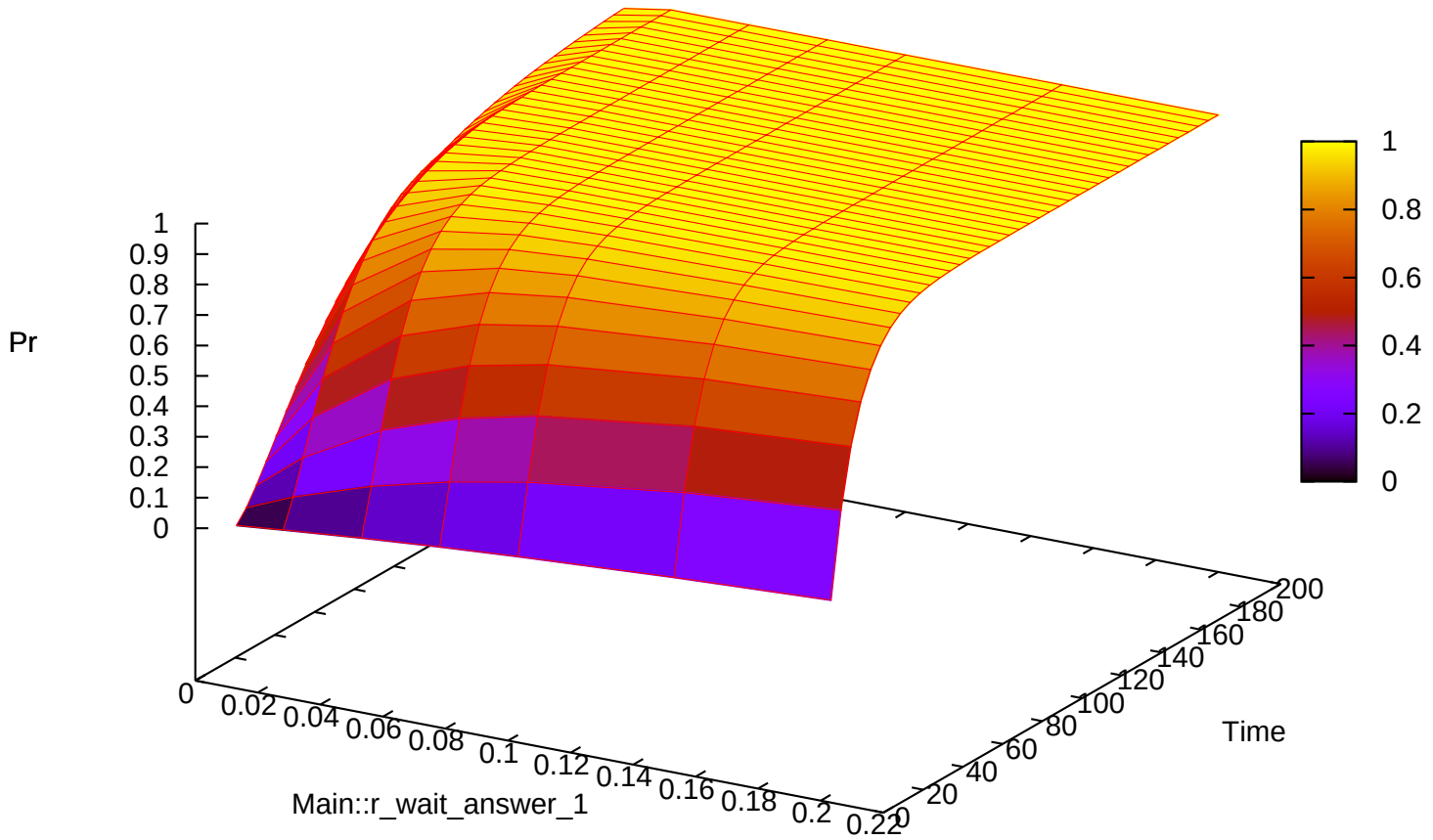
Sensitivity analysis, $r\_wait\_answer\_3 = 0.01$



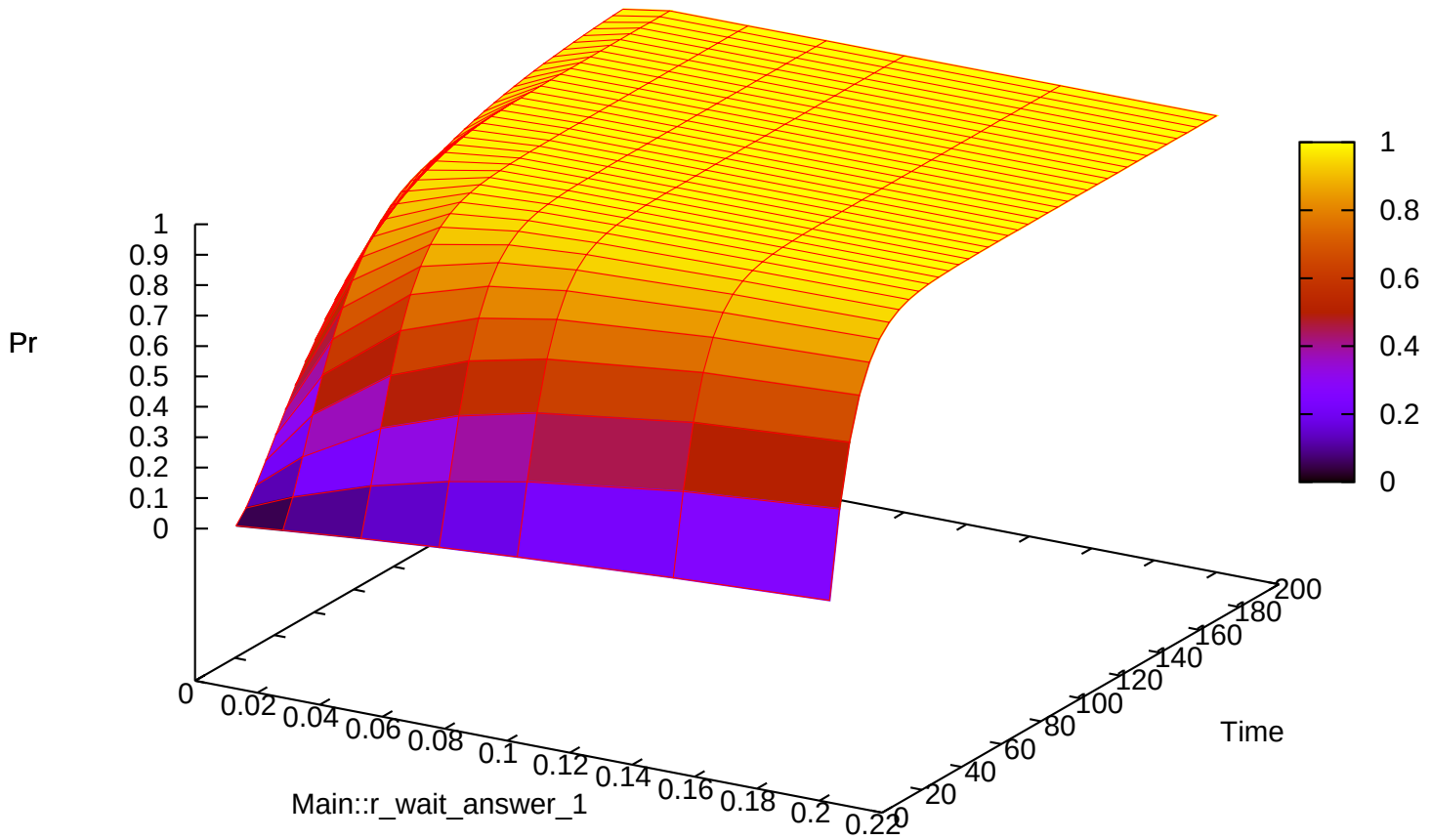
Sensitivity analysis, $r\_wait\_answer\_3 = 0.025$



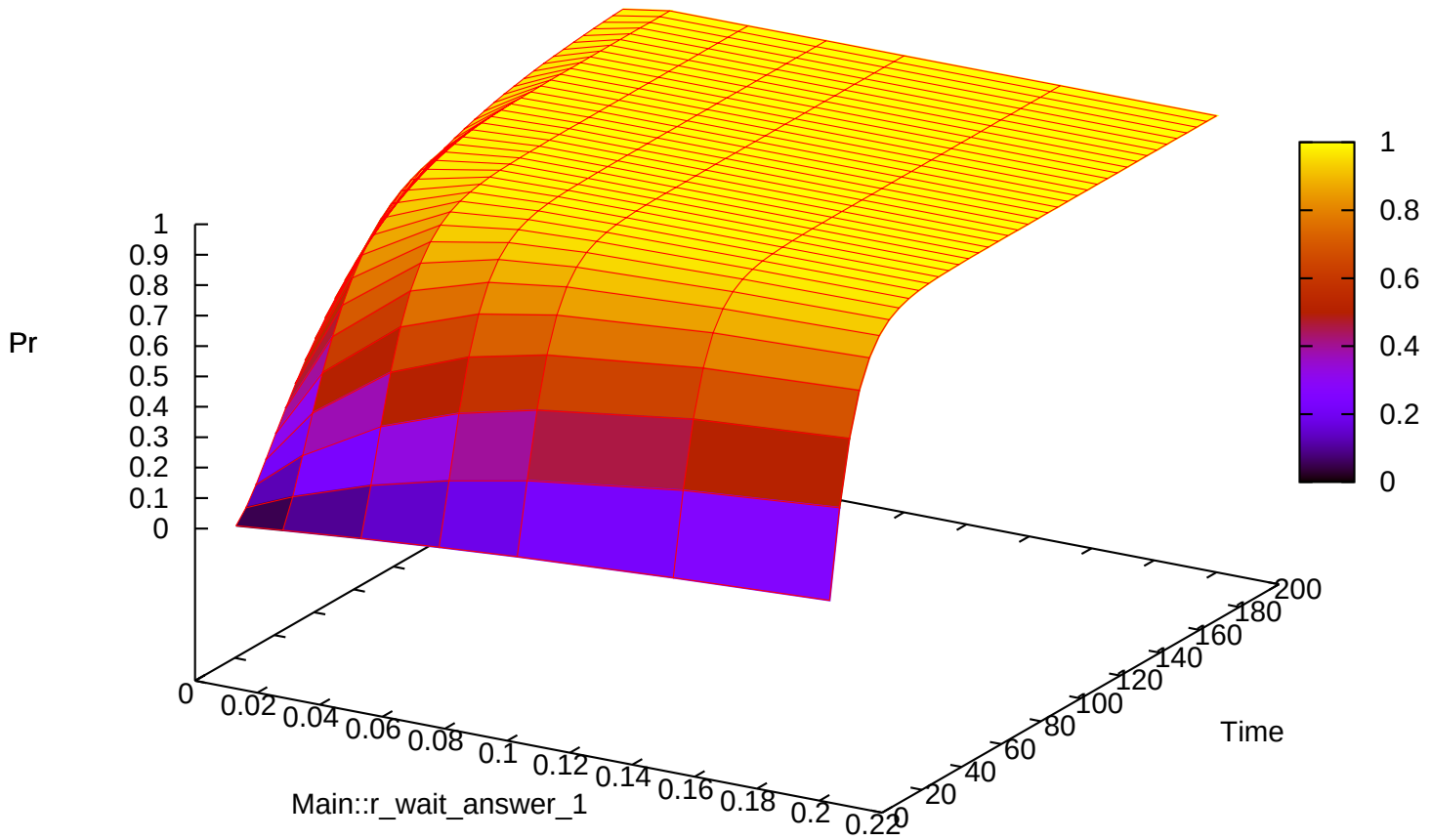
Sensitivity analysis, $r\_wait\_answer\_3 = 0.05$



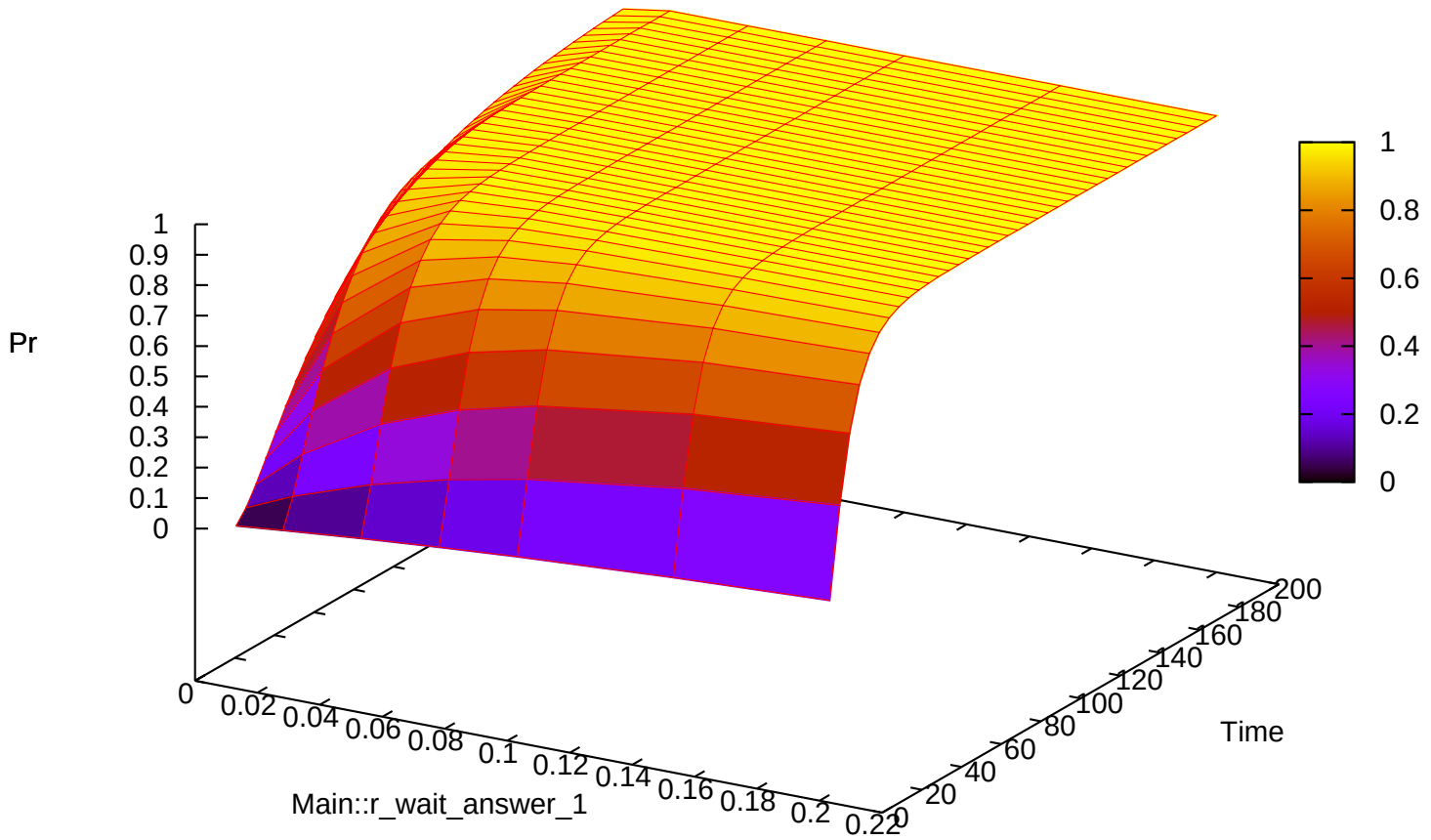
Sensitivity analysis, $r\_wait\_answer\_3 = 0.075$



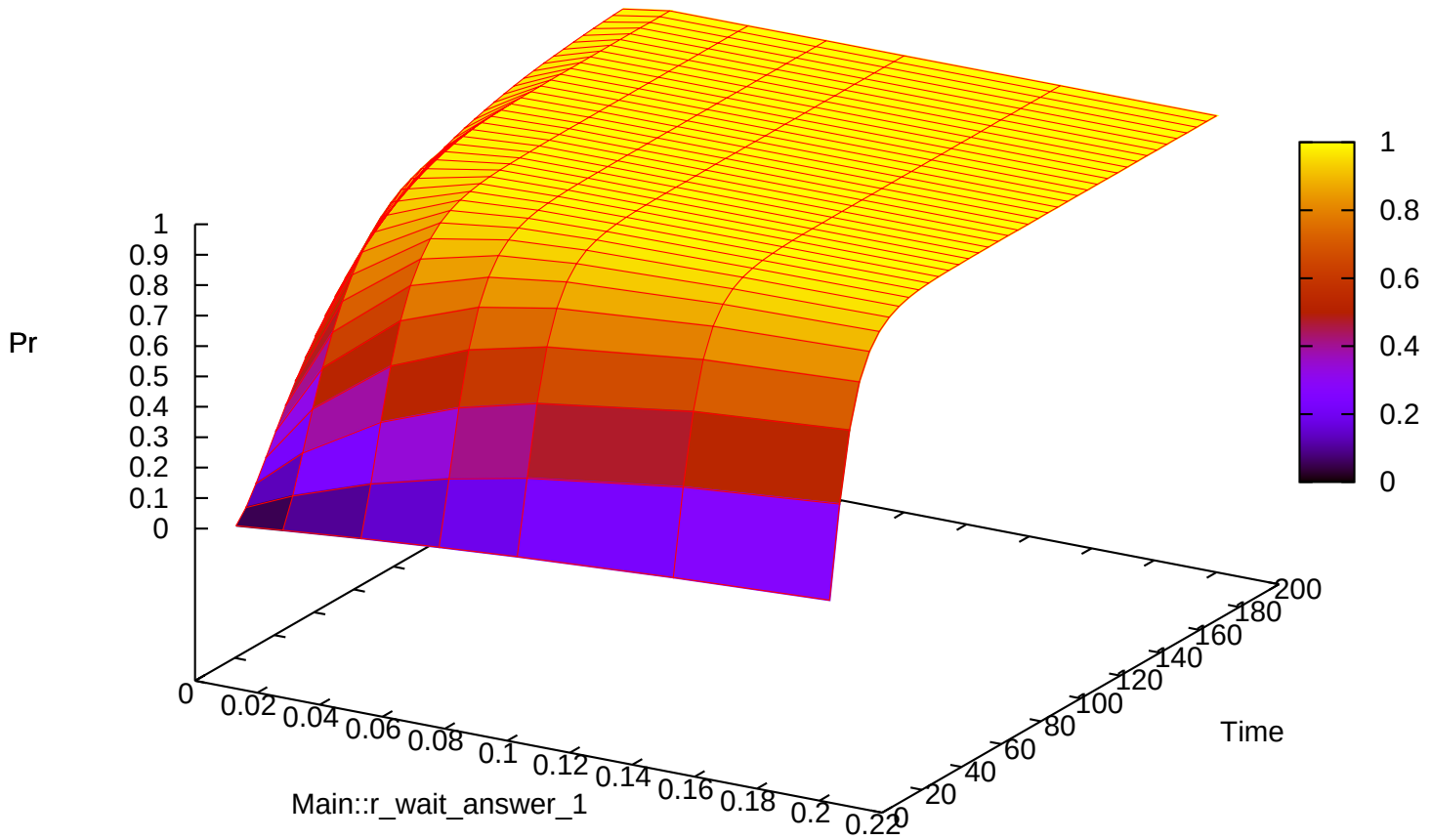
Sensitivity analysis, $r\_wait\_answer\_3 = 0.1$



Sensitivity analysis, $r\_wait\_answer\_3 = 0.15$



Sensitivity analysis, $r\_wait\_answer\_3 = 0.2$



Insight obtained

- Increasing the rate at which the final call is made has a significant impact on the probability of completion of the work of the accident assistance service by a given time bound.

Insight obtained

- Increasing the rate at which the final call is made has a significant impact on the probability of completion of the work of the accident assistance service by a given time bound.
- Here probability of completion rises to near certainty more quickly in more cases.

Insight obtained

- Increasing the rate at which the final call is made has a significant impact on the probability of completion of the work of the accident assistance service by a given time bound.
- Here probability of completion rises to near certainty more quickly in more cases.

Insight obtained

- Increasing the rate at which the final call is made has a significant impact on the probability of completion of the work of the accident assistance service by a given time bound.
- Here probability of completion rises to near certainty more quickly in more cases.

This suggests that the service would best improve its response time by waiting for less time on the second call attempt than on the first, and less time on the third than the second.

Conclusions

- We completed a thorough analysis of the accident assistance service.

Conclusions

- We completed a thorough analysis of the accident assistance service.
- Model-checking allowed us to detect errors in the model to ensure accuracy of the modelling work.

Conclusions

- We completed a thorough analysis of the accident assistance service.
- Model-checking allowed us to detect errors in the model to ensure accuracy of the modelling work.
- Passage-time and sensitivity analysis allowed us to identify the most productive place in the workflow to target for improvement.

Conclusions

- We completed a thorough analysis of the accident assistance service.
- Model-checking allowed us to detect errors in the model to ensure accuracy of the modelling work.
- Passage-time and sensitivity analysis allowed us to identify the most productive place in the workflow to target for improvement.
- Modelling tools hosted on a state-of-the-art modelling environment.