Too Big To Fail: Model Implementation Based On Repeated Game

Master thesis in Computer Science

submitted by

Igor Bozic

Presented to the

Department of Informatics and

Department of Banking and Finance

of the University of Zurich

Prof. Dr. L. Hilty
Prof. Dr. U. Birchler

Overview

- Motivation
- Related Work
- Actual Problems and Approach
- Web Application
- Models and Games
- Results
- Requirements
- Discussion
- Lessons Learned
- Questions

Motivation

- Topic: Too Big To Fail
- Learning effect
- Solution from previous work
  - Models implemented in Matlab
  - Not available for everyone (Code and Matlab needed)
  - Change of input values needs to be done directly in Matlab-Code

Related Work

- “Never again!” – the dynamics of bank bailouts (Prof. Dr. U. Birchler)
- Modeling bank bailout decisions by the state
- Models implemented in Matlab
- Graphical (Numerical) output of the results (pdf-files)

Actual Problems and Approach

- Problem: Change of input needs a change in Matlab-Code
- Problem: Not able to play from a state/bank view
- Problem: Not available for everyone
- New experimental environment in form of a web-application
  - Available for everyone
  - Model implementation
  - Game design
  - Extensible to other economic problems
  - Additional informations for users

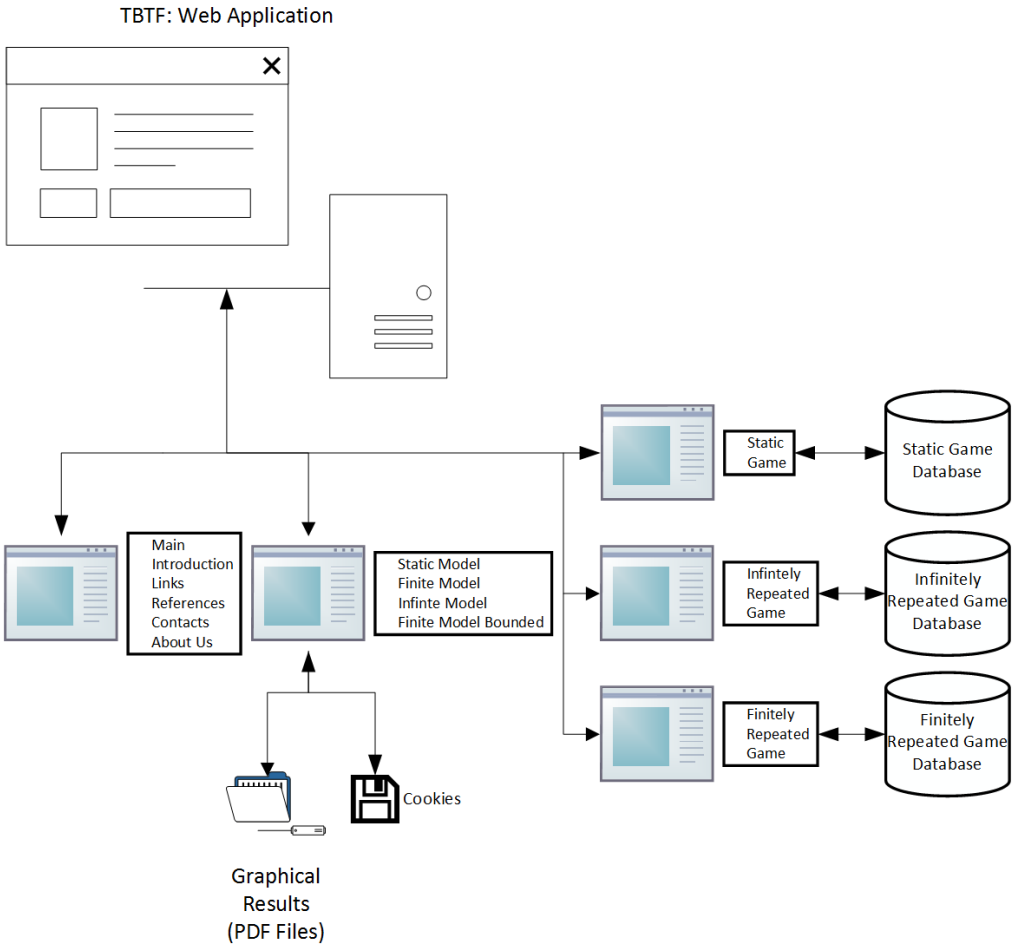
Web Application: Analysis

- Acquisition of knowledge
  - Too Big To Fail
  - Game Theory
- Frameworks and tools
  - Matlab (License and costs)
  - R – Studio Shiny (Limited number of forms)
  - Mathematica (CDF Player needed)
  - Python
- Requirements specification
- Web application development and model implementation

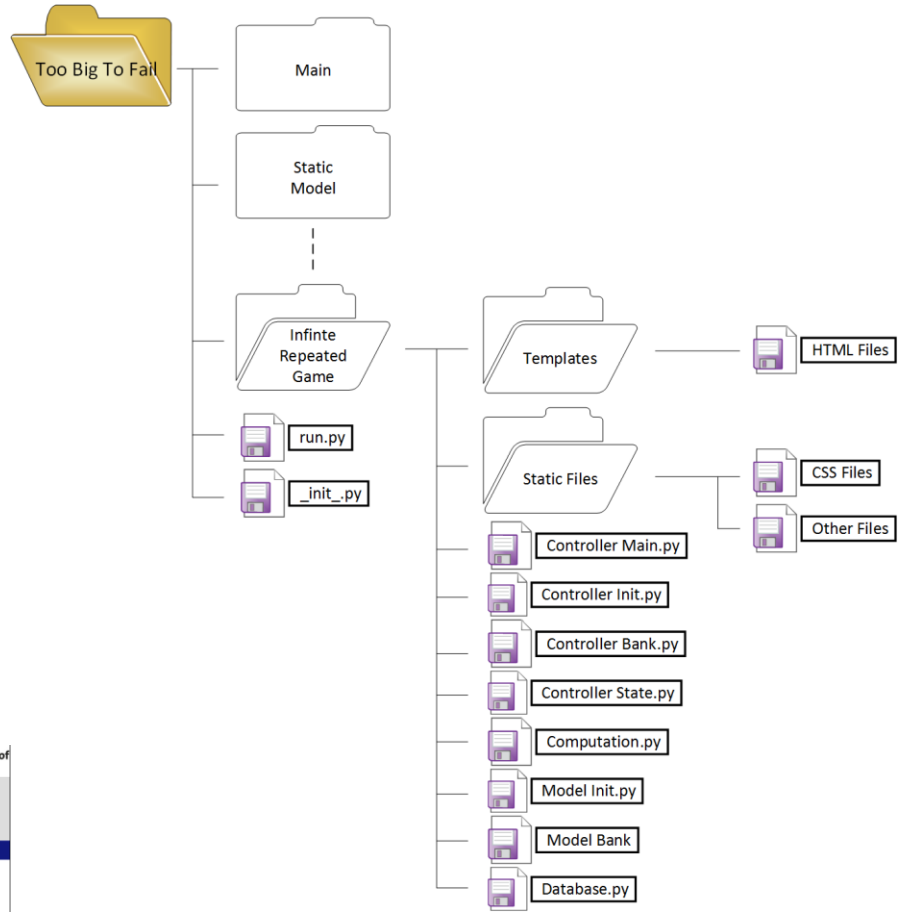
Web Application: Technologies and Frameworks

- Python
- IDE: Anaconda
- IDE: PyCharm
- Macro-framework: Django
- Micro-framework: Flask
- Data storage (Database, Cookies)
- Folder structure (functional, divisional)
- HTML
- CSS

Web Application: Overall Structure



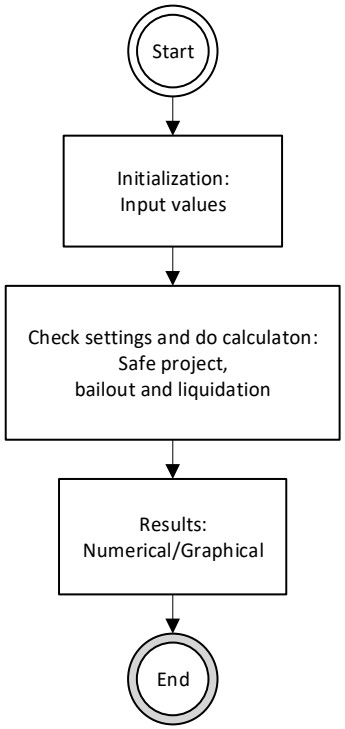
Web Application: Functional Folder Structure



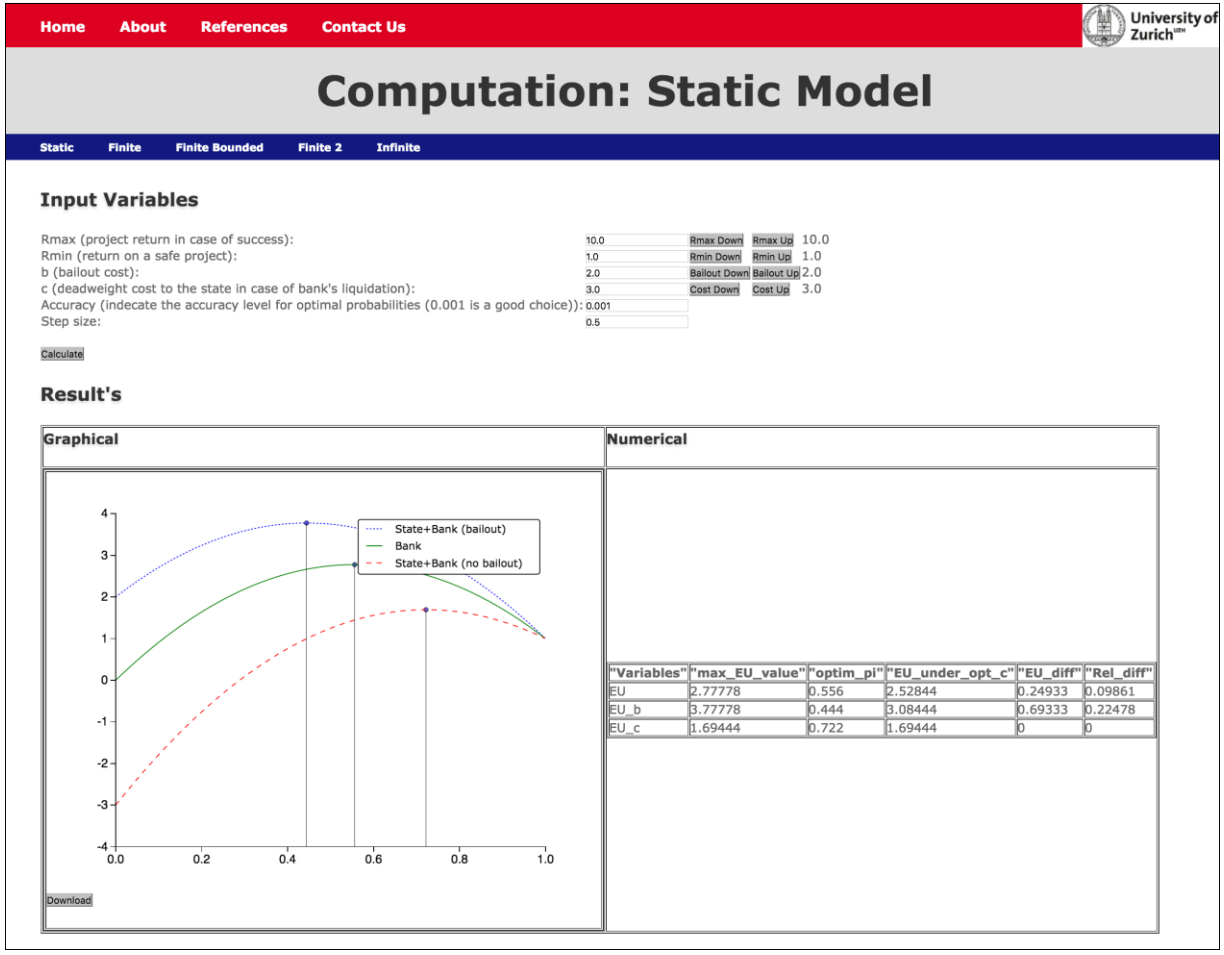
Models

- Static
- Finite
- Finite Bounded
- Infinite

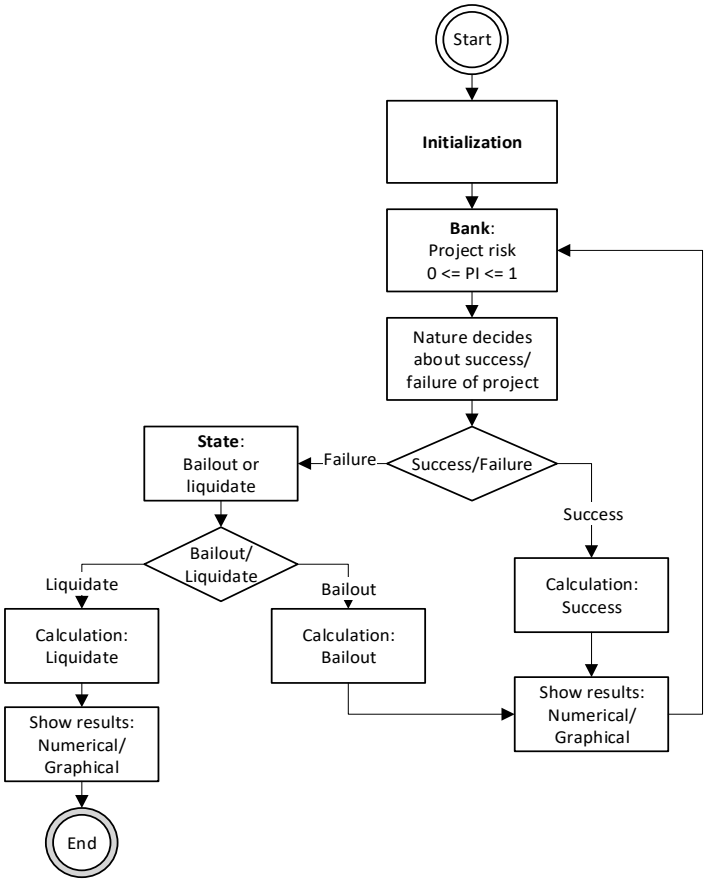
- Matlab –> Python



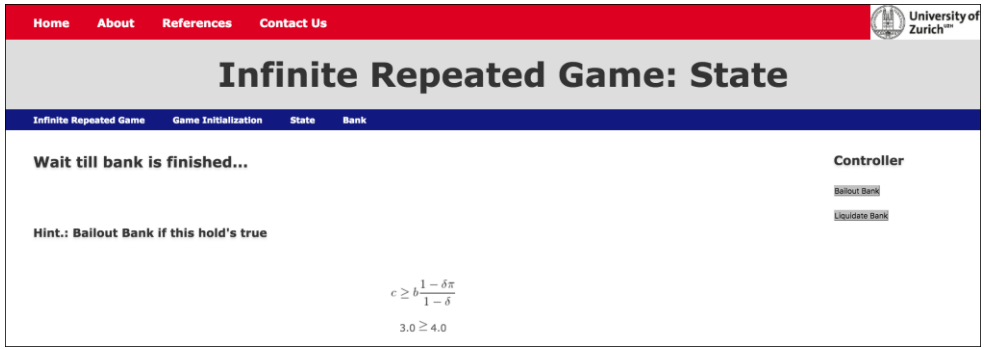
Models: Static



Game: Infinitely Repeated Game



Game: Infinitely Repeated Game



Results: Infinitely Repeated Game

Home About References Contact Us

University of Zurich

Infinite Repeated Game: Initialization

Infinite Repeated Game | Game Initialization | State | Bank

Bank succeded: Start the next round...

Controller

Start Game

New Game

Input Variables

Rmax (project return in case of success):	10.0
Rmin (return on a safe project):	1.0
b (bailout cost):	2.0
c (deadweight cost to the state in case of bank's liquidation):	3.0
discount rate (delta):	0.5
Accuracy (indecate the accuracy level for optimal probabilities (0.001 is a good choice)):	0.001

Game Results

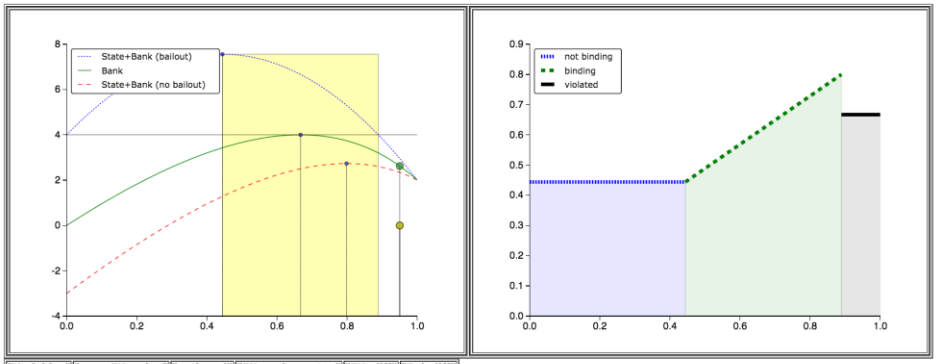
Actual Game Results

"Game round"	V^S	V^B	$E(R)$	$R(\pi)$	R_{max}	R_{min}	b	c	π	$\bar{\pi}$	"Bailout"	"Liquidate"	"Total Nr. of Bailouts"
2.0	0.0	0.0	2.62381	1.45	10.0	1.0	2.0	3.0	0.95	0.5	No	No	0.0

Previous Game Results

"Game round"	V^S	V^B	$E(R)$	$R(\pi)$	π	"Bailout"
0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	3.10909	1.9	0.9	0.0
2.0	0.0	0.0	2.62381	1.45	0.95	0.0

Theoretical Model Result's: Graphical and Numerical



"Variables"	"max_EU_value"	"optim_pi"	"EU_under_opt_c"	"EU_diff"	"Rel_diff"
EU	4.0	0.667	3.7417	0.25829	0.06903
EU_b	7.55555	0.444	5.30553	2.25002	0.42409
EU_c	2.73338	0.798	2.73338	0	0

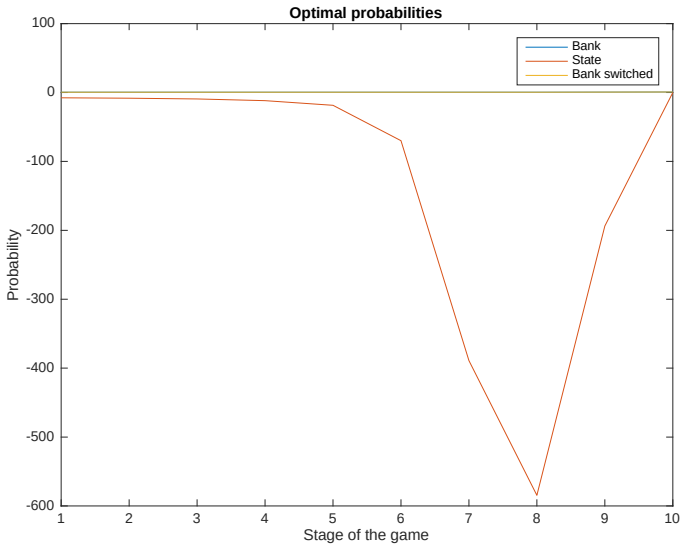
Requirements

Requirement ID	Description
REQ 01:	No user guid needed
REQ 02:	No installation of additional software or specific applications
REQ 03:	Application extensibility
REQ 04:	Model View Controller (MVC)
REQ 05:	Graphical User Interface (GUI)
REQ 06:	Function Implementation
REQ 07:	Implementation: Static Model
REQ 08:	Implementation: Finite Model
REQ 09:	Implementation: Finite Bounded Model
REQ 10:	Implementation: Infinite Model
REQ 11:	Modelling: Static Game
REQ 12:	Modelling: Infinitely Repeated Game
REQ 13:	Modelling: Finitely Repeated Game
REQ 14:	Export results into PDF format

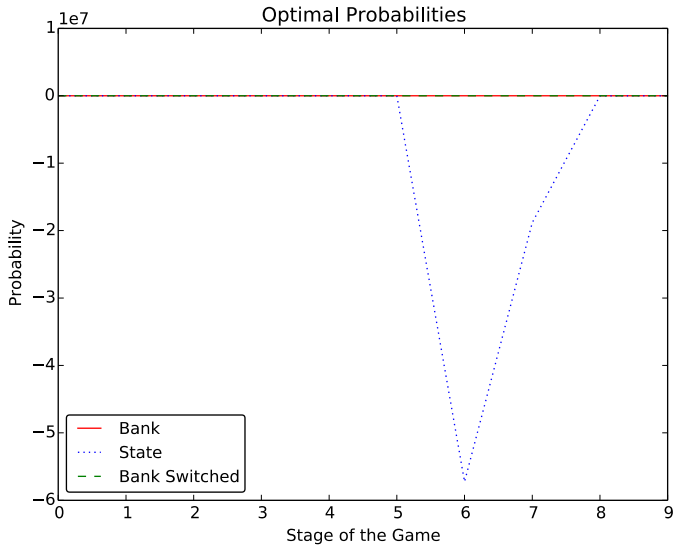
Discussion

- Solver
  - f(x) = 0
  - Matlab: Levenberg-marquard (non-linear least squares)
  - Python: fsolve (solution of system of non-linear equations)

Matlab



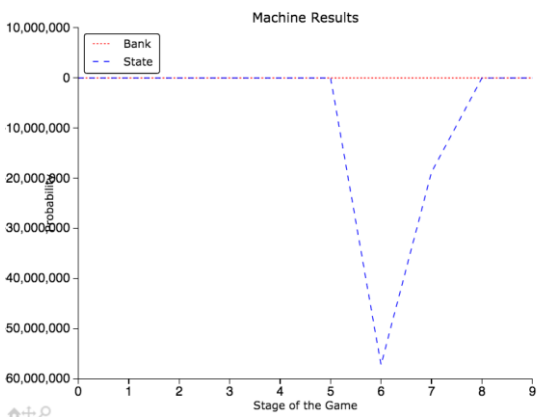
Python

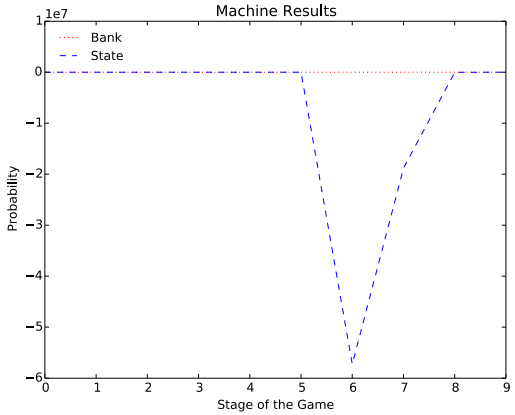


Discussion

- Graphical User Interface (GUI)
  - User experience
  - Additional forms
  - Different forms
- Graphical results using mpld3

mpld3 Bringing Matplotlib to the Browser
MPLD3





Lessons Learned

- Python
- HTML
- Database
- Web Application
- Too Big To Fail
- Game Theory
- Requirements/Specification Analysis
- Matlab
- Optimization Methods

Questions