Towards Automatic Decision Support for Bike-Sharing System Design

Maurice ter Beek, Stefania Gnesi, Diego Latella & Mieke Massink

Formal Methods && Tools lab, ISTI–CNR, Pisa, Italy

VERY*SCART
York, UK
8/9/2015

Outline

1. Background
   - Bike-sharing systems (BSS)
   - Software Product Line Engineering (SPLE)
   - On-the-fly mean field model checking
2. Aim: ~~Towards~~ Automatic Decision Support for BSS Design
3. Using variability analysis to (re)configure a BSS
   - Attributed feature model of a BSS
   - ClaferMOO: multi-objective optimisation
4. Using performance analysis to (re)configure a BSS
   - BSS as a Markov population model
   - FlyFast: mean field model checking
5. Conclusions and future work

Bike-Sharing Systems (BSS)

- Popular sustainable transportation in urban environments
- Case study with challenging run-time optimisation problems

- Simple concept: a user arrives at a docking station, pays for a bike, uses it for a while and returns it to a station
- Multiple benefits: reduction of vehicular traffic (congestion), pollution, energy consumption, etc.

- Docking stations distributed over a city, typically close to other public transportation hubs (e.g. subway and tram stations)
- (Subscribed) users may rent an available bike and drop it off at any station in the city
- To improve the efficiency and the user satisfaction of BSS, the load between the different stations may be balanced
  - incentive schemes (rewards) to change the behaviour of users
  - efficient (dynamic) redistribution of bikes between stations

A short history of BSS

DeMaio @ *Journal of Public Transportation*, 2009

- 1st generation free BSS introduced in Amsterdam (*witte fietsen*)
- 2nd generation large-scale BSS launched in Copenhagen (*Bycyklen*)
- 3rd generation technology-based BSS in $>$ 500 cities worldwide (*Bicing* in Barcelona: $\pm 6,000$ bikes and 420 stations; *Vélib'* in Paris: $\pm 20,000$ bikes and $1,800$ stations; largest in Hangzhou: $\pm 50,000$ bikes and $2,000$ stations, one every 100m)
- 4th generation BSS are already being developed, incl. movable and solar-powered stations, electric bikes and mobile (i)phone real-time availability applications

In the context of Quanticol we collaborate with *PisaMo*, an in-house public mobility company of Pisa's administration that introduced the BSS *CicloPi* in Pisa in 2013 (currently ± 150 bikes and 15 stations), and with its supplier *Bicincittà*

BSS as a product line

Finding the right BSS for a particular city poses many questions

- How many and what kind of bikes?
- How many and what kind of stations and where to place them?
- How to avoid stations being completely full/empty for periods?
- Which features (antitheft, maintenance, smart services, etc.)?
- With (what kind) or without dynamic redistribution of bikes?
- Incentives for users to return bikes to less popular stations?
- Costs and charging policy (credit card, keycard, etc.)?

How to evaluate the various options, costs/benefits, improvements and changes in a systematic way?

Aim: decision support for BSS

Provide automatic decision support for the initial design of BSS to be deployed, as well as for successive adaptations and reconfigurations, considering both qualitative and performance aspects

⇒ highly configurable system and collective adaptive system

Two complementary strategies for the evaluation of BSS designs by means of automated tool support:

1. Using variability analysis to (re)configure a BSS
   - CLAFER toolset to perform multi-objective optimisation of attributed feature models known from SPLE
2. Using performance analysis to (re)configure a BSS
   - Mean field model checker FLYFAST to assess performance and user satisfaction aspects of variants of large-scale BSS

(Software) Product Line Engineering

SPLE: develop a product line (family) using a shared architecture or platform (commonalities) and mass customisation (variabilities) to serve, e.g., different markets, thus allowing for (software) reuse

$\Rightarrow$ maximise commonalities whilst minimising cost of variations (i.e. of individual products)

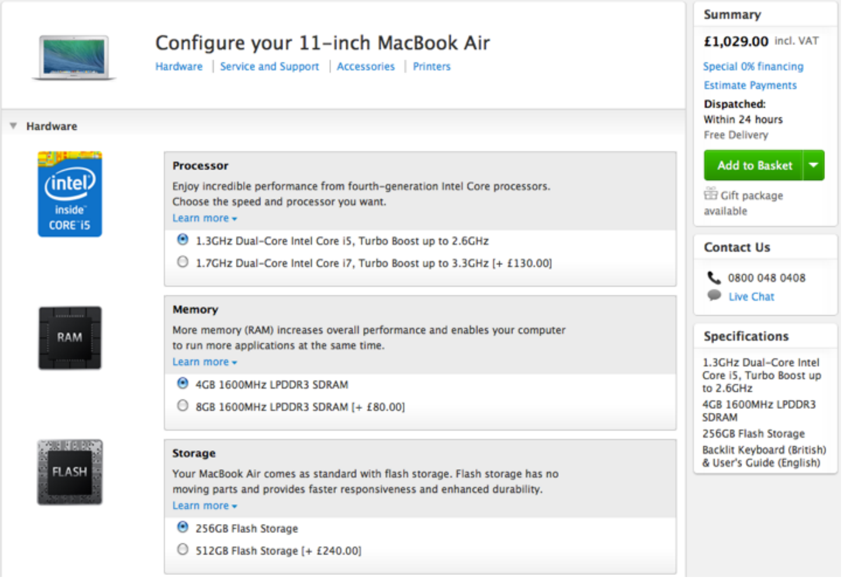
Variability in terms of features:

- stakeholder visible pieces of functionality representing both commonalities (e.g. mandatory, required) and variabilities (e.g. optional, alternative)
- only specific feature combinations concern valid products

"We always have 126,000,000 different bicycles in store! But only the parts for 1,000..."

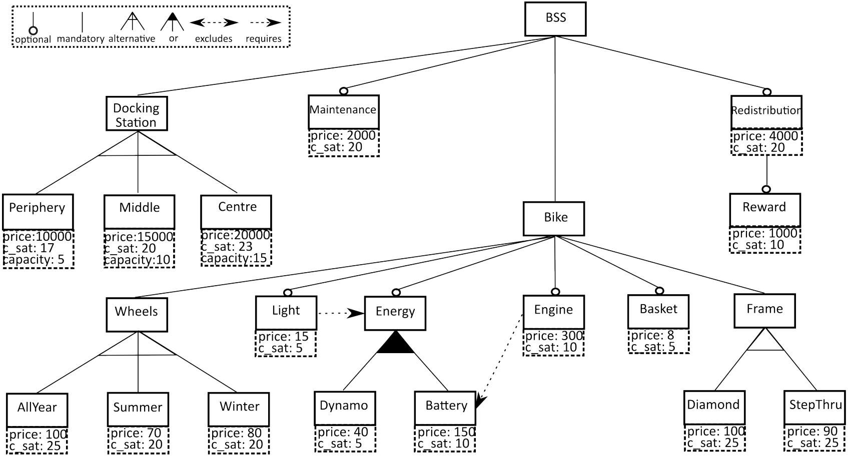
SPL example

www.quanticol.eu



Attributed feature model of a BSS

quanticol
www.quanticol.eu



product: subset of *features* satisfying all variability constraints

$$cost(product) = \sum \{ cost(feature) \mid feature \in product \}$$

CLAFER: lightweight textual SPL modelling language

Bąk, Diskin, Antkiewicz, Czarnecki, Wąsowski @ *Softw. Syst. Model., 2015*

CLAFERMOO: compare different system configurations (variants) w.r.t. various quality dimensions, select the most desirable variant, possibly by resolving trade-offs, and understand the impact of reconfigurations on a variant's quality dimensions

(Re)configuring: (recall Periphery: capacity 5; Middle: capacity 10; Centre: capacity 15)

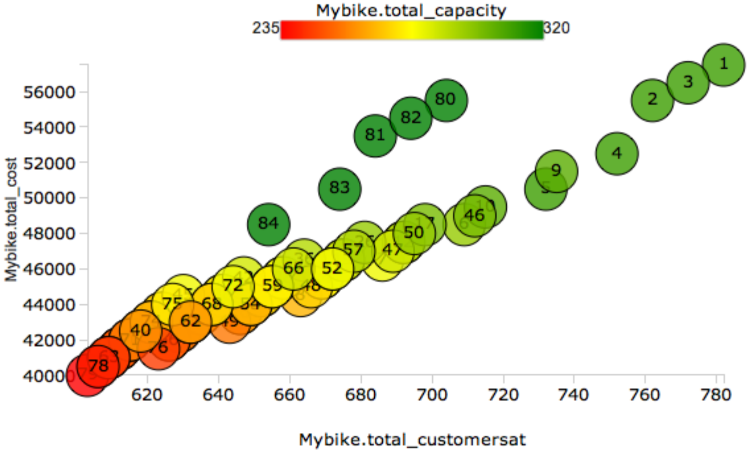
Configuration 1 330 stations of which 100 of type P, 150 of type M and 80 of type C ($\pm$ size of BSS in Barcelona in 2009)

Froehlich, Neumann, Oliver @ IJCAI'09

Configuration 2 390 stations of which 200 of type P, 150 of type M and 40 of type C

ClaferMOO Visualizer: Pareto front

Murashkin, Antkiewicz, Rayside, Czarnecki @ SPLC'13



Minimising cost whilst maximising customer satisfaction and capacity

Model checking: $M \models \phi$?

Decide whether a model M satisfies a (temporal) logic formula ϕ

- If $M \not\models \phi$, then it is usually easy to generate a counterexample

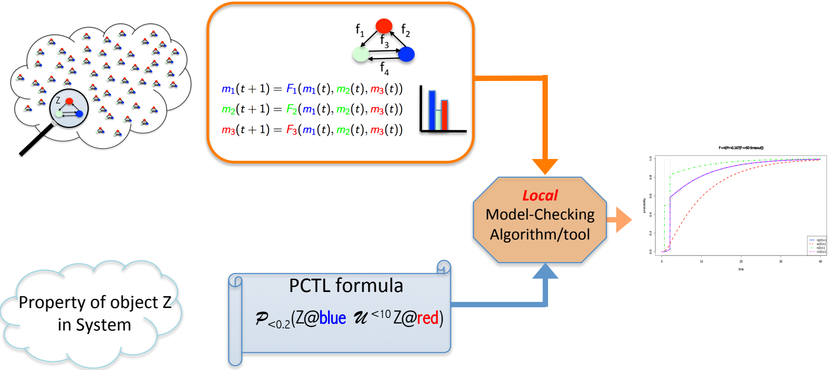
On-the-fly model checking

- Generate only as much of the state space as necessary to verify ϕ
- Improves performance and allows to handle infinite-state systems

Mean-field model checking discrete time Markov population models:

- Analyse the properties of the behaviour of a single agent in the context of the overall system behaviour (time-inhomogeneous)
- Approximation technique replacing actual stochastic/probabilistic system interactions, that often lead to a combinatorial explosion, by an approximation of the average system behaviour over time in terms of fractions (numbers) of elements present in a population
- These are defined as the solution of a set of difference equations

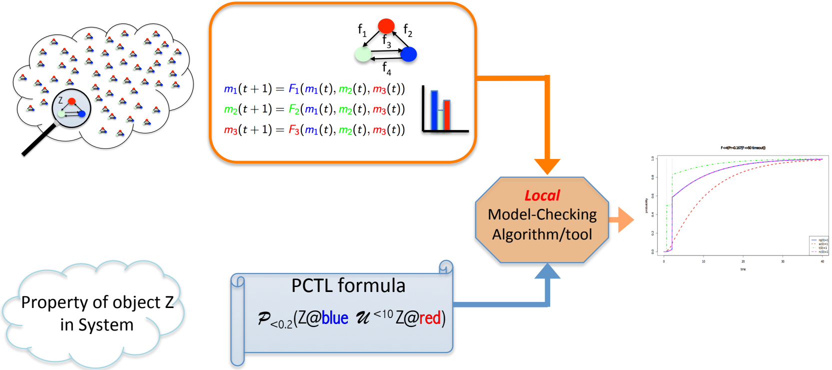
Latella, Loreti, Massink @ TGC'13, *Sci. Comput. Program.*, 2015



Local on-the-fly model-checking algorithm based on mean-field approximation in discrete time implemented in FlyFast

Fast on-the-fly model checking IDTMCs

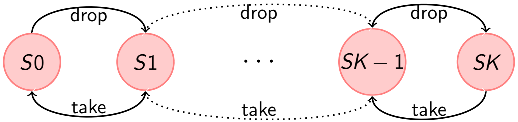
Latella, Loreti, Massink @ TGC'13, *Sci. Comput. Program.*, 2015



FLYFAST: highly suitable for CAS: only solve difference equations that depend on local object states (but *not* on the population size, i.e. no need to generate huge state spaces)

BSS as a Markov population model

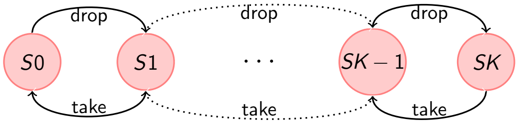
A docking station S with K parking slots:



Fricker, Gast @ EURO Journal on Transportation and Logistics, 2014

BSS as a Markov population model

A docking station S with K parking slots:



Fricker, Gast @ EURO Journal on Transportation and Logistics, 2014

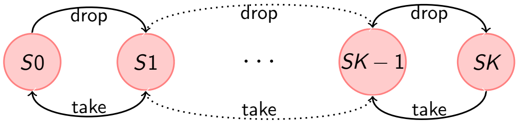
Oscillating functions for request/return rates model commuter flows

Hayden, *Scalable Performance Analysis of Massively Parallel Stochastic Systems*

PhD thesis, Imperial College London, 2011

BSS as a Markov population model

A docking station S with K parking slots:



Fricker, Gast @ EURO Journal on Transportation and Logistics, 2014

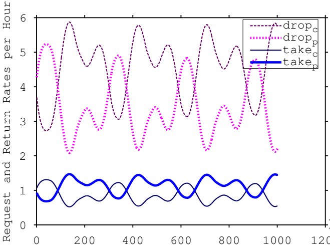
Oscillating functions for request/return rates model commuter flows

Hayden, *Scalable Performance Analysis of Massively Parallel Stochastic Systems*

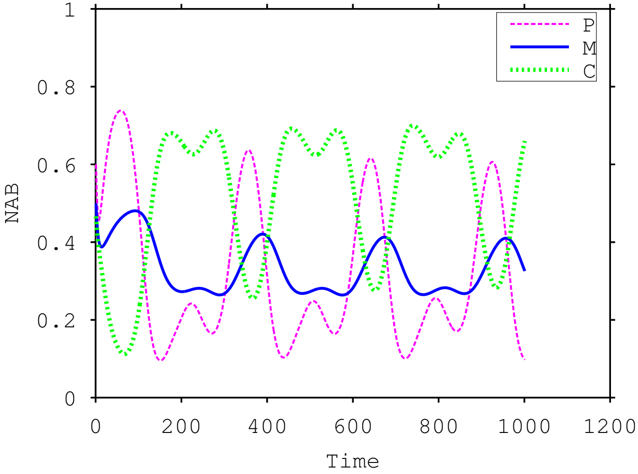
PhD thesis, Imperial College London, 2011

In line with data of BSS in Barcelona:

Froehlich, Neumann, Oliver @ IJCAI'09

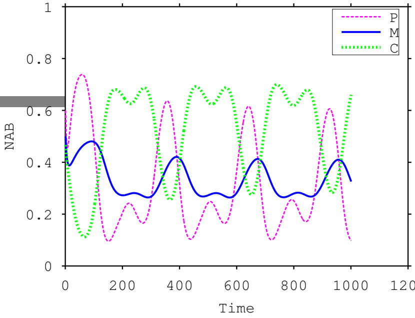


Mean field analysis (1/3)



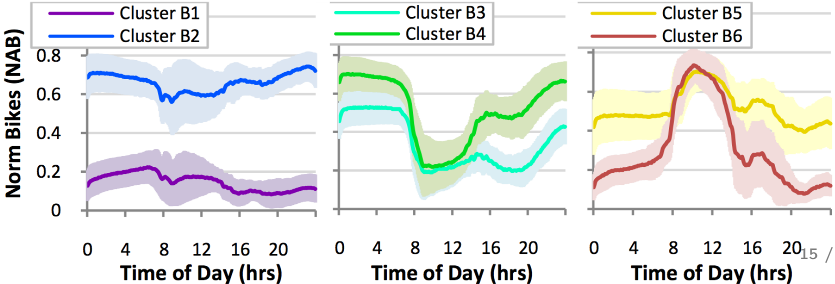
Configuration 1 (recall ± size of BSS in Barcelona in 2009)

Mean field analysis (1/3)

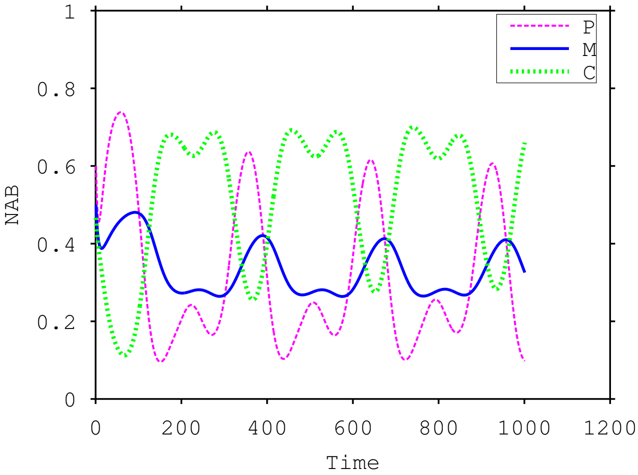


⇒ In line with average filling degree of docking stations over time in Barcelona in 2009 (NAB = normalised activity/bicycle data)

Froehlich, Neumann, Oliver @ IJCAI'09

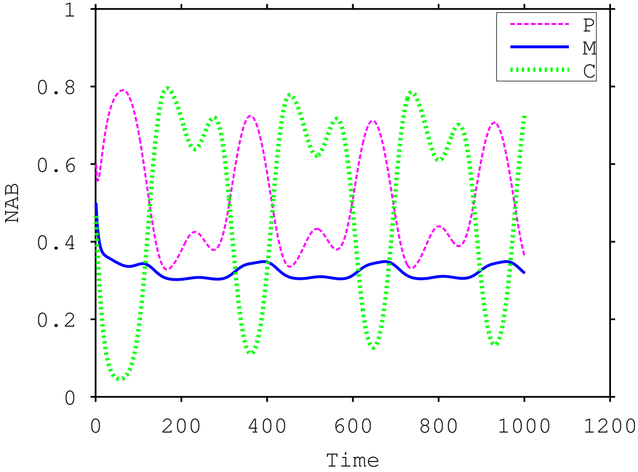


Mean field analysis (2/3)



Now compare with configuration 2

Mean field analysis (3/3)



While P-type stations are in general less empty in configuration 1, filling degree of C-type stations fluctuates more in configuration 2

Mean field model checking

What's the probability that within 30 minutes a P-type docking station

1. gets full, but then, with high probability ($\geq$0.99) has a free slot within 6 minutes,

or

2. does not get full, but then, with low probability ($\leq$0.01) gets full within 6 minutes

$$\mathcal{P}_{=?}(\ true\ \mathcal{U}^{\leq 30}\ ((SP5 \wedge \mathcal{P}_{\geq 0.99}(SP5\ \mathcal{U}^{\leq 6}\ \neg SP5)) \vee (\neg SP5 \wedge \mathcal{P}_{\leq 0.01}(\neg SP5\ \mathcal{U}^{\leq 6}\ SP5)))\ ) \qquad \text{(P2)}$$

Mean field model checking

What's the probability that within 30 minutes a P-type docking station

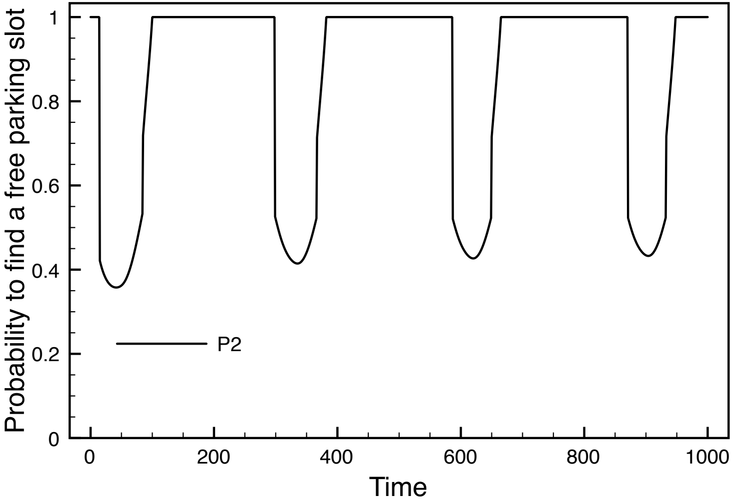
1. gets full, but then, with high probability ($\geq$0.99) has a free slot within 6 minutes,

or

2. does not get full, but then, with low probability ($\leq$0.01) gets full within 6 minutes

$$\mathcal{P}_{=?}(\ \mathit{true}\ \mathcal{U}^{\leq 30}\ ((SP5\ \wedge\ \mathcal{P}_{\geq 0.99}(SP5\ \mathcal{U}^{\leq 6}\ \neg SP5))\ \vee\ (\neg SP5\ \wedge\ \mathcal{P}_{\leq 0.01}(\neg SP5\ \mathcal{U}^{\leq 6}\ SP5)))\ ) \qquad \text{(P2)}$$

Result of model checking P2



Analysed in just a few seconds!

Conclusions and future work

quanticol
www.quanticol.eu

Orthogonal approaches evaluating BSS designs with automated tools

1. Variability analysis (by multi-objective optimisation with ClaferMOO) on system configurations
2. Performance analysis (by mean field model checking with FlyFast) on behavioural models

⇒ Integrate into a decision support system for BSS design, using performance analysis results as input for variability modelling (VERY*SCART 2016?)

e.g. the probability of finding an empty/full docking station, based on the capacity of a BSS configuration, may directly impact user satisfaction

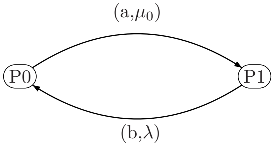
! Feasible for BSS of realistic size since model-checking time of FlyFast is independent from the population size

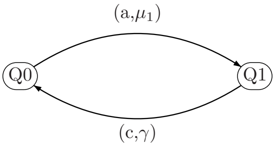
Thanks!

And see you on Friday for my talk at SEFM? ☺

Scalability: synchronise two automata

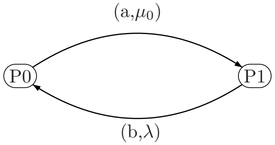
www.quanticol.eu

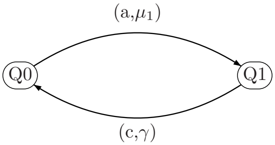




$$System = P_0 \bowtie_a Q_0$$

Scalability: synchronise 500 automata





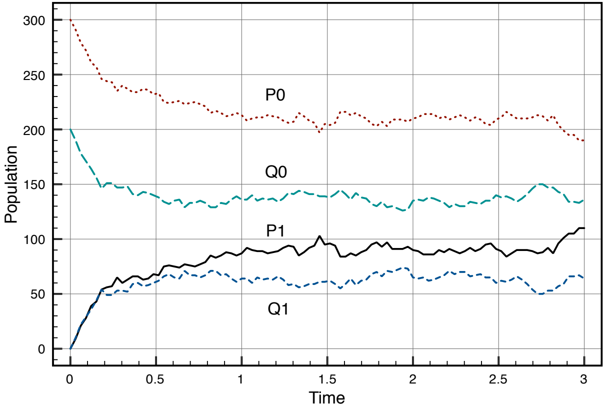
$$System = P_0[300] \bowtie_a Q_0[200]$$

State space explosion

quanticol
www.quanticol.eu

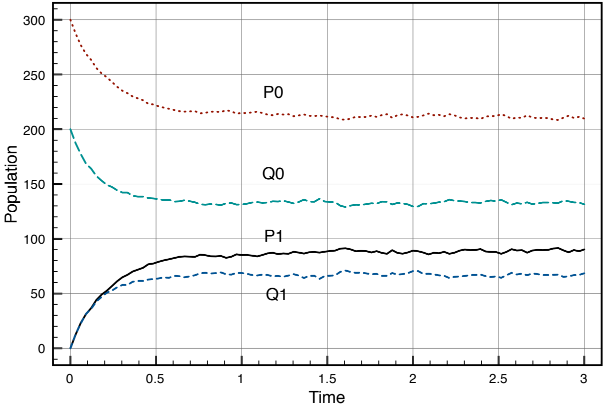
$\#P_0$	$\#Q_0$	$2^{\#P_0+\#Q_0}$	$\#P_0$	$\#Q_0$	$2^{\#P_0+\#Q_0}$
1	1	4	6	5	2,048
2	1	8	6	6	4,096
2	2	16	7	6	8,192
3	2	32	7	7	16,384
3	3	64	8	7	32,768
4	3	128	8	8	65,536
4	4	256	9	8	131,072
5	4	512	9	9	262,144
5	5	1,024	10	9	524,288
			10	10	1,048,576

Simulation, 1 run



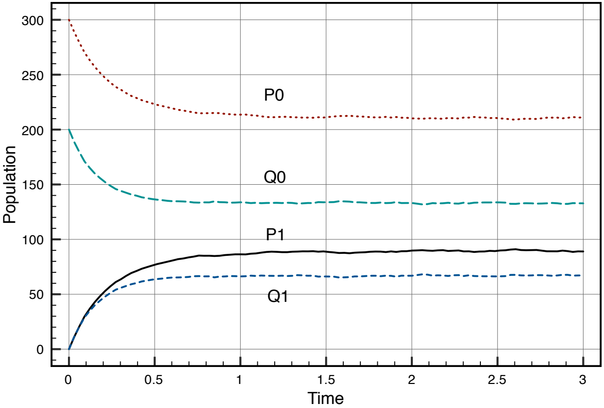
$$P0 = 300;\ Q0 = 200;\ \mu_0 = \mu_1 = 2;\ \lambda = 3;\ \gamma = 4$$

Simulation, 10 run



$$P0 = 300;\ Q0 = 200;\ \mu_0 = \mu_1 = 2;\ \lambda = 3;\ \gamma = 4$$

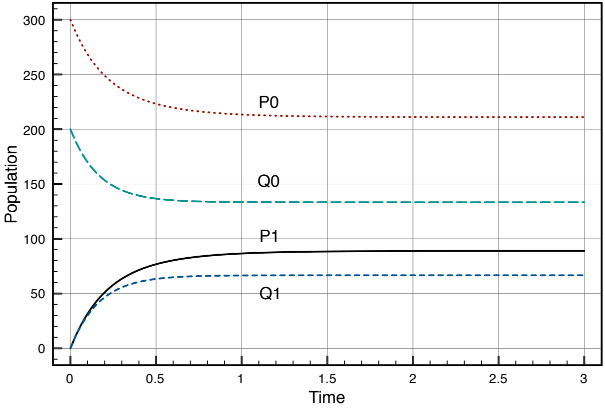
Simulation, 100 run



$$P0 = 300;\ Q0 = 200;\ \mu_0 = \mu_1 = 2;\ \lambda = 3;\ \gamma = 4$$

Fluid flow approximation

Hillston @ QEST'05



$P0 = 300;\ Q0 = 200;\ \mu_0 = \mu_1 = 2;\ \lambda = 3;\ \gamma = 4$