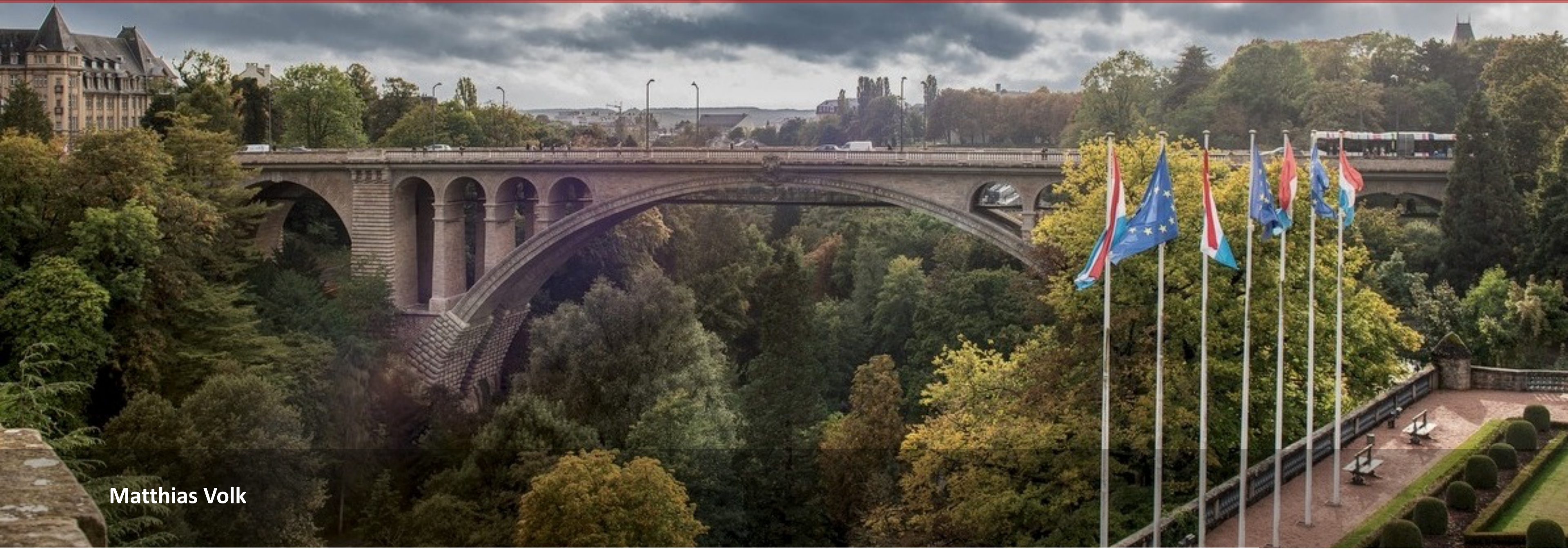
Verification of Parametric Markov Chains

SYNCOP 2024



Matthias Volk

Formal System Analysis Group

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Probabilistic models

Fault-tolerant systems

Robotics

Randomized algorithms

Communication protocols

Artificial intelligence

Probabilistic models

Fault-tolerant systems

<u>Problem</u>:
probabilities/rates must be precisely known

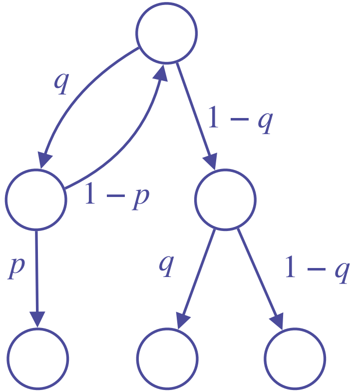
Randomized algorithms

Communication protocols

Artificial intelligence

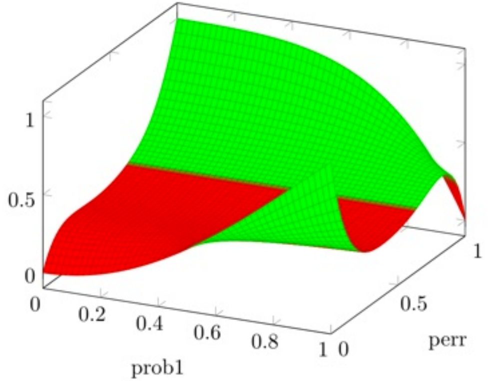
Parametric Markov Chains

use **parameters** to capture uncertainties on probabilities



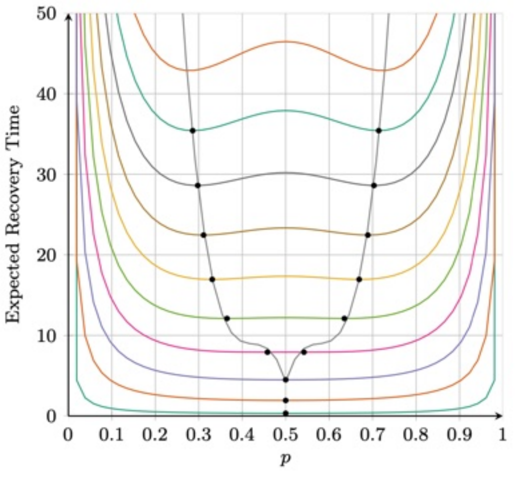
Goals:

- find **one** parameter valuation satisfying the requirement
- find **all** parameter valuation satisfying the requirement
- find the parameter valuation **optimizing** the requirement
- **partition** the parameter space into valuations satisfying/ violating the requirement

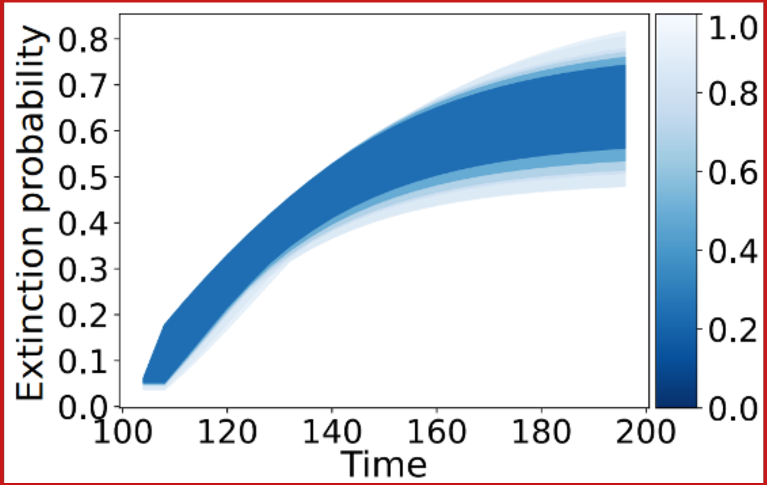


Parametric Markov chains

Synthesize optimal probabilities for randomized algorithms

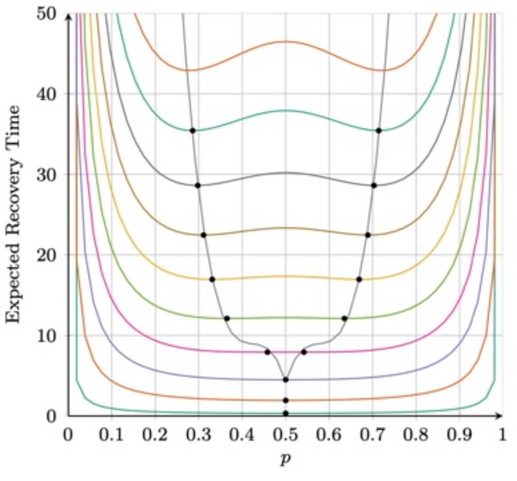


Handle uncertainty in continuous-time models



Parametric Markov chains

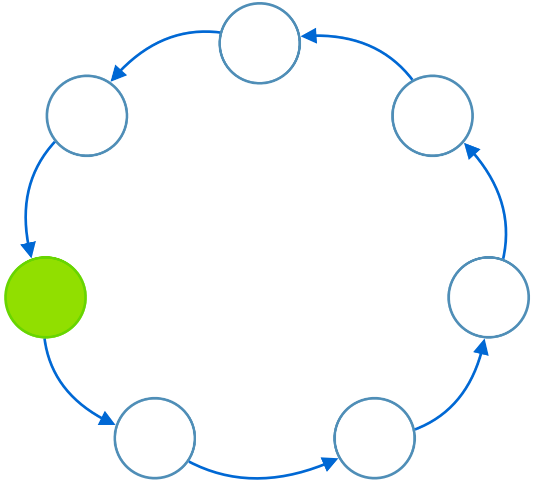
Synthesize optimal probabilities
for randomized algorithms



Joint work with Borzoo Bonakdarpour,
Joost-Pieter Katoen and Saba Aflaki

Self-stabilizing Algorithms

- Distributed system with several processes
- anonymous and synchronous
- exactly one process has token (stable configuration)
- after faults occur, reach a stable configuration again



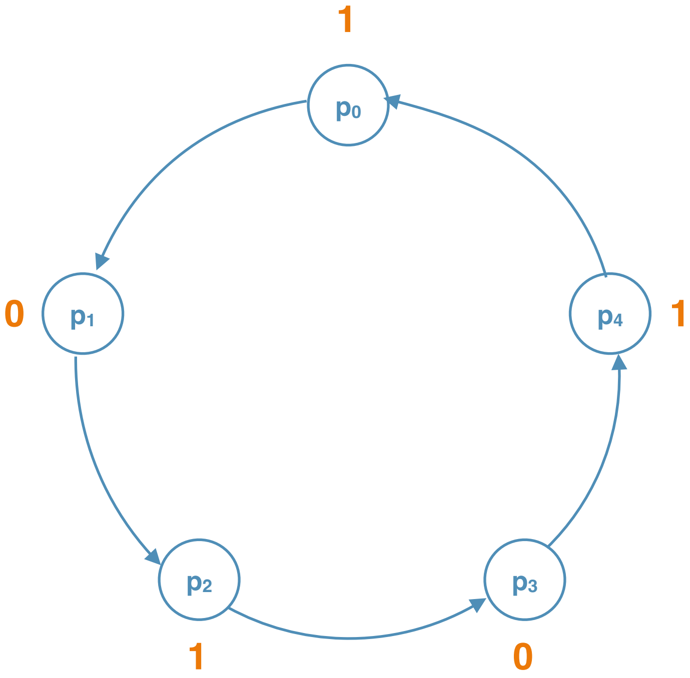
Problem:
In **anonymous** networks where all processes are **identical**
no deterministic algorithm exists.

[Angluin, 1980]

use randomization

Herman's Protocol

[Herman, 1990]



- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \; + \; (1-p) : x_i = 1;$$

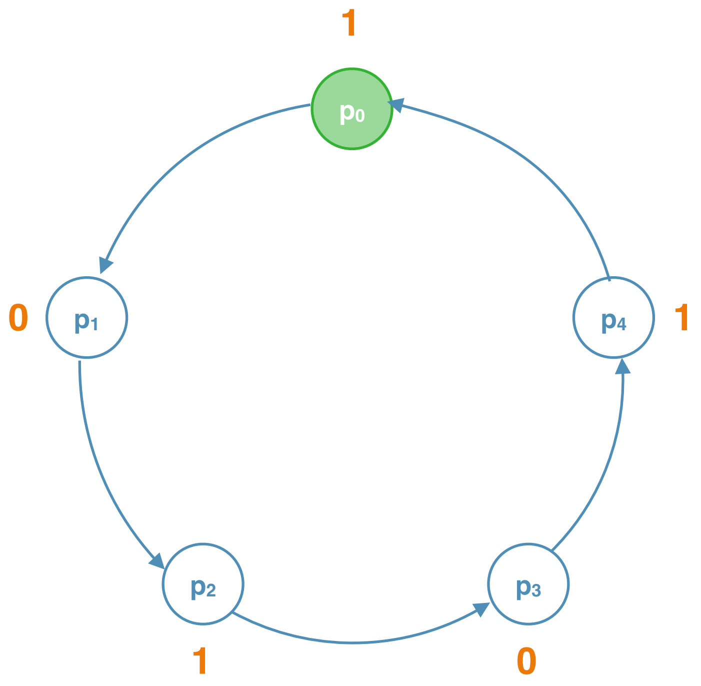
$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$

[Herman, 1990]

Herman's Protocol

- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \; + \; (1-p) : x_i = 1;$$
$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$

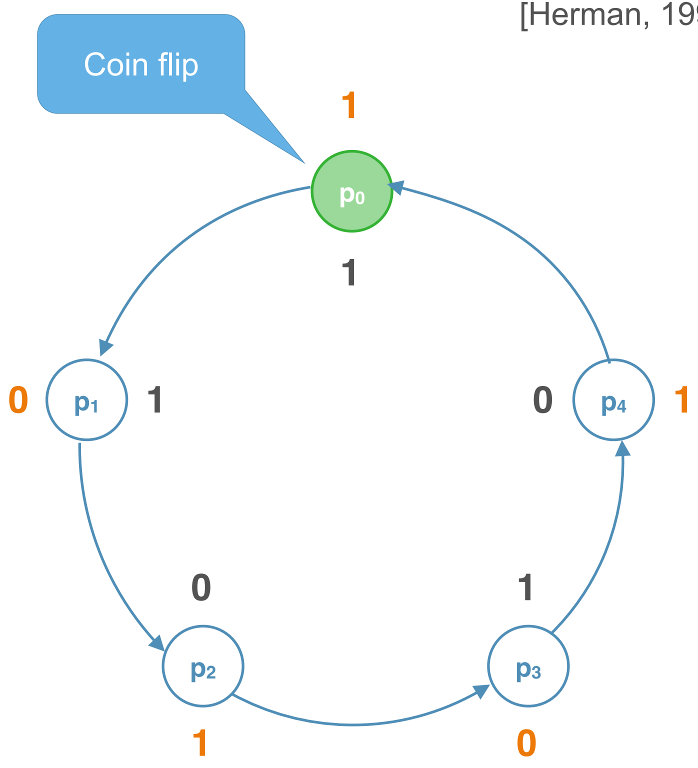


Herman's Protocol

[Herman, 1990]

- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \; + \; (1-p) : x_i = 1;$$
$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$

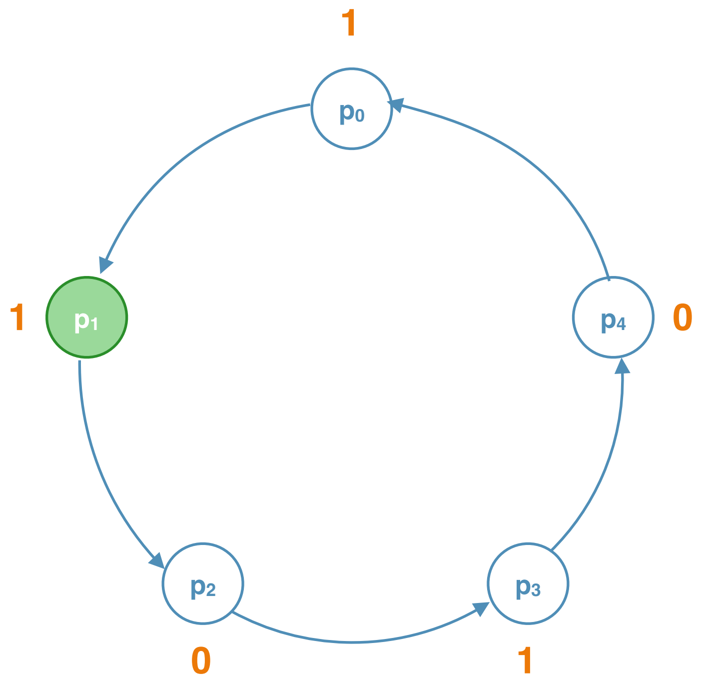


[Herman, 1990]

Herman's Protocol

- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \; + \; (1-p) : x_i = 1;$$
$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$

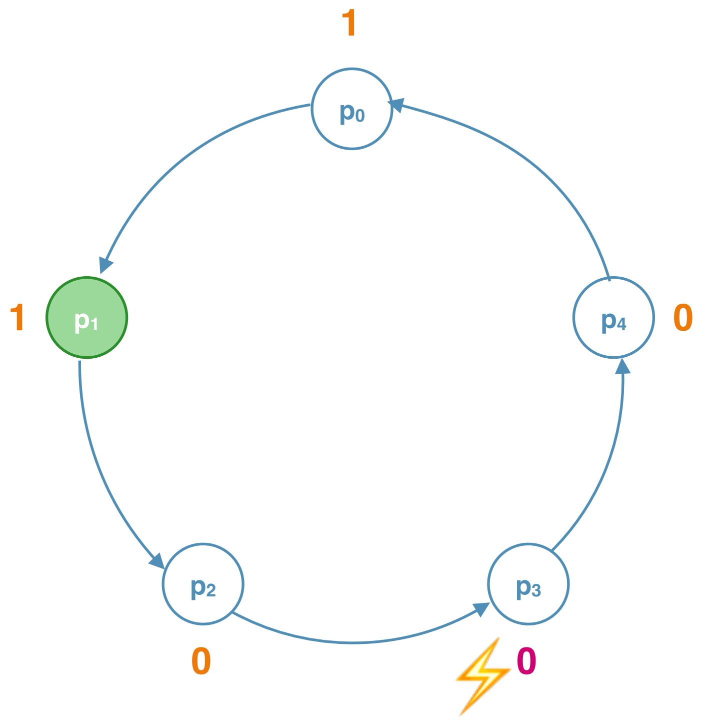


[Herman, 1990]

Herman’s Protocol

- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \ + \ (1-p) : x_i = 1;$$
$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$

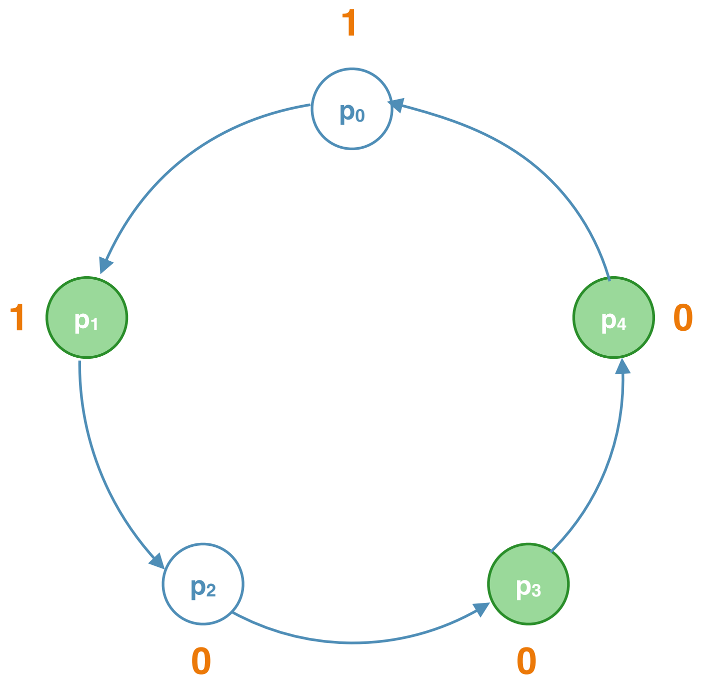


[Herman, 1990]

Herman's Protocol

- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

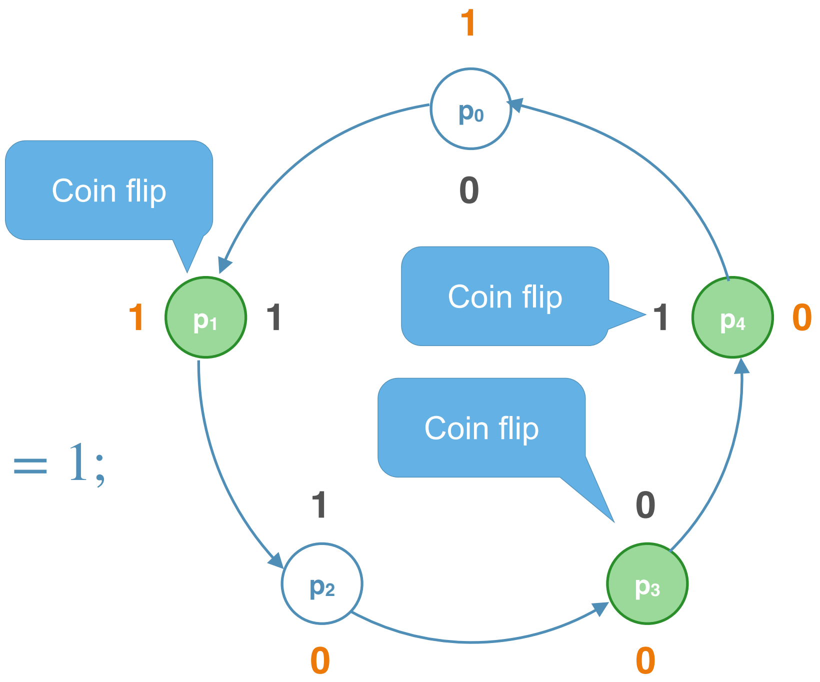
$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \; + \; (1-p) : x_i = 1;$$
$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$



Herman's Protocol

- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \; + \; (1-p) : x_i = 1;$$
$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$



TU/e

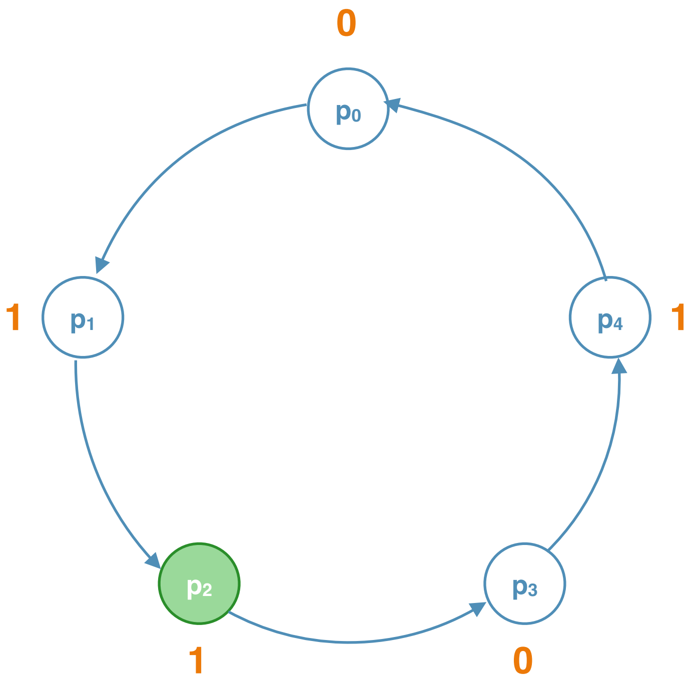
[Herman, 1990]

Herman's Protocol

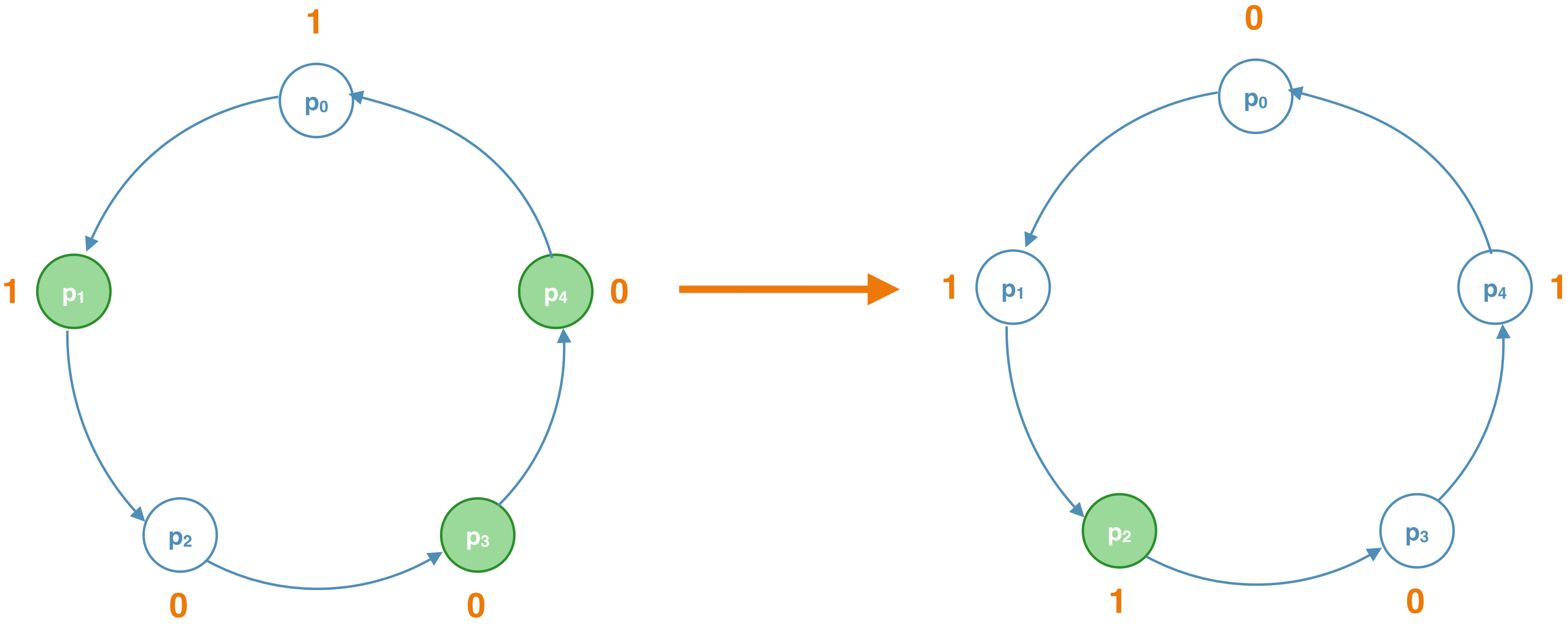
- Processes throw coin with bias $p = 0.5$
- Process has token iff $x_i = x_{i-1}$
- Protocol:

$$x_i = x_{i-1} \longrightarrow p : x_i := 0 \ + \ (1-p) : x_i = 1;$$

$$x_i \neq x_{i-1} \longrightarrow 1 : x_i := x_{i-1};$$

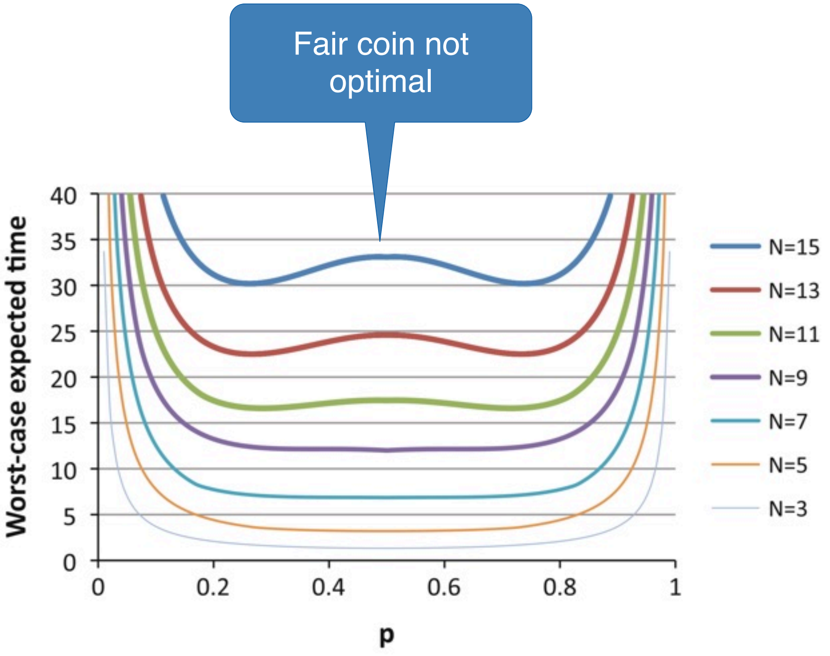


Recovery



Research questions

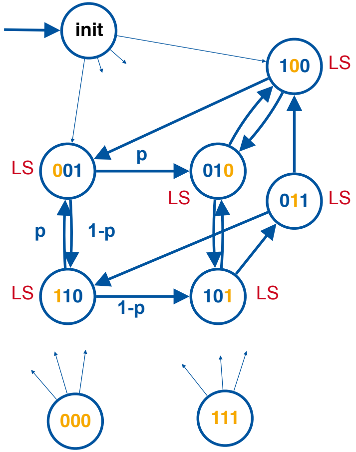
- How fast does the algorithm recover?
- worst-case recovery time for fair coin:
    - $4/27 \cdot N^2$ with 3 tokens [McIver, Morgan, Inform. Proc. Lett. 2005]
    - proof by [Bruna et al., ICALP 2016]
- average recovery time
- How does the bias of the coin influence the recovery time?



[Kwiatkowska, Norman, Parker, Form. Asp. Comp. 2012]

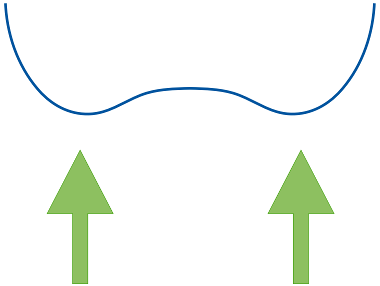
Approach

randomized distributed algorithm (pDTMC) → **parameter synthesis** → optimal coin bias



different approaches:

1. **exact solution function**
2. **approximation**



Solution function via state elimination

Recovery time for 9 processes (513 states)

```
(-1 * (66679200000*p^150+6029419284*p^2+(-48023212859)*p^3+303169269311*p^4+(-1591706649555)*p^5+7132430478302*p^6+(-27621831189630)*p^7+92568942443223*p^8+(-264909526201823)*p^9+619803167921730*p^10+
(-1021619558785267)*p^11+2077630473658 18*p^12+6905522850360698*p^13+(-35269259178104861)*p^14+119132331068733574*p^15+(-315751948911924840)*p^16+666223844077176651*p^17+
(-1014661442776628817)*p^18+4923527123011 14739*p^19+3584715082080701590*p^20+(-17436271051278706743)*p^21+52143258290153557618*p^22+(-120824372129707434362)*p^23+222669778239260495673*p^24+
(-297706943349633909580)*p^25+131944680466840615121*p^26+795681352288241394885*p^27+(-3480177908687899647893)*p^28+9396537644362384212113*p^29+(-19930023667324720583790)*p^30+34475174362403995631160*p^31+
(-46135194272503394559034)*p^32+34031163244175383657844*p^33+47569457268509400718502*p^34+(-279603784340770466621477)*p^35+777137708239224789746258*p^36+(-1654131093386834838583954)*p^37+2916226220664687228802955*p^38+
(-4232026032488155073737171)*p^39+4540711797528773171317340*p^40+(-1508014981101474183611311)*p^41+(-9018862734159392226561034)*p^42+33149079078370175984804968*p^43+(-77892217385089921080465225)*p^44+147707022929047442377660195*p^45+
(-237237271004639215921869694)*p^46+319247479124112285518407445*p^47+(-328506390597214254203120457)*p^48+146138625543858853590129435*p^49+404795416282777122374902252*p^50+
(-1533595624834062606235975128)*p^51+3399759522664754268492230736*p^52+(-5947020812448464629612443266)*p^53+8641143606020243349150000649*p^54+
(-10141788730000063411813409125)*p^55+8018889933728499474039455165*p^56+1262221310795264388681 1762725*p^57+(-21748589273333637213067671209)*p^58+56429118693701155309804758336*p^59+
(-104498108547486979069837101240)*p^60+157833186367130301141642877394*p^61+(-198135491248488644954584090539)*p^62+197667387261876891872763551098*p^63+(-127945366724613905869838580104)*p^64+
(-189810866130671417363332599 18)*p^65+194658561557243997988388391108*p^66+(-249419992153322751566661238790)*p^67+(-101370218484520022756708475810)*p^68+1259576040524385366815119630741*p^69+
(-3613462229688457248756620644867)*p^70+7223479879092115834347349224441*p^71+(-11280717703151243263990420283564)*p^72+13385411559811172476779562006623*p^73+(-8877263344189491274681179339228)*p^74+
(-9265624261122030467982678340492)*p^75+49095720731894898995772693558450*p^76+(-115774215553121343634764583141671)*p^77+203856696154201331198253860214052*p^78+
(-285064865159864375070200639336715)*p^79+292098498846861986255529628 62676*p^80+(-101261748492092514141584226372563)*p^81+(-480302482467014180469850922386987)*p^82+1713763305401551699173868557725008*p^83+
(-3902883017763874800988492057480874)*p^84+7332513522182466645728505678392560*p^85+(-12165046794673884994180789930848579)*p^86+18298692329882839591202930821632113*p^87+
(-25206789166074944733201139632169466)*p^88+31795326831558368577001011498734785*p^89+(-36331221380870522802508044563500103)*p^90+36499395047215679923281378704666004*p^91+
(-29634964750562340741997135802134564)*p^92+13143543770757495518331149822141860*p^93+14929916540375781901734973949012813*p^94+(-55284003121966118170404177408723588)*p^95+106845224750035730765181593200861201*p^96+
(-166519747209009634473077773838701970)*p^97+229344147481584236414743922725969350*p^98+(-289080524716193630947226140553041469)*p^99+339181137783924043254413042398803064*p^100+
(-373929767178486905110666526631269792)*p^101+389493542531258084420904565982687197*p^102+(-384621830185579326177676674331633702)*p^103+360814265329688608188085934432705250*p^104+
(-321923566476303834373416964658169561)*p^105+273312795360938328906296405148760565*p^106+(-220799169371465462148217983690380021)*p^107+169651835050635095214331961324918405*p^108+
(-123863683586328614730497398992628223)*p^109+85812032895370091106288381182571122*p^110+(-56302063374139874898878065653740792)*p^111+34891519966735510772670885075868121*p^112+
(-20350135126132905137016213055095920)*p^113+11114659707241836993234463740710020*p^114+(-5644142549202475444561961612115389)*p^115+2636213257602626181335853239912277*p^116+
(-1112716296393482573554252224155807)*p^117+410867280033761312416140162884528*p^118+(-123317106727131027234501821292014)*p^119+23242131303048076971231766550641*p^120+2897270317252492364427408513316*p^121+
(-5286712927118235615543898829569)*p^122+2758135958693617704790492155886*p^123+(-708697319566281430474524331303)*p^124+(-182076646433466979844816657452)*p^125+368526856242442146091303316684*p^126+
(-290525328163275190300841627632)*p^127+172160834149573967022988536654*p^128+(-86074170144485315481082841899)*p^129+37885785183378230264928652347*p^130+(-14977578495225272230930809145)*p^131+5374066615418642232204220917*p^132+
(-1759731947580416312917560295)*p^133+527206031659862802634485418*p^134+(-144595797858519896313454020)*p^135+36272634607377449989802112*p^136+(-8304560110679941200630486)*p^137+1729541194356667714668489*p^138+
(-326189928883327705278600)*p^139+55388092206101232051285*p^140+(-8405824580483131751130)*p^141+1129644075959437501530*p^142+(-132859269498151305000)*p^143+13468870582378093125*p^144+(-1153467509991930000)*p^145+81161068525110000*p^146+
(-4505773116600000)*p^147+185078835900000*p^148+(-5000940000000)*p^149+(-544388567)*p+27367011))/
(768 * ((p) * (p+(-1)) * (p^2+(-1)*p+1) * (3*p^6+19*p^2+(-23)*p^3+19*p^4+(-9)*p^5+(-9)*p+3) * (105*p^20+(-40)*p^2+91*p^3+(-99)*p^4+(-140)*p^5+919*p^6+(-2449)*p^7+4775*p^8+(-8313)*p^9+14741*p^10+(-26199)*p^11+41813*p^12+(-55321)*p^13+58533*p^14+
(-48608)*p^15+31066*p^16+(-14805)*p^17+4970*p^18+(-1050)*p^19+11*p+(-6)) * (105*p^36+531*p^2+(-138)*p+(-1890)*p^35+16209*p^34+(-88128)*p^33+340911*p^32+(-997992)*p^31+2294992*p^30+(-4244892)*p^29+6405978*p^28+(-7935824)*p^27+8040708*p^26+
(-6530706)*p^25+4006236*p^24+(-1491192)*p^23+(-192160)*p^22+823736*p^21+(-722264)*p^20+374038*p^19+(-99515)*p^18+(-28929)*p^17+71429*p^16+(-88756)*p^15+97710*p^14+(-91384)*p^13+66413*p^12+(-32086)*p^11+3796*p^10+8899*p^9+
(-7957)*p^8+1986*p^7+2316*p^6+(-3362)*p^5+2542*p^4+(-1360)*p^3+19) * (7875*p^86+(-3994)*p^2+(-50890)*p^4+723*p+(-338625)*p^85+7041975*p^84+(-94281075)*p^83+912296430*p^82+(-6785831280)*p^81+40254843740*p^80+(-194774748950)*p^79+777996208074*p^78+
(-2570432003126)*p^77+6939579335863*p^76+(-14680414368238)*p^75+21067532992446*p^74+(-4549324218607)*p^73+(-87116069123937)*p^72+329977280006443*p^71+(-740801107503090)*p^70+1039499180835883*p^69+(-213300202885163)*p^68+
(-4023553809967423)*p^67+15560802251943411*p^66+(-39536082410664394)*p^65+80919232846166193*p^64+(-142157578819263248)*p^63+220718226204346765*p^62+(-307843693676318247)*p^61+389614615458508805*p^60+
(-450416428300641227)*p^59+477635027880677701*p^58+(-465645981677885387)*p^57+417423160453701120*p^56+(-343225463463671045)*p^55+257134665530188855*p^54+(-172990307877657035)*p^53+101205853851844786*p^52+
(-47276564964329749)*p^51+11977172556978726*p^50+7314673356136844*p^49+(-14907030698472545)*p^48+15294888603333145*p^47+(-12192473693347075)*p^46+8139429301167013*p^45+(-4542717910224291)*p^44+1951742066380323*p^43+
(-389859483935965)*p^42+(-366336999050958)*p^41+600424218157499*p^40+(-557349986025491)*p^39+407951675709205*p^38+(-250340535612776)*p^37+127636958947078*p^36+(-48993723345366)*p^35+7097983540773*p^34+10218938969419*p^33+
(-13919256450287)*p^32+11645218379663*p^31+(-7782538161356)*p^30+4355434919560*p^29+(-1988327204514)*p^28+629387241600*p^27+6984125674*p^26+(-216766317215)*p^25+221331742177*p^24+(-155558377353)*p^23+86257799832*p^22+
(-37224316263)*p^21+10236248500*p^20+1074922110*p^19+(-3856419115)*p^18+3201296469*p^17+(-1831837959)*p^16+786522594*p^15+(-228070785)*p^14+10084624*p^13+40190417*p^12+(-32764517)*p^11+16739491*p^10+(-6226727)*p^9+1549929*p^8+(-75708)*p^7+
(-174547)*p^6+119545*p^5+16245*p^3+(-91))))
```

Solution function via state elimination

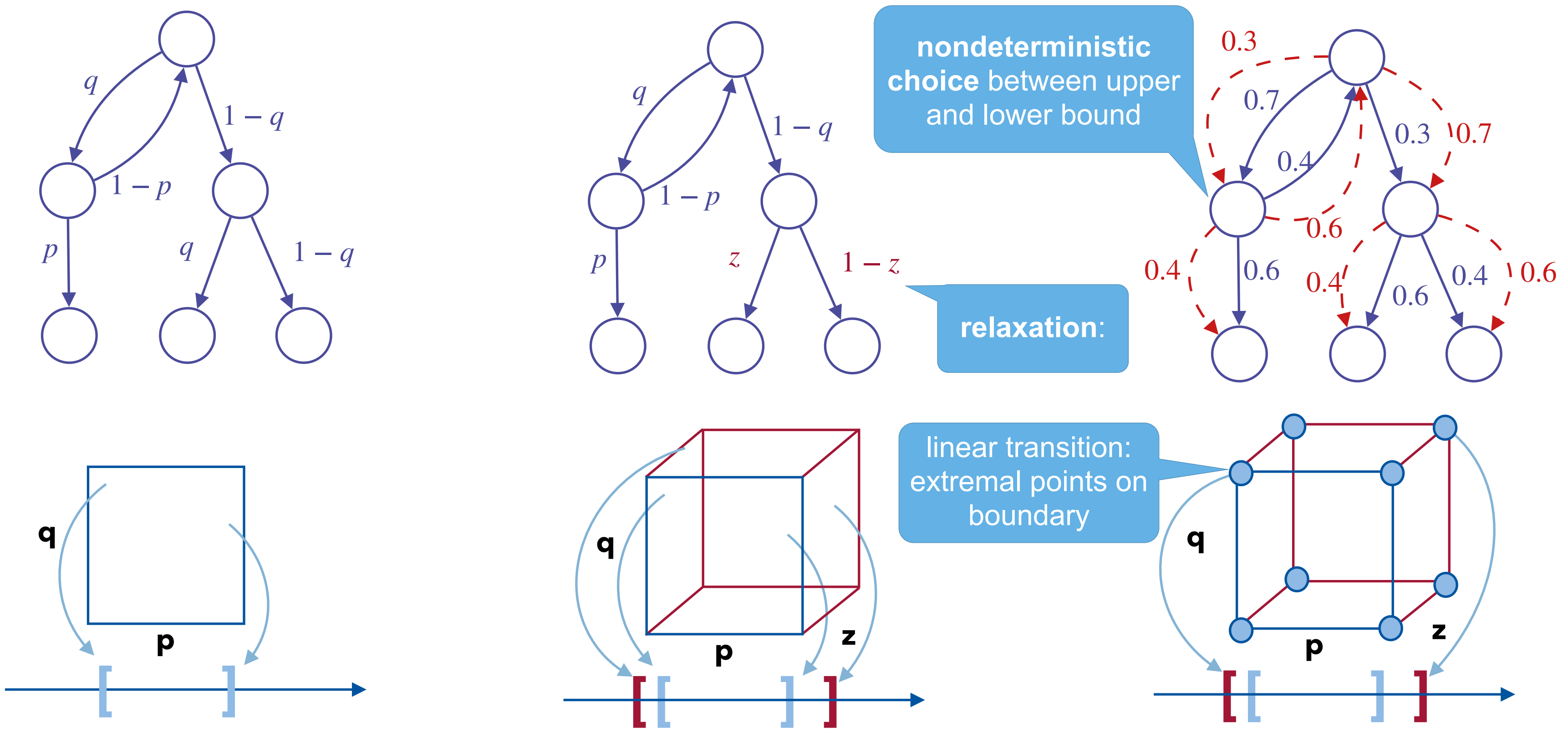
Recovery time for 9 processes (513 states)

```
(-1 * (66679200000*p^150+6029419284*p^2+(-48023212859)*p^3+303169269311*p^4+(-1591706649555)*p^5+7132430478302*p^6+(-27621831189630)*p^7+92568942443223*p^8+(-264909526201823)*p^9+619803167921730*p^10+
(-1021619558785267)*p^11+207763047365818*p^12+6905522850360698*p^13+(-35269259178104861)*p^14+119132331068733574*p^15+(-315751948911924840)*p^16+666223844077176651*p^17+
(-1014661442776628817)*p^18+4923527123011114739*p^19+3584715082080701590*p^20+(-17436271051278706743)*p^21+52143258290153557618*p^22+(-120824372129707434362)*p^23+222669778239260495673*p^24+
(-297706943349633909580)*p^25+1319446804668406151 21*p^26+795681352288241394885*p^27+(-3480177908687899647893)*p^28+9396537644362384212113*p^29+(-19930023667324720583790)*p^30+34475174362403995631160*p^31+
(-46135194272503394559034)*p^32+34031163244175383657844*p^33+47569457268509400718502*p^34+(-279603784340770466621477)*p^35+777137708239224789746258*p^36+(-1654131093386834838583954)*p^37+2916226220664687228802955*p^38+
(-4232026032488155073737171)*p^39+4540711797528773171317340*p^40+(-15080149811014741836113 11)*p^41+(-9018862734159392226561034)*p^42+33149079078370175984804968*p^43+(-77892217385089921080465225)*p^44+147707022929047442377660195*p^45+
(-237237271004639215921869694)*p^46+319247479124112285518407445*p^47+(-328506390597214254203120457)*p^48+146138625543858853590129435*p^49+404795416282777122374902252*p^50+
(-1533595624834062606235975128)*p^51+3399759522664754268492230736*p^52+(-5947020812448464629612443266)*p^53+8641143606020243349150000649*p^54+
(-10141788730000063411813409125)*p^55+8018889933728499474039455165*p^56+1262213107952643886811762725*p^57+(-21748589273333637213067671209)*p^58+56429118693701155309804758336*p^59+
(-104498108547486979069837101240)*p^60+157833186367130301141642877394*p^61+(-198135491248488644954584090539)*p^62+197667387261876891872763551098*p^63+(-127945366724613905869838580104)*p^64+
(-18981086613067141736333259918)*p^65+194658561557243997988388391108*p^66+(-249419992153322751566661238790)*p^67+(-101370218484520022756708475810)*p^68+1259576040524385366815119630741*p^69+
(-3613462229688457248756620644867)*p^70+7223479879092115834347349224441*p … *p^73+(-8877263344189491274681179339228)*p^74+
(-9265624261122030467982678340492)*p^75+490957207318948989957726935584 … 2*p^78+
(-285064865159864375070200639336715)*p^79+2920984988468619862525529628 … 86987)*p^82+1713763305401551699173868557725008*p^83+
(-3902883017763874800988492057480874)*p^84+733251352218246666457285056 7 … 0821632113*p^87+
(-25206789166074944733201139632169466)*p^88+3179532683155836857700101 14 … 378704666004*p^91+
(-29634964750562340741997135802134564)*p^92+13143543770757495518331149 8 … 77408723588)*p^95+106845224750035730765181593200861201*p^96+
(-166519747209009634473077773838701970)*p^97+22934414748158423641474392 … 4413042398803064*p^100+
(-373929767178486905110666526631269792)*p^101+38949354253125808442090456 … 08188085934432705250*p^104+
(-321923566476303834373416964658169561)*p^105+273312795360938328906296405148760565*p^106+(-220799169371465462148217983690380021)*p^107+169651835050635095214331961324918405*p^108+
(-123863683586328614730497398992628223)*p^109+85812032895370091106288381182571122*p^110+(-56302063374139874898878065653740792)*p^111+34891519966735510772670885075868121*p^112+
(-20350135126132905137016213055095920)*p^113+11114659707241836993234463740710020*p^114+(-5644142549202475444561961612115389)*p^115+2636213257602626181335853239912277*p^116+
(-1112716296393482573554252224155807)*p^117+410867280033761312416140162884528*p^118+(-123317106727131027234501821292014)*p^119+23242131303048076971231766550641*p^120+2897270317252492364427408513316*p^121+
(-5286712927118235615543898829569)*p^122+2758135958693617704790492155886*p^123+(-708697319566281430474524331303)*p^124+(-182076646433466979844816657452)*p^125+368526856242442146091303316684*p^126+
(-290525328163275190300841627632)*p^127+172160834149573967022988536654*p^128+(-86074170144485315481082841899)*p^129+37885785183378230264928652347*p^130+(-14977578495225272230930809145)*p^131+5374066615418642232204220917*p^132+
(-1759731947580416312917560295)*p^133+527206031659862802634485418*p^134+(-144595797858519896313454020)*p^135+36272634607377449989802112*p^136+(-8304560110679941200630486)*p^137+1729541194356667714668489*p^138+
(-326189928883327705278600)*p^139+55388092206101232051285*p^140+(-8405824580483131751130)*p^141+1129644075959437501530*p^142+(-132859269498151305000)*p^143+13468870582378093125*p^144+(-1153467509991930000)*p^145+81161068525110000*p^146+
(-4505773116600000)*p^147+185078835900000*p^148+(-5000940000000)*p^149+(-544388567)*p+27367011))/
(768 * ((p) * (p+(-1)) * (p^2+(-1)*p+1) * (3*p^6+19*p^2+(-23)*p^3+19*p^4+(-9)*p^5+(-9)*p+3) * (105*p^20+(-40)*p^2+91*p^3+(-99)*p^4+(-140)*p^5+919*p^6+(-2449)*p^7+4775*p^8+(-8313)*p^9+14741*p^10+(-26199)*p^11+41813*p^12+(-55321)*p^13+58533*p^14+
(-48608)*p^15+31066*p^16+(-14805)*p^17+4970*p^18+(-1050)*p^19+11*p+(-6)) * (105*p^36+531*p^2+(-138)*p+(-1890)*p^35+16209*p^34+(-88128)*p^33+340911*p^32+(-997992)*p^31+2294992*p^30+(-4244892)*p^29+6405978*p^28+(-7935824)*p^27+8040708*p^26+
(-6530706)*p^25+4006236*p^24+(-1491192)*p^23+(-192160)*p^22+823736*p^21+(-722264)*p^20+374038*p^19+(-99515)*p^18+(-28929)*p^17+71429*p^16+(-88756)*p^15+97710*p^14+(-91384)*p^13+66413*p^12+(-32086)*p^11+3796*p^10+8899*p^9+
(-7957)*p^8+1986*p^7+2316*p^6+(-3362)*p^5+2542*p^4+(-1360)*p^3+19) * (7875*p^86+(-3994)*p^2+(-50890)*p^4+723*p+(-338625)*p^85+7041975*p^84+(-94281075)*p^83+912296430*p^82+(-6785831280)*p^81+40254843740*p^80+(-194774748950)*p^79+777996208074*p^78+
(-2570432003126)*p^77+6939579335863*p^76+(-14680414368238)*p^75+21067532992446*p^74+(-4549324218607)*p^73+(-87116069123937)*p^72+329977280006443*p^71+(-740801107503090)*p^70+1039499180835883*p^69+(-213300202885163)*p^68+
(-4023553809967423)*p^67+15560802251943411*p^66+(-39536082410664394)*p^65+80919232846166193*p^64+(-142157578819263248)*p^63+220718226204346765*p^62+(-307843693676318247)*p^61+389614615458508805*p^60+
(-450416428300641227)*p^59+477635027880677701*p^58+(-465645981677885387)*p^57+417423160453701120*p^56+(-343225463463671045)*p^55+257134665530188855*p^54+(-172990307877657035)*p^53+101205853851844786*p^52+
(-47276564964329749)*p^51+11977172556978726*p^50+7314673356136844*p^49+(-14907030698472545)*p^48+15294888603333145*p^47+(-12192473693347075)*p^46+8139429301167013*p^45+(-4542717910224291)*p^44+1951742066380323*p^43+
(-389859483935965)*p^42+(-366336999050958)*p^41+600424218157499*p^40+(-557349986025491)*p^39+407951675709205*p^38+(-250340535612776)*p^37+127636958947078*p^36+(-48993723345366)*p^35+7097983540773*p^34+10218938969419*p^33+
(-13919256450287)*p^32+11645218379663*p^31+(-7782538161356)*p^30+4355434919560*p^29+(-1988327204514)*p^28+629387241600*p^27+6984125674*p^26+(-216766317215)*p^25+221331742177*p^24+(-155558377353)*p^23+86257799832*p^22+
(-37224316263)*p^21+10236248500*p^20+1074922110*p^19+(-3856419115)*p^18+3201296469*p^17+(-1831837959)*p^16+786522594*p^15+(-228070785)*p^14+10084624*p^13+40190417*p^12+(-32764517)*p^11+16739491*p^10+(-6226727)*p^9+1549929*p^8+(-75708)*p^7+
(-174547)*p^6+119545*p^5+16245*p^3+(-91))))
```

Solution:
Approximate optimal value

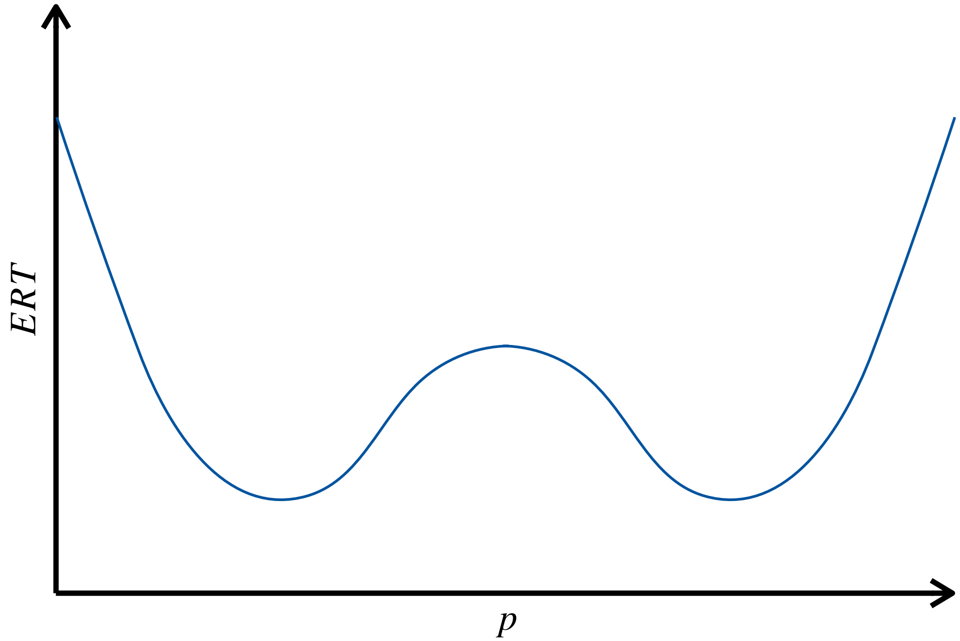
Parameter Lifting Algorithm

[Quatmann et al., 2016]



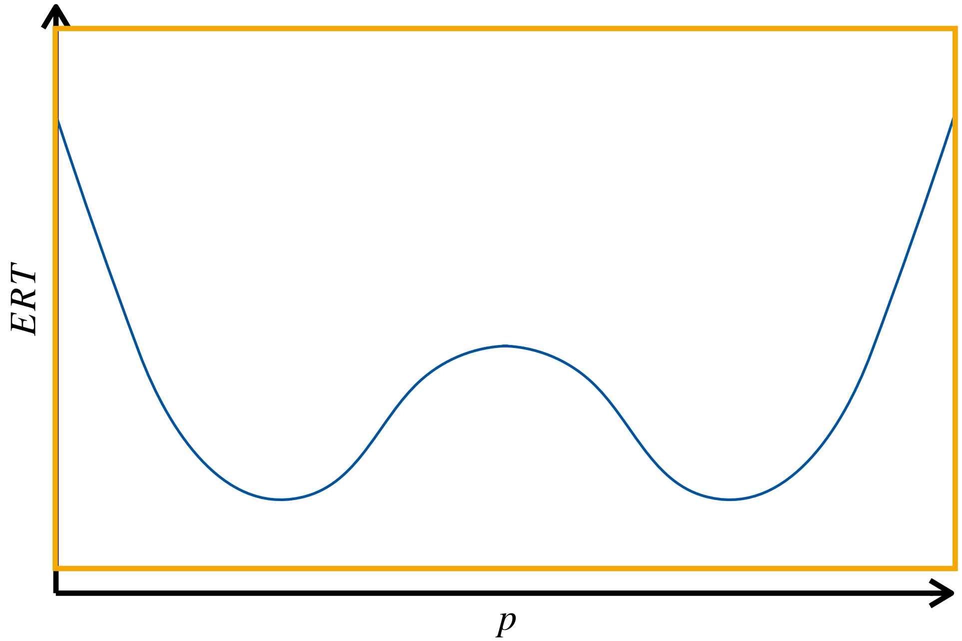
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



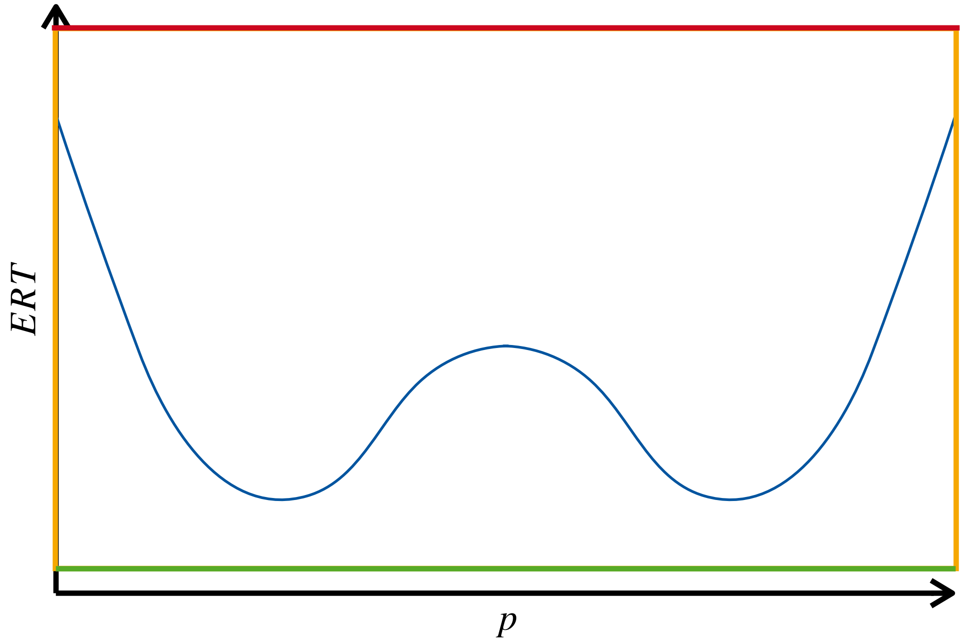
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



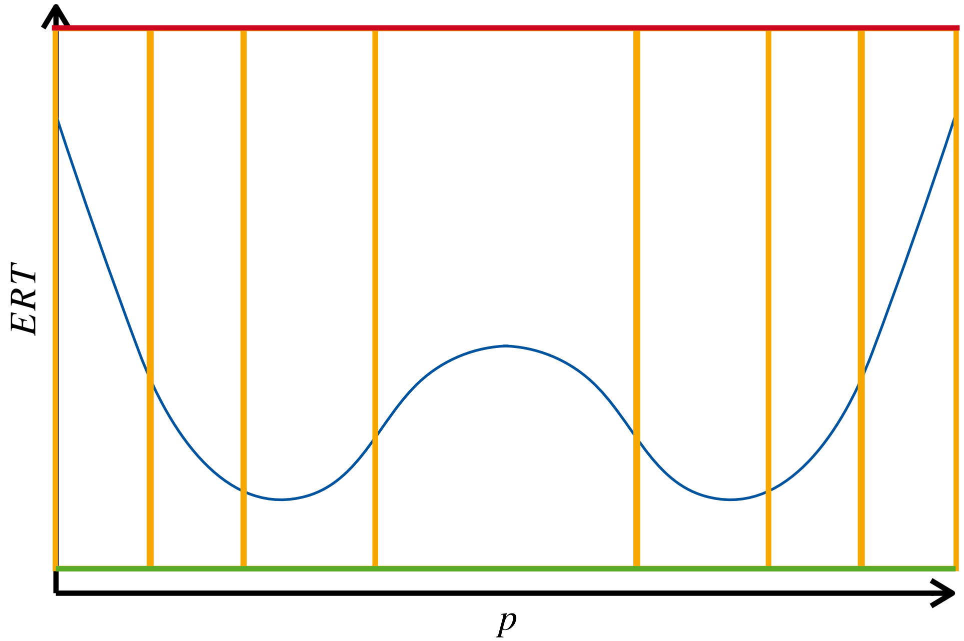
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



Parameter synthesis via PLA

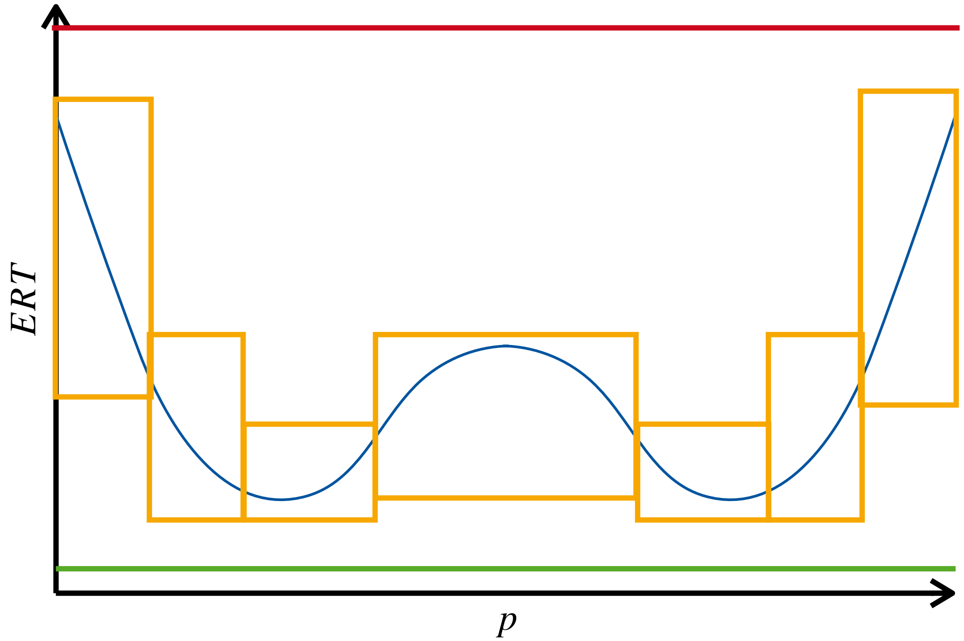
- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



TU/e

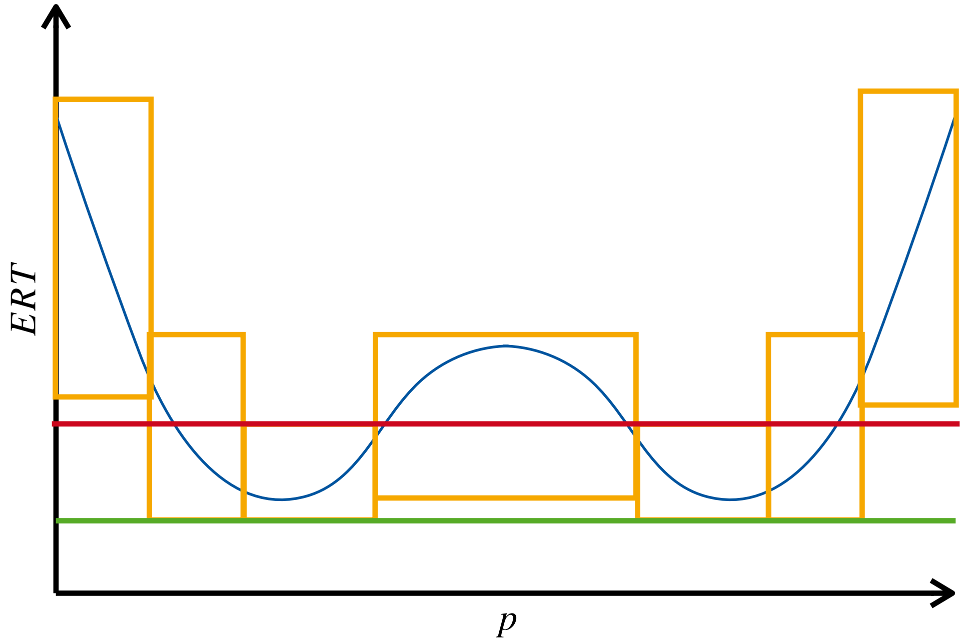
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



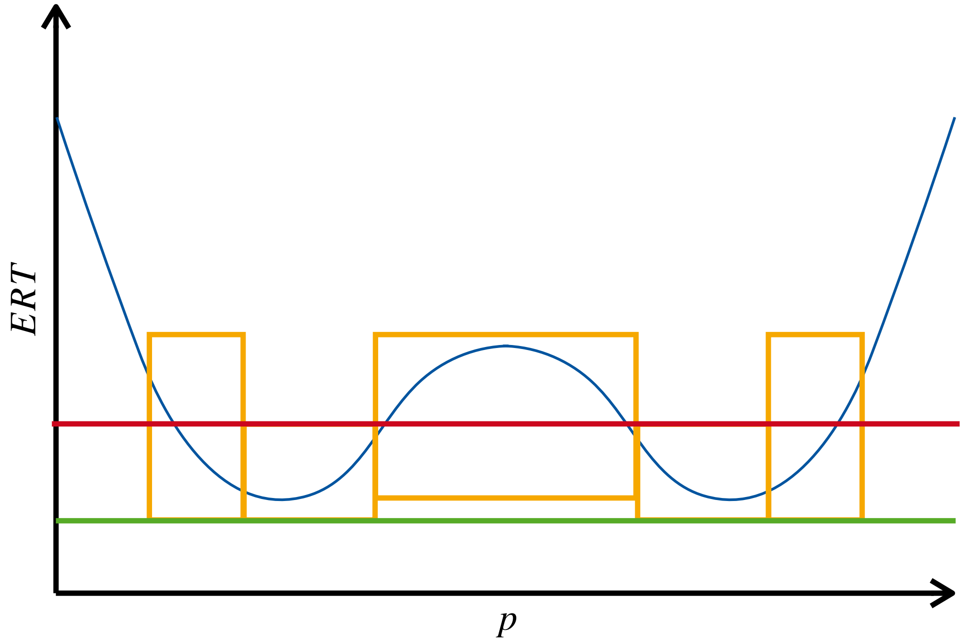
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



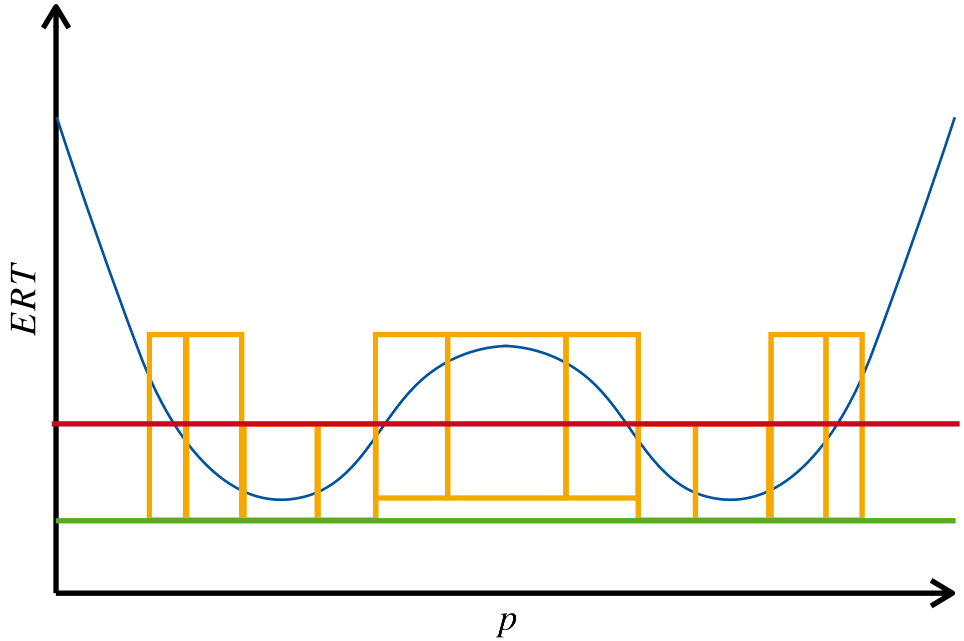
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



Parameter synthesis via PLA

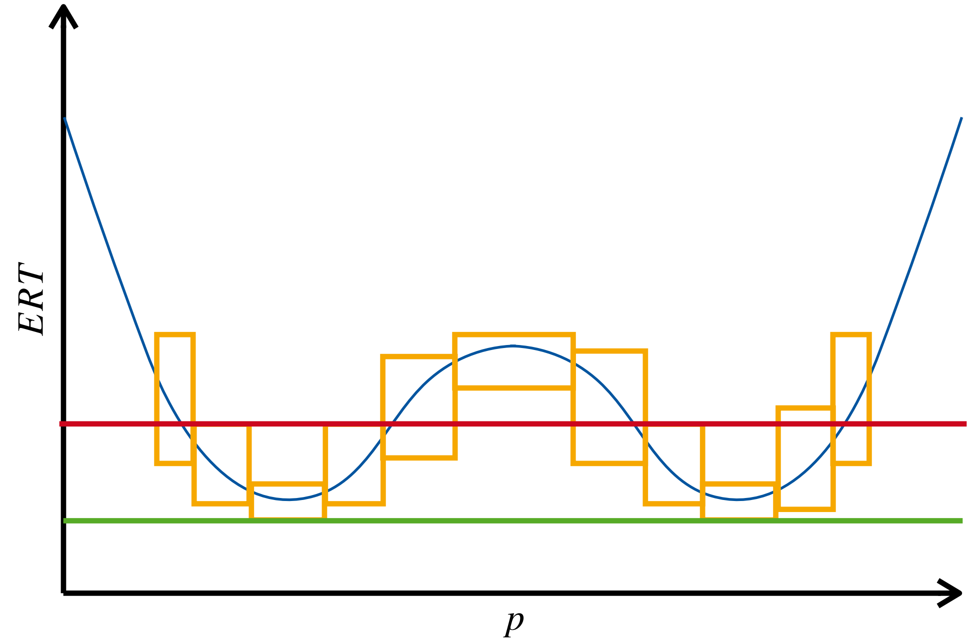
- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



TU/e

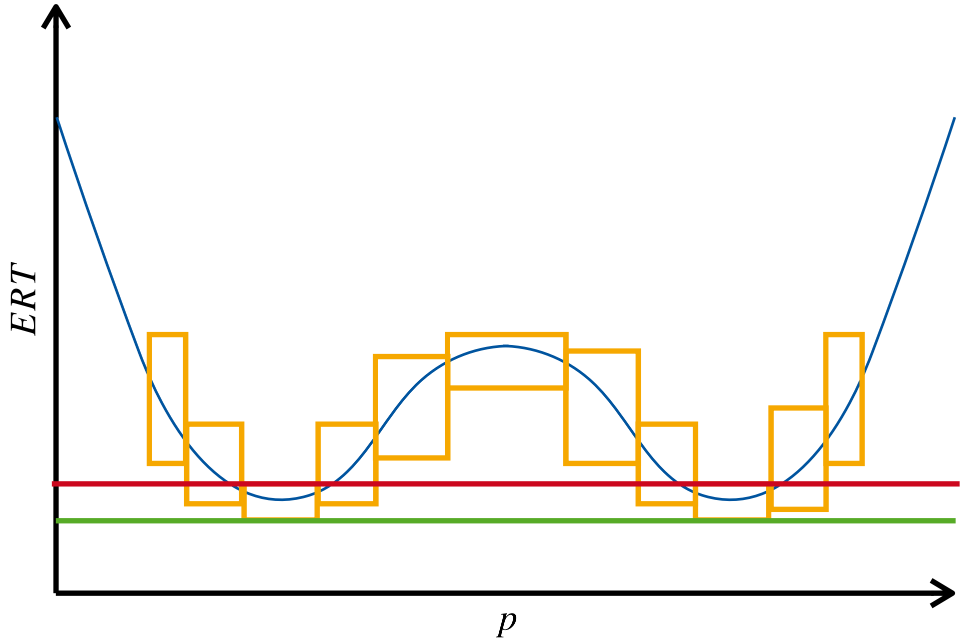
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



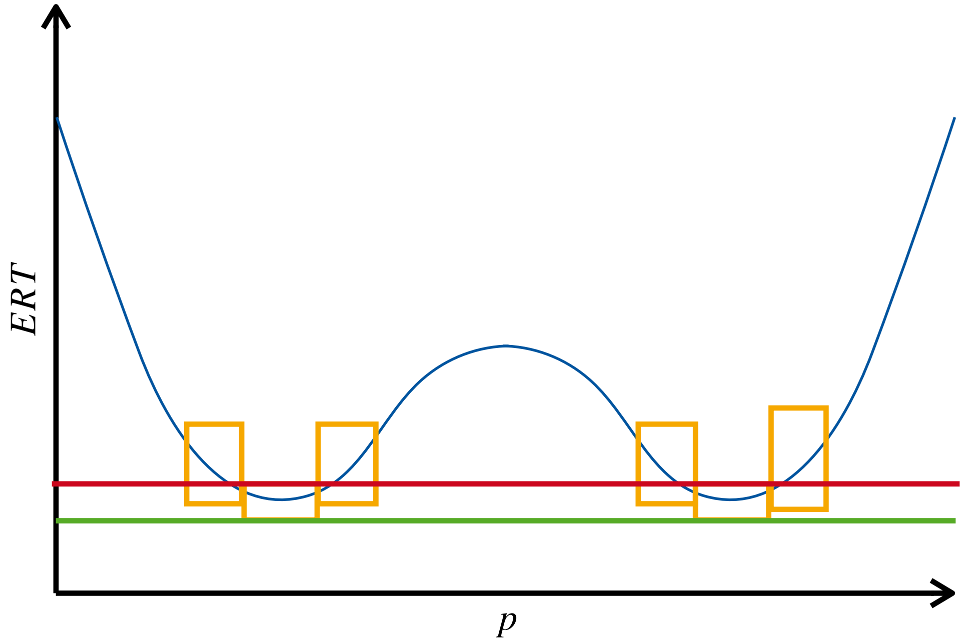
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



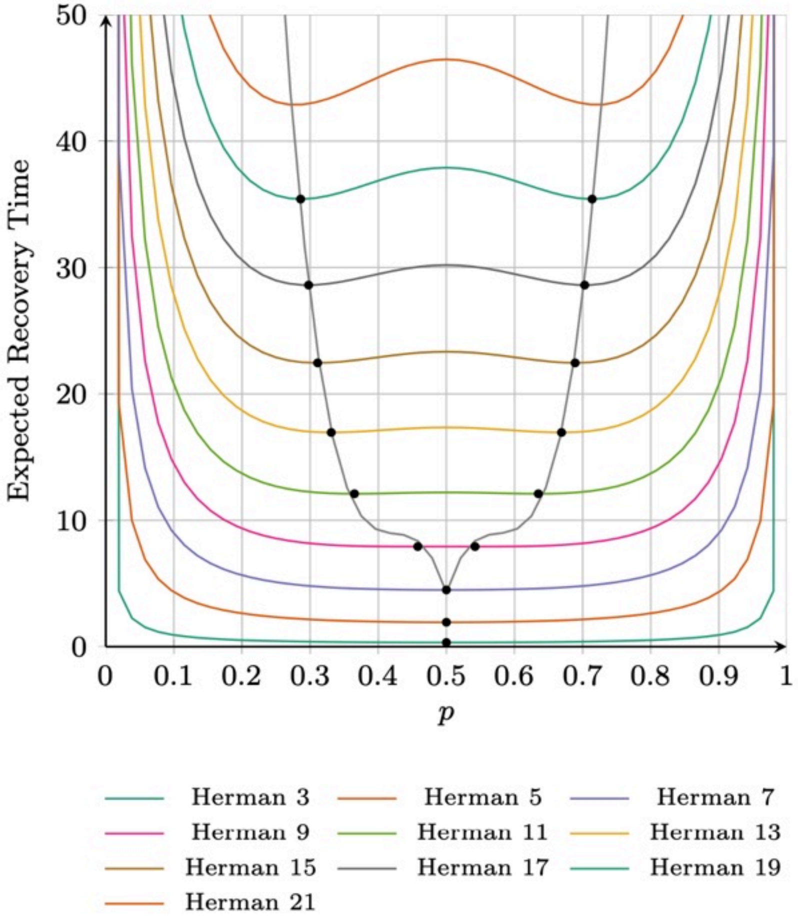
Parameter synthesis via PLA

- use Parameter Lifting Algorithm
- iteratively refine and tighten regions containing optimal parameter values



Optimal bias for Herman's Protocol

[Distributed Comput., 2022]



the **bias of the coin** should increase for increasing number of processes

19 processes:

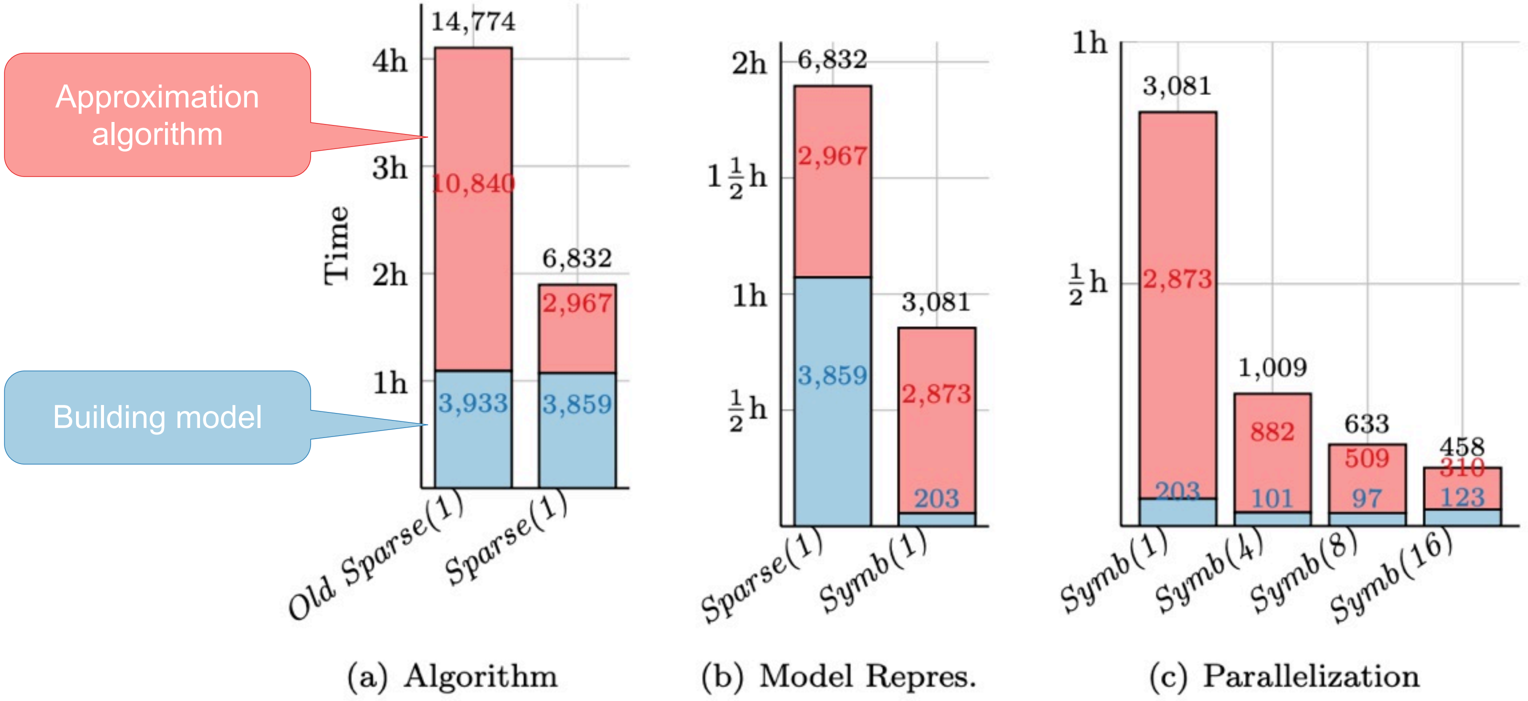
- states: $2^{19} = 524{,}288$
- transitions: > 1 billion
- time: 135 min
- optimal recovery time guaranteed in $[35.406, 35.416]$ for optimal bias in $[0.279, 0.292]$ and $[0.708, 0.721]$

TU/e

Optimized algorithms and tooling

[Distributed Comput., 2022]

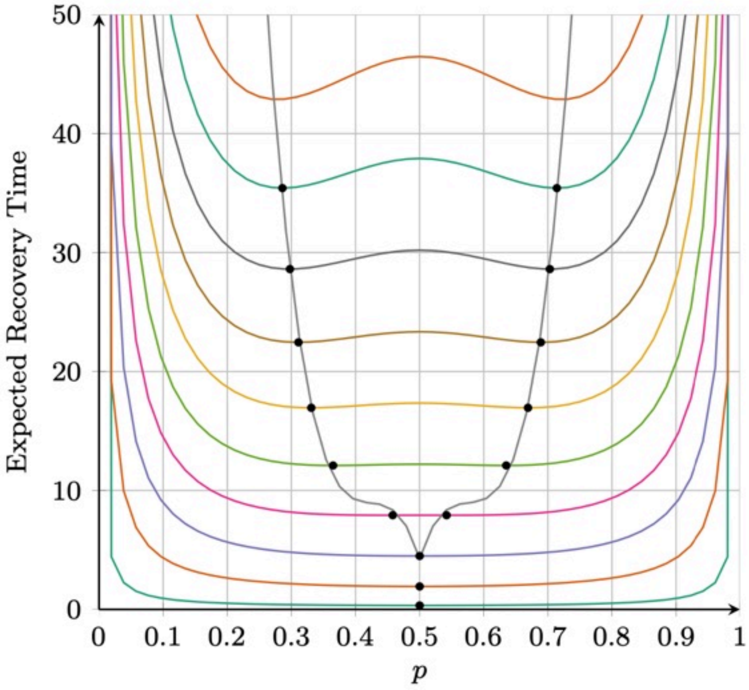
17 processes



TU/e

Take-home message

- parameter synthesis allows to obtain optimal parameters for randomized algorithms
- efficient algorithms and tooling tailored to use case can achieve good scalability



Uncertainty in continuous-time Markov chains

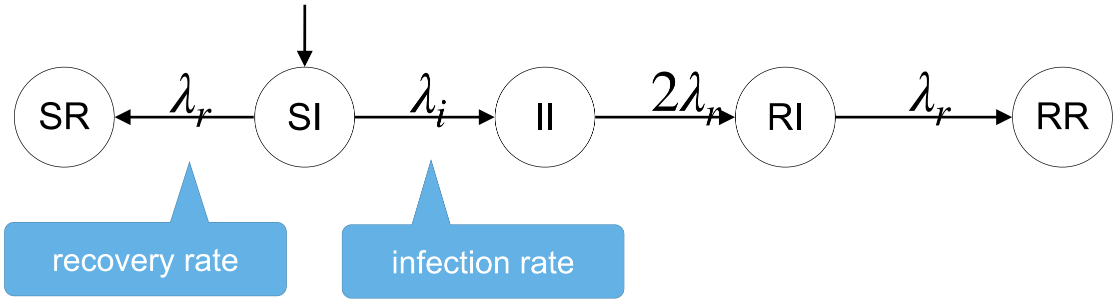
so far: **discrete-**time model

how to incorporate time?
→ **continuous**-time model

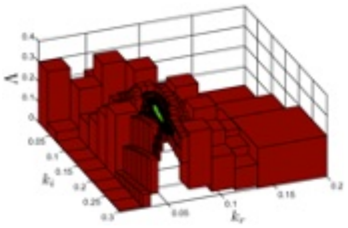
Joint work with Thom Badings, Nils Jansen,
Sebastian Junges and Marielle Stoelinga

Uncertain parametric CTMCs

- Parametric Continuous-time Markov chain (pCTMC):



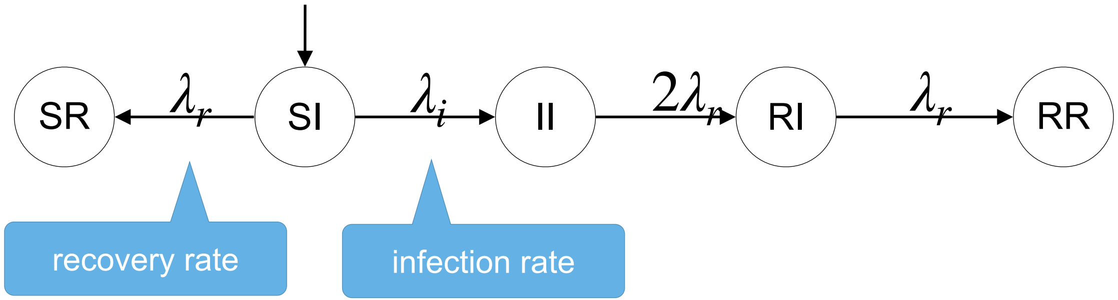
- Apply parameter synthesis?
  - conservative: find all parameter valuations which satisfy measure
  - limited to few parameters



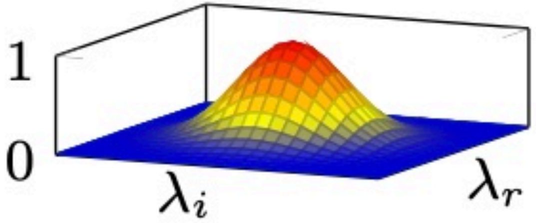
[Češka et al., 2017]

Uncertain parametric CTMCs

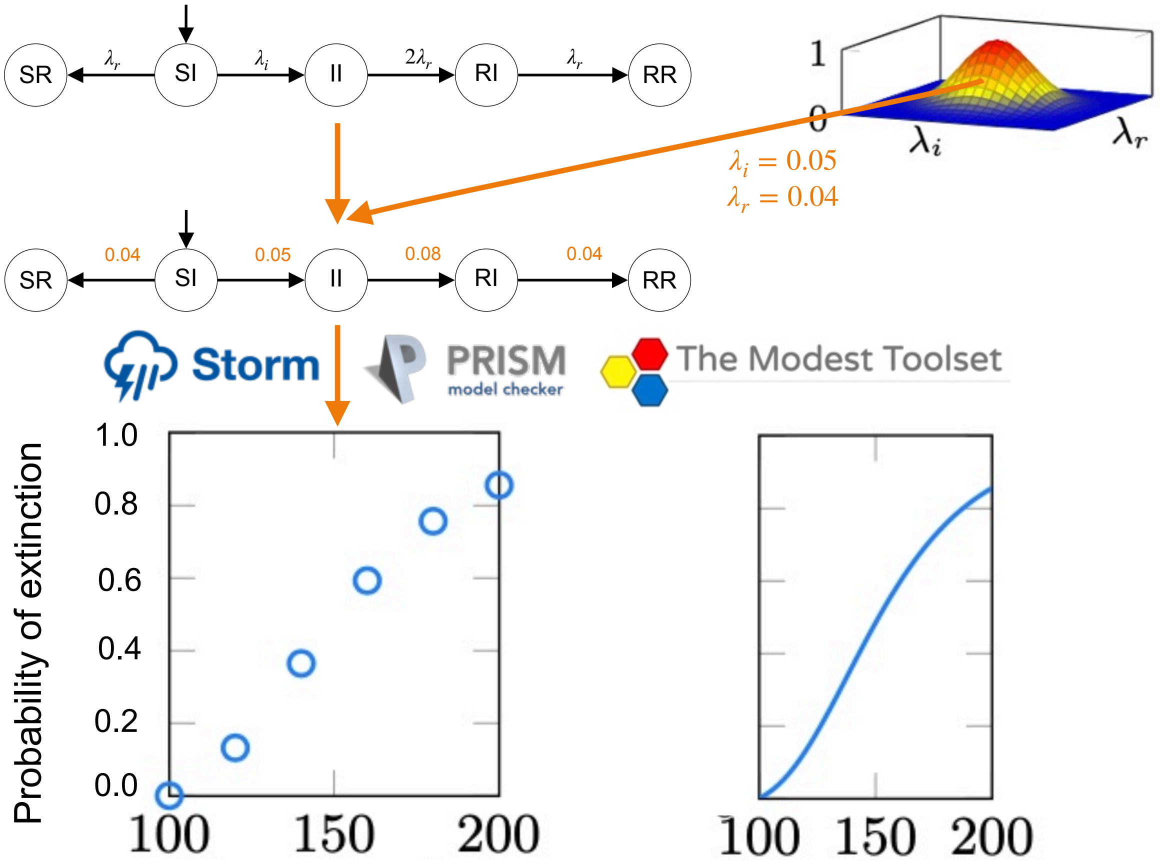
- Parametric Continuous-time Markov chain (pCTMC):



- Uncertain pCTMC (upCTMC):
  - pCTMC + probability distribution over parameter space
  - get rates from observations (historical data, experts, …)
  - only sampling access needed, no assumption on distribution



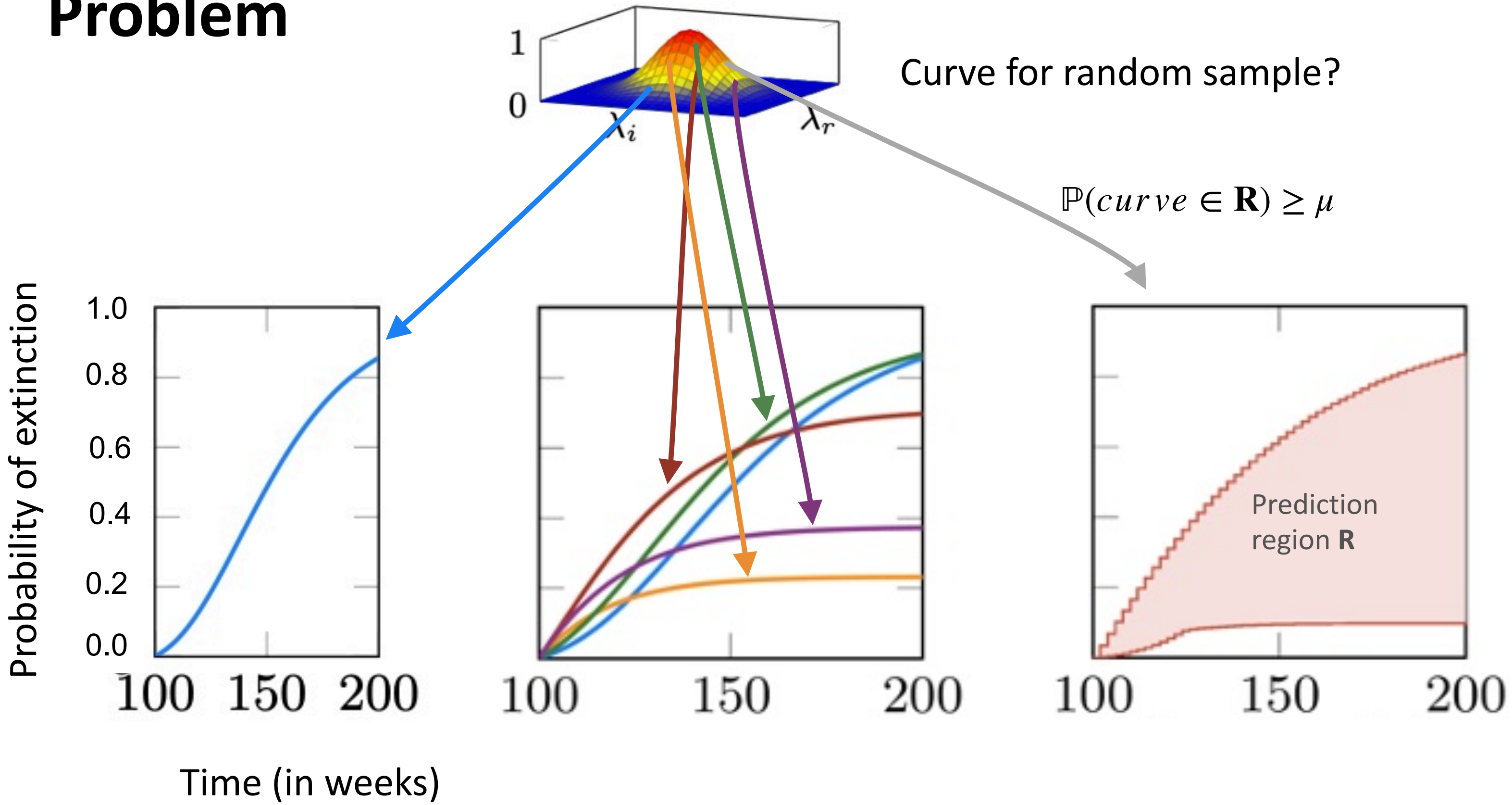
Problem



What is the probability that the disease becomes extinct within X weeks?

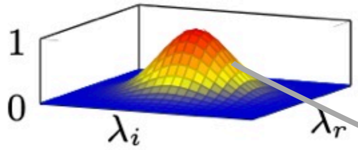
TU/e

Problem



1
0
λ_i
λ_r
Curve for random sample?
$\mathbb{P}(curve \in \mathbf{R}) \geq \mu$
Probability of extinction
1.0
0.8
0.6
0.4
0.2
0.0
100 150 200
100 150 200
100 150 200
Prediction region **R**
Time (in weeks)

Problem



Curve for random sample?

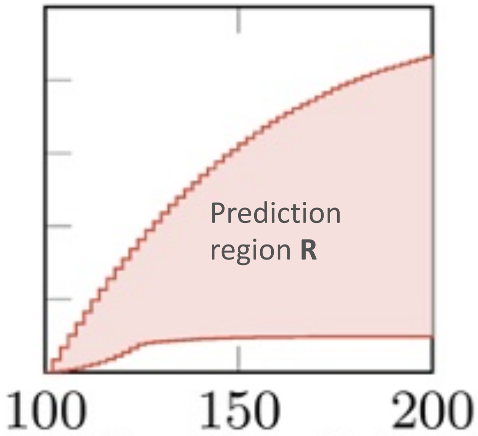
$\mathbb{P}(curve \in \mathbf{R}) \geq \mu$

Given:

- a **pCTMC**
- a **probability distribution** over parameters
- a set of **measures** Φ

Find:

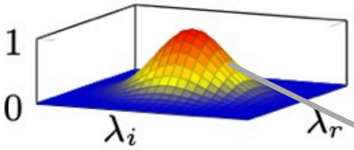
1. (tight) prediction region **R**
2. (tight) lower bound **μ** on containment probability



such that:

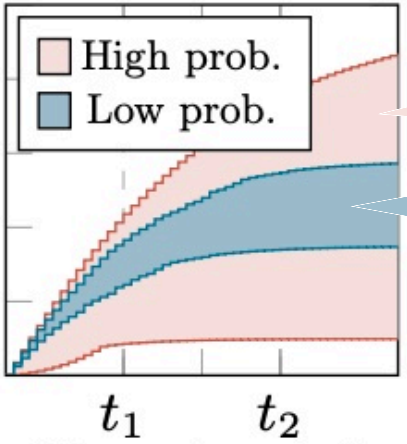
$\mathbb{P}(curve \in \mathbf{R}) \geq \mu$ holds with (user-specified) confidence level of at least **β**

Problem



Curve for random sample?

$\mathbb{P}(curve \in \mathbf{R}) \geq \mu$



containment probability is **79.4%** (with **confidence** of 99%)

containment probability is **7.5%** (with **confidence** of 99%)

Given:

- a **pCTMC**
- a **probability distribution** over parameters
- a set of **measures** Φ

Find:

1. (tight) prediction region **R**
2. (tight) lower bound **μ** on containment probability

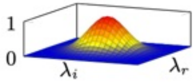
such that:

$\mathbb{P}(curve \in \mathbf{R}) \geq \mu$ holds with (user-specified) confidence level of at least **β**

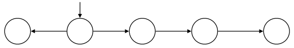
Slurf Approach

[CAV, 2022]

1. Distribution over parameters



2. pCTMC



3. Measures Φ
4. Confidence level β

Sample N parameter values

For each sample compute solution (curve)

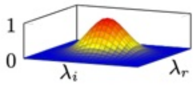
Find prediction region → **Prediction region R**

Find lower bound on containment probability → **Lower bound μ** on **containment probability**

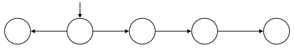
Slurf Approach

[CAV, 2022]

1. Distribution over parameters



2. pCTMC



3. Measures Φ

4. Confidence level β

Sample N parameter values → **For each sample compute solution (curve)** → **Find prediction region** → **Find lower bound on containment probability**

Find prediction region → **Prediction region R**

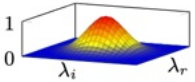
Find lower bound on containment probability → **Lower bound μ** on **containment probability**

TU/e

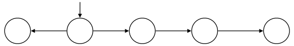
Slurf Approach

[CAV, 2022]

1. Distribution over parameters



2. pCTMC



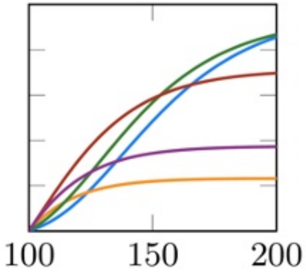
3. Measures Φ

4. Confidence level β

Sample N parameter values

For each sample compute solution (curve)

model checking



Find prediction region

Prediction region R

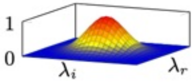
Find lower bound on containment probability

Lower bound μ on **containment probability**

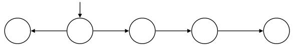
Slurf Approach

[CAV, 2022]

1. Distribution over parameters



2. pCTMC



3. Measures Φ

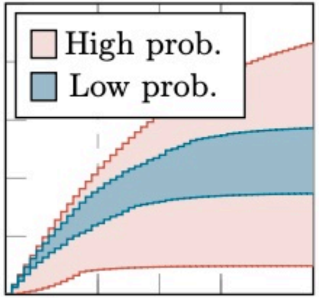
4. Confidence level β

Sample N parameter values

For each sample compute solution (curve)

Find prediction region

Find lower bound on containment probability



Prediction region R

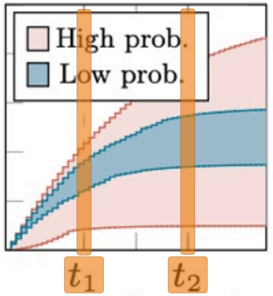
Lower bound μ on **containment probability**

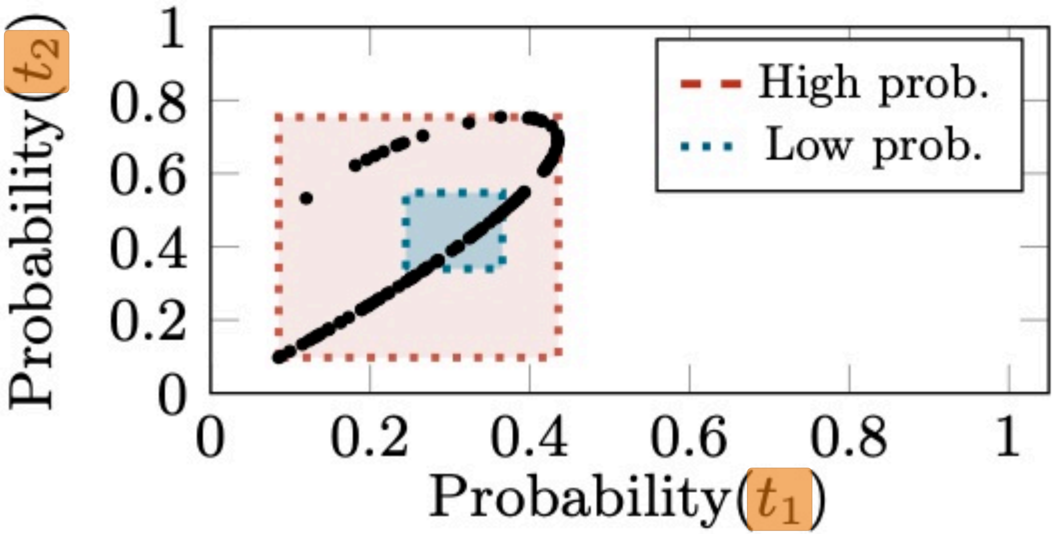
Use scenario optimization

TU/e

A Change of perspective

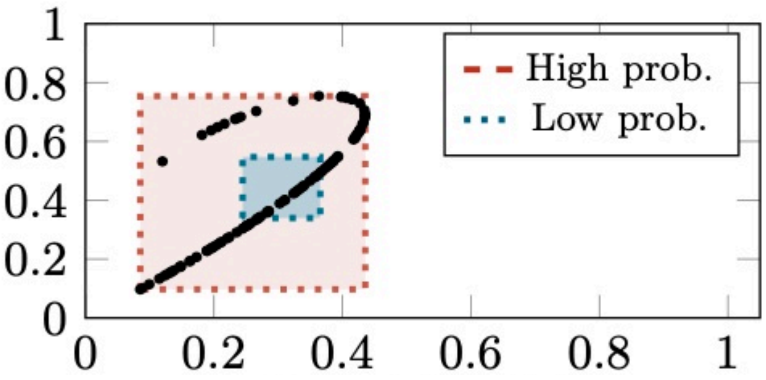
- Time bounds are multiple measures Φ
- Sample points in $\mathbb{R}^{|\Phi|}$





Scenario optimization

- Two goals:
  1. Minimize size of region R
  2. Minimize distance of sample points to R
     - sample points within R have distance 0
     - otherwise use Manhattan distance
- Encode as convex optimization problem



size of R

cost of relaxation

distance to R

$$\mathfrak{L}_{\mathcal{U}}^{\rho}: \text{ minimize } \quad \|\bar{x} - \underline{x}\|_1 + \rho \sum_{i=1}^{n} \|\xi_i\|_1$$

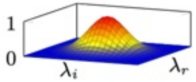
$$\text{subject to } \quad \underline{x} - \xi_i \leq \mathsf{sol}(u_i) \leq \bar{x} + \xi_i \quad \forall i = 1, \ldots, n.$$

TU/e

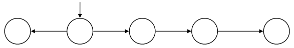
Slurf Approach

[CAV, 2022]

1. Distribution over parameters



2. pCTMC



3. Measures Φ

4. Confidence level β

Sample N parameter values

For each sample compute solution (curve)

Find prediction region

Prediction region R

Find lower bound on containment probability

Lower bound μ on **containment probability**

Estimate complexity

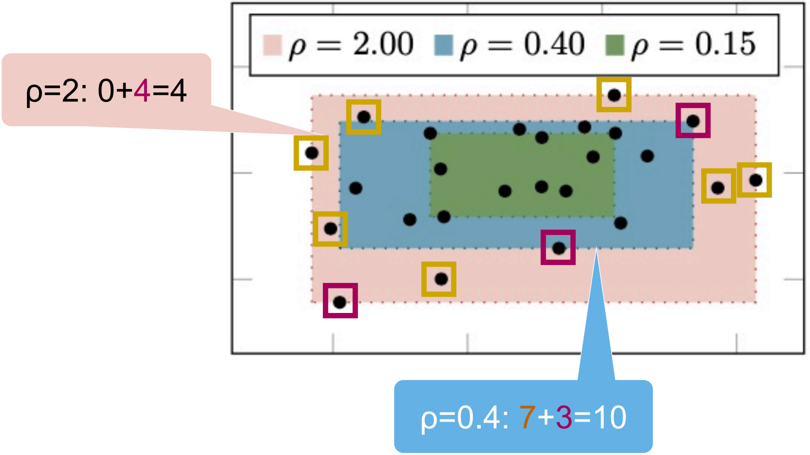
Complexity

- Complexity of a region:

  Number of sample points not in R

  \+ Number of samples needed on boundary of R to keep solution unchanged

Complexities depending on ρ



Lower bound

- Use complexity c of region to estimate lower bound on containment probability

Prediction region

lower bound for containment probability

confidence level

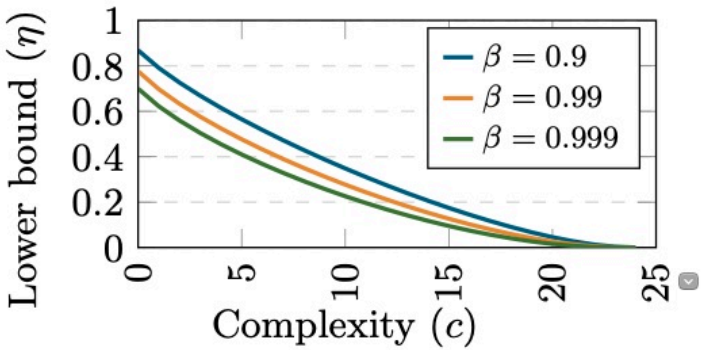
$$\mathbb{P}^n\{\text{contain}(R) \geq \eta(c)\} \geq \beta$$

η(c) is smallest (positive real-valued) solution to following polynomial inequality in variable t

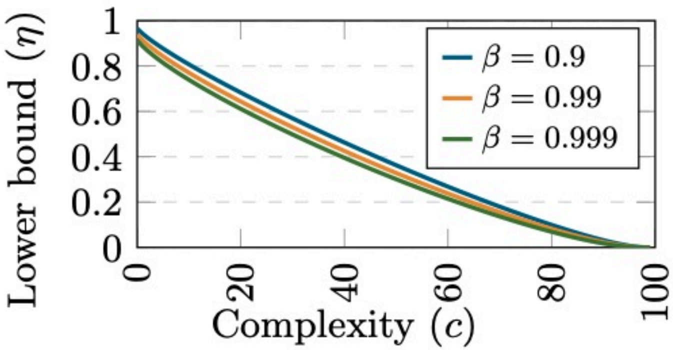
$$\binom{n}{c} t^{n-c} - \frac{1-\beta}{2n} \sum_{i=c}^{n-1} \binom{i}{c} t^{i-c} - \frac{1-\beta}{6n} \sum_{i=n+1}^{4n} \binom{i}{c} t^{i-c} = 0.$$

Lower bound

- Use complexity c of region to estimate lower bound on containment probability



(a) Number of samples $n = 25$.

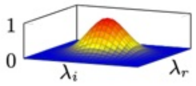


(b) Number of samples $n = 100$.

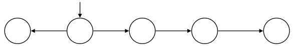
Slurf Approach

[CAV, 2022]

1. Distribution over parameters



2. pCTMC



3. Measures Φ

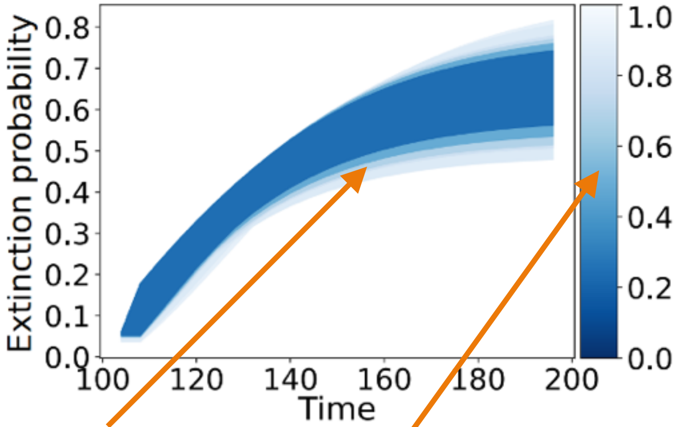
4. Confidence level β

Sample N parameter values

For each sample compute solution (curve)

Find prediction region → **Prediction region R**

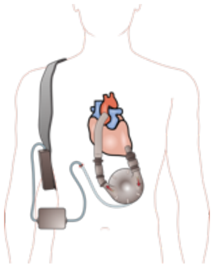
Find lower bound on containment probability → **Lower bound μ on containment probability**

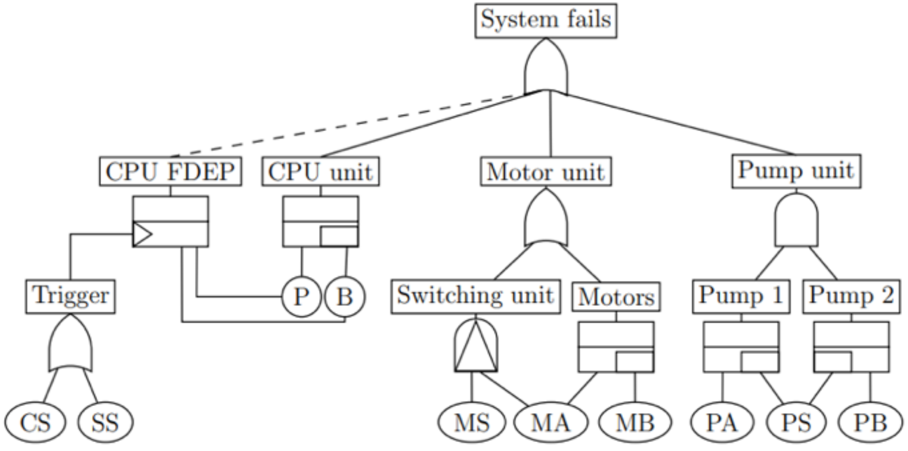


Example

[CAV, 2022]

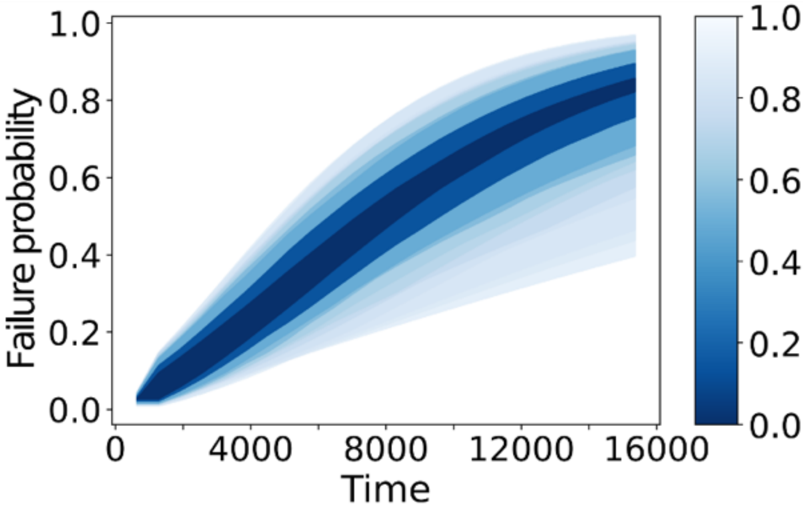
- dynamic fault tree for cardiac assist system
- failures rates assumed normally distributed





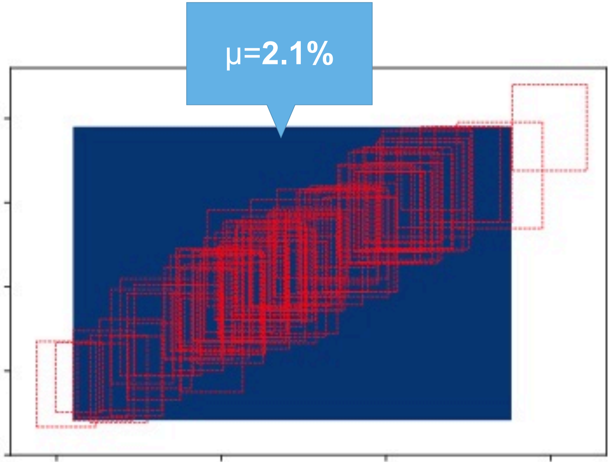
[Arnold et al., 2013]

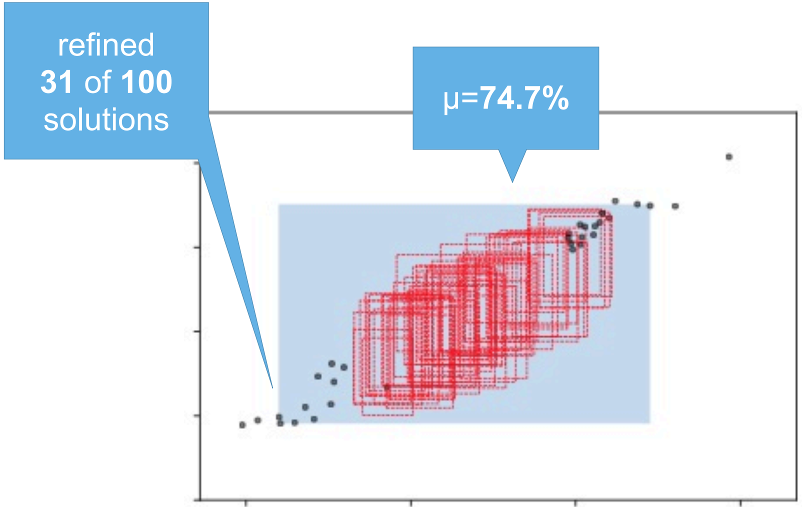
200 samples



Imprecise sampling

- Imprecise (approximate) samples:
  - faster to compute
  - give upper and lower bounds
- need to refine all imprecise sample points on the boundary





TU/e

Evaluation

single threaded, 3.7Ghz, 64GB RAM

computing sample points more expensive than scenario optimization

		Model size			Storm run time [s]		Scen.opt. time [s]	
benchmark	$\|\Phi\|$	#pars	#states	#trans	Init.	Sample ($\times$100)	$N = 100$	$N = 200$
SIR (140)	26	2	9 996	19 716	0.29	2947.29	18.26	63.27
SIR (140)[a]	26	2	9 996	19 716	0.29	544.27	25.11	129.66
Kanban (3)	4	13	58 400	446 400	4.42	46.95	2.28	6.69
Kanban (5)	4	13	2 546 432	24 460 016	253.39	4363.63	2.03	5.94
polling (9)	2	2	6 912	36 864	0.64	22.92	2.13	6.66
buffer	2	6	5 632	21 968	0.48	20.70	1.21	4.15
tandem (31)	2	5	2 016	6 819	0.11	862.41	5.19	24.30
rbc	40	6	2 269	12 930	0.01	1.40	5.27	16.88
rc (1,1)	25	21	8 401	49 446	27.20	74.90	5.75	20.34
rc (1,1)[a]	25	21	n/a[b]	n/a[b]	0.02	2.35	29.23	150.61
rc (2,2)[a]	25	29	n/a[b]	n/a[b]	0.03	27.77	24.86	132.63
hecs (1,1)[a]	25	5	n/a[b]	n/a[b]	0.02	9.83	26.78	145.77
hecs (2,2)[a]	25	24	n/a[b]	n/a[b]	0.02	194.25	33.06	184.32

Evaluation

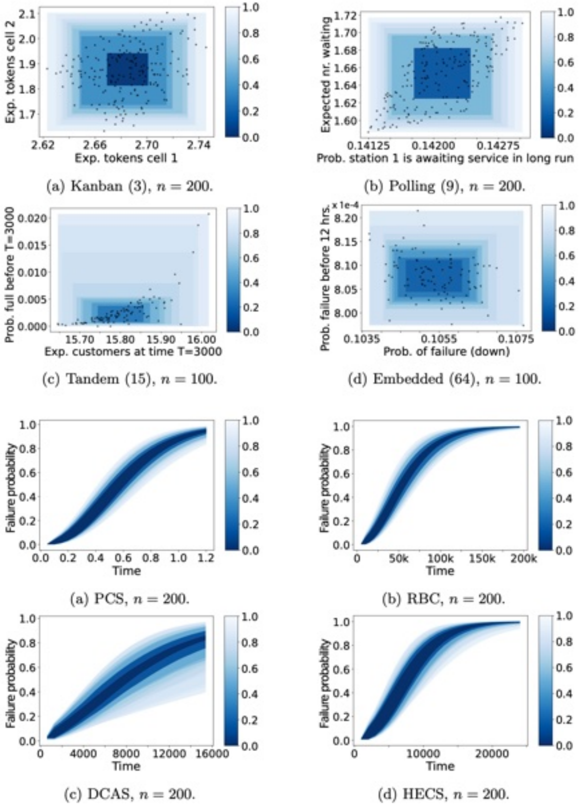
single threaded, 3.7Ghz, 64GB RAM

		Model size			Storm run time [s]		Scen.opt. time [s]	
benchmark	$\|\Phi\|$	#pars	#states	#trans	Init.	Sample ($\times$100)	$N = 100$	$N = 200$
SIR (140)	26	2	9 996	19 716	0.29	2947.29	18.26	63.27
SIR (140)[a]	26	2	9 996	19 716	0.29	544.27	25.11	129.66
Kanban (3)	4	13	58 400	446 400	4.42	46.95	2.28	6.69
Kanban (5)	4	13	2 546 432	24 460 016	253.39	4363.63	2.03	5.94
polling (9)	2	2	6 912	36 864	0.64	22.92	2.13	6.66
buffer	2	6	5 632	21 968	0.48	20.70	1.21	4.15
tandem (31)	2	5	2 016	6 819	0.11	862.41	5.19	24.30
rbc	40	6	2 269	12 930	0.01	1.40	5.27	16.88
rc (1,1)	25	21	8 401	49 446	27.20	74.90	5.75	20.34
rc (1,1)[a]	25	21	n/a[b]	n/a[b]	0.02	2.35	29.23	150.61
rc (2,2)[a]	25	29	n/a[b]	n/a[b]	0.03	27.77	24.86	132.63
hecs (1,1)[a]	25	5	n/a[b]	n/a[b]	0.02	9.83	26.78	145.77
hecs (2,2)[a]	25	24	n/a[b]	n/a[b]	0.02	194.25	33.06	184.32

approximate model checker significantly faster

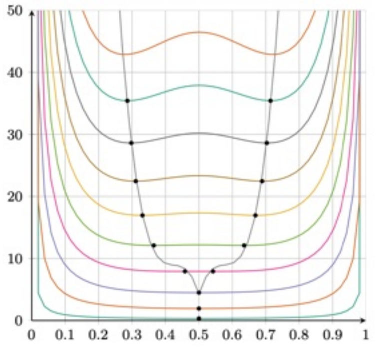
Take-home message

- **SLURF** is a scenario-optimization approach for parametric CTMCs with probability distribution over parameters
- High-confidence results can be given based on a few hundred sample points

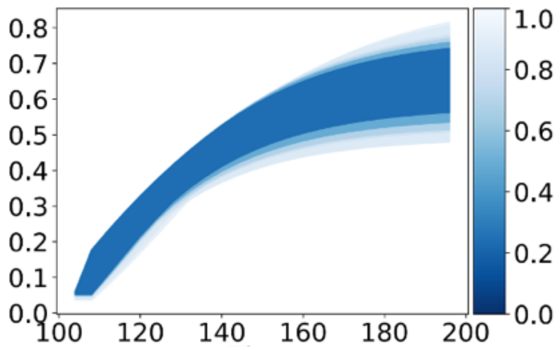


Conclusion

- parametric Markov chains allow for modeling uncertainty
- analysis finds
  exact/approximative
  all/one/optimal valuations



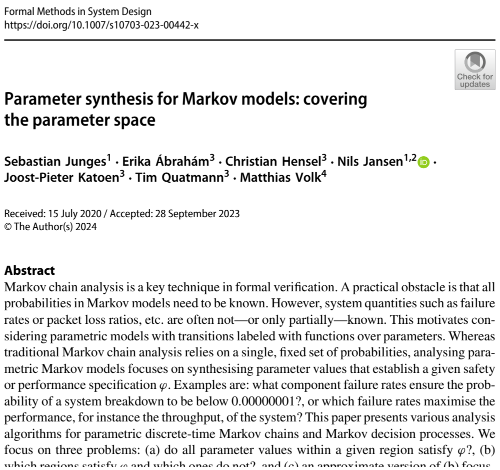
- probability distribution over parameters yields confidence regions for pCTMCs



TU/e

More Information

- **Literature:**
  Junges, Ábrahám, Hensel, Jansen, Katoen, Quatmann, Volk:
  „Parameter synthesis for Markov models: covering the parameter space"


Formal Methods in System Design
https://doi.org/10.1007/s10703-023-00442-x

Check for updates

Parameter synthesis for Markov models: covering the parameter space

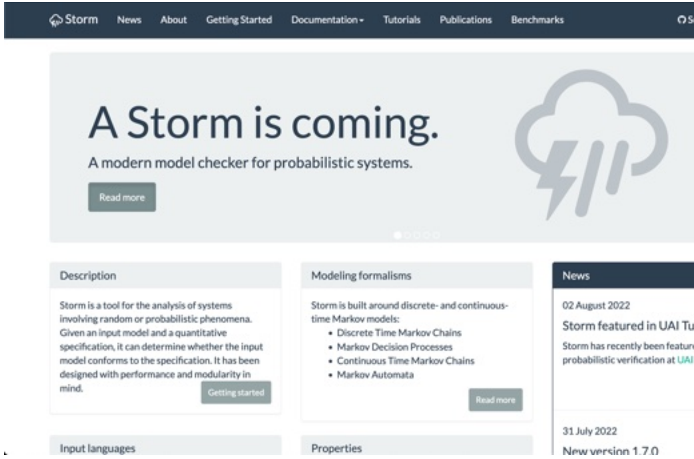
Sebastian Junges[1] · Erika Ábrahám[3] · Christian Hensel[3] · Nils Jansen[1,2] · Joost-Pieter Katoen[3] · Tim Quatmann[3] · Matthias Volk[4]

Received: 15 July 2020 / Accepted: 28 September 2023

Abstract
Markov chain analysis is a key technique in formal verification. A practical obstacle is that all probabilities in Markov models need to be known. However, system quantities such as failure rates or packet loss ratios, etc. are often not—or only partially—known. This motivates considering parametric models with transitions labeled with functions over parameters. Whereas traditional Markov chain analysis relies on a single, fixed set of probabilities, analysing parametric Markov models focuses on synthesising parameter values that establish a given safety or performance specification φ. Examples are: what component failure rates ensure the probability of a system breakdown to be below 0.00000001?, or which failure rates maximise the performance, for instance the throughput, of the system? This paper presents various analysis algorithms for parametric discrete-time Markov chains and Markov decision processes. We focus on three problems: (a) do all parameter values within a given region satisfy φ?, (b)

- Probabilistic model checking tool Storm

  https://www.stormchecker.org



- Contact: m.volk@tue.nl

TU/e