Structured Analysis (Overview)

Structured Analysis of Markov Chains

PART II: Structured Analysis Techniques

P. Buchholz
Informatik IV, Universität Dortmund, Germany
(partially based on joint work with
Tugrul Dayar, Peter Kemper, ...)

- ⇨ Compact Matrix Representations (Matrix Diag. – Hierarchical Kronecker)
- ⇨ Overview Solution Techniques
- ⇨ (Exact) Numerical Analysis
- ⇨ Simulative Techniques
- ⇨ Approximations
- ⇨ Tools
- ⇨ Conclusions

Structured Analysis (Compact Matrix Representations)

Available Matrix Representations
based on Kronecker Operations

- Modular Kronecker Representation
- Matrix Diagrams
- Hierarchical Kronecker Representation

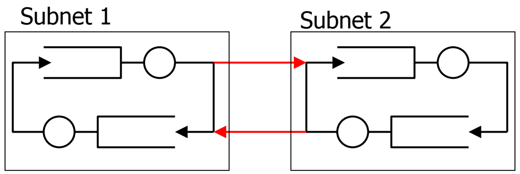
With unreachable states
should only be used in combination with functions testing reachability!!

Without unreachable states
natural representations for almost all (state based) analysis methods!!

Structured Analysis (Compact Matrix Representations)

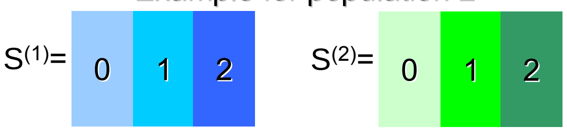
Hierarchical Kronecker representation by an example:

Example of a QN specified via hierarchical models

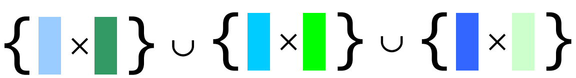


Distribution of population N (one class case)
0,..,N per submodel
⇒ Macro state space with N+1 macro states
{(0,N),...,(N,0)}

Example for population 2



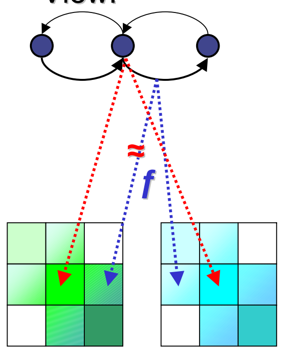
Reachable combinations



Structured Analysis (Compact Matrix Representations)

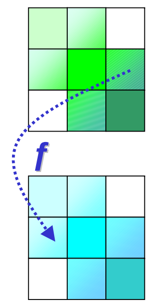
Matrix Representations
(local matrices are combined via Kronecker operations):

Hierarchical View:



$$Q = \begin{pmatrix} Q_{11} & \cdots & Q_{1N} \\ \vdots & \ddots & \vdots \\ Q_{N1} & \cdots & Q_{NN} \end{pmatrix}$$

Compositional View:



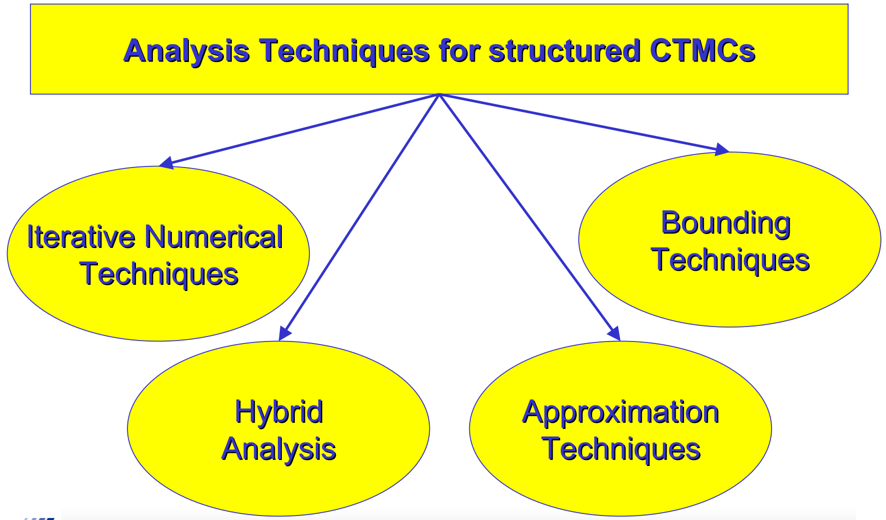
Structured Analysis (Compact Matrix Representations)

Generation of the hierarchical representation

- From the model specification
  E.g., in closed queueing networks by considering subnet population, in SPNs by computing regions, …
- From the description as an automaton by an algorithm computation of the coarsest representation for a given decomposition in automata

➢ **Both approaches have been implemented and work very well in practice**

Structured Analysis (Overview Techniques)



Structured Analysis (Numerical Techniques)

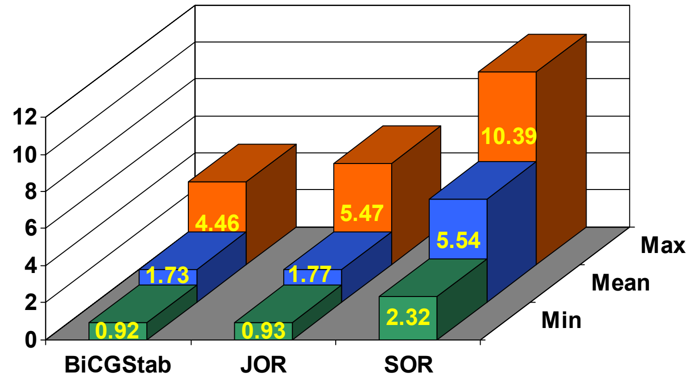
Basic problem computation of $\pi Q = 0$

For large state spaces:

- generate ***Q*** and represent it in compact form (as shown no problem for state space sizes of 10^{10} and beyond)
- compute **p** with an iterative solution technique
  Problems:
  - memory requirements grow linearly in the state space size
  - computation of vector matrix products is expensive
  - numerical stability of the used method might be a problem

Structured Analysis (Numerical Techniques)

➢ Price for the compact matrix representation in iteration time
(measured as the relation between hierarchical Kronecker and sparse matrix representation for 10 different example models)



Structured Analysis (Numerical Techniques)

For methods which exploit vector matrix products
(like JOR, BiCGStab)

- The mean slow down is less than 2
- In some lucky cases a small speedup is observed
  (more lucky cases occur for discrete time models with simultaneous events)
- The maximum slow down (about 5) results from an availability model with local failures and synchronized repairs among all processes (worst case example)
  – all other models result in slow downs of less than 3

For SOR is the effort per iteration increased by a factor of about 5.5
(for sparse matrices no slow down occurs if SOR is used)

Structured Analysis (Numerical Techniques)

Sometimes much larger slow downs are reported in the literature, what are the reasons?

- Use of a wrong representation
  – including non reachable states
  – causing a lot of overhead
- Inefficient implementation of structured techniques
  – approaches are hard to implement (many pointers, …)
  – implementation is done as student work whereas sparse matrix implementations are highly optimized

Structured Analysis (Numerical Techniques)

Since sparse matrix and Kronecker representations describe different ways to compute vector matrix products, numerical results will not be identical!

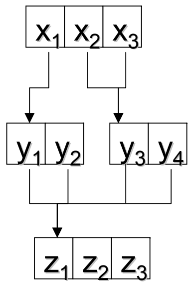
Our observations:

- No approach yields more accurate results in general
- In all examples the number of iterations for JOR, SOR and GMRES are identical
- Differences in the number of iterations occur in BiCGStab and TFQMR
  – Usually the differences are small,
  – but in a few examples significantly different numbers of iterations have been observed

Structured Analysis (Numerical Techniques)

What about compact representations of vectors?

General idea
(like for matrices, like for state spaces, ...):



Values $x_1y_1z_1$, ..., $x_3y_4z_3$

This and similar data structures have been tried, but the breakthrough is missing since

- **data structures remain compact only for very symmetric models**
- **performance of vector matrix computations is terrible**
- **numerical stability is questionable**

Current state for exact structured numerical analysis (with known numerical methods):

1. systems with 10^7-10^8 states can be solved (on a PC)
2. solution times may become huge
3. convergence is not always observed

Suggestions to solve the problems

- 2 and 3 use new numerical methods which exploit the matrix structure for analysis
- 1,2 and 3 use approximation or simulation techniques

Can we gain from the model structure also for the solution?

- Much less work for solution has been done than for representation
- Structure of the model often corresponds to a behaviorally oriented decomposition and can be exploited for solution
- Most solvers for CTMCs exploit some structure
  (e.g., A/D method, ML, ..)
- First experience with newly developed solvers for structured representations are very encouraging

There is much potential for the development of efficient numerical solution algorithms for compact matrix representations

Preconditioning techniques for structured representations

(Preconditioner has to be represented in a compact form)

Solve $\pi \boldsymbol{MQ}=\boldsymbol{0}$ instead of $\pi \boldsymbol{Q}=\boldsymbol{0}$

- Separable preconditioner resulting from the inversion of component matrices (Buchholz NSMC 99)
  - Approximation of the Neumann expansion of the matrix
  - Compact representation
  - Solution effort is reduced only for some examples
- Nearest Kronecker product preconditioner (Stewart/Langville J. on Num. Lin. Alg. 02)
  - Approximation of $\boldsymbol{Q}$ by a single Kronecker product which is easy to invert
  - Only proprietary implementation for a restricted class of models available
  - First results show improved solution times, but results for large models are missing

- BSOR for structured matrices
  (Buchholz/Dayar NSMC 03)
  - Hierarchical representations naturally define blocks, additional blocks due to Kronecker structure
  - Solution of blocks by LU- or Shur-decomposition
  - Efficient solver for a large class of models
- BSOR as preconditioner for projection methods
  (Buchholz/Dayar SIAM J. Sci. Comp. 04)
  - BSOR is applied to the residual vector of projection methods yielding a new preconditioner
  - Efficient solver for a large class of models

Idea of BSOR:

- Define diagonal block submatrices Q_{II} and solve iteratively $\pi_I^{(k)} Q_{II} = b_I^{(k)}$ where $-b_I^{(k)} = \Sigma_{J<I}\, \pi_J^{(k)} Q_{JI} + \Sigma_{K>I}\, \pi_J^{(k-1)} Q_{KI}$
- Approach can be defined in a nested way over several levels

Approach is known for a long time, what is the specific role of compact matrix representation?

- Kronecker representation of non-diagonal blocks and those diagonal blocks that are solved iteratively
- Exploitation of specific structure of diagonal blocks for direct solution:
  - Several blocks are of the form $Q_{II} = B + \lambda_I I$ for some matrix **B**, i.e. identical diagonal blocks up to the diagonal elements!

Solution of $\pi_I Q_{II} = b_I$ where $Q_{II} = B + \lambda_I I$

1. Compute Shur-factorization $B = ZTZ^T$ for quasi-triangular matrix T (i.e. block triangular with diagonal blocks of order 1 or 2) and orthogonal matrix Z (i.e. $ZZ^T = I$)
2. Compute $c_I = b_I Z$
3. Solve $y_I(T + \lambda_I I) = c_I$
4. Compute $\pi_I = y_I Z^T$

Remarks

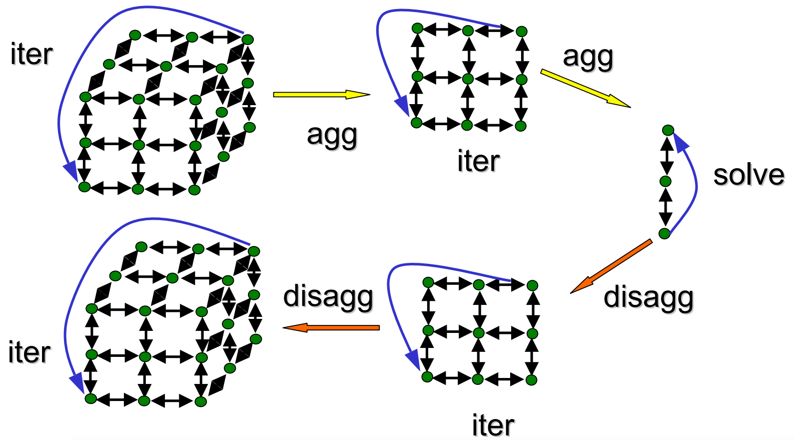
- Step 1. needs to be done only once per matrix B and resulting matrices Z and T need to be stored only once
- 2. und 4. are vector matrix products
- 3. is a quasi-upper triangular solve

- Aggregation/Disaggregation for structured representations
  (Buchholz EJOR 99, APNUM 99)
  - Aggregation/Disaggregation steps at a block level as in methods for sparse matrices
  - Additional aggregation at a component level
    - Projection of the current solution vector for a single component
    - Solution of an aggregated system and correction of the current overall solution
    - Combination with standard iterative solver (JOR, SOR, ..)
  - Efficient implementation due to exploitation of Kronecker structure
  - Improved solution times if components are loosely coupled

- Multi-level solution for structured systems
  (Buchholz SIGMAX 00, Buchholz/Dayar Computing 04)
  - Method uses ideas from multigrid methods
  - Generation of aggregated system with respect to the model structure by aggregation of components
  - Iteration at different levels
  - Multiplicative projection and correction functions
  - Very efficient implementation due to exploitation of Kronecker structure
  - Aggregated matrix results from detailed matrix after removing some matrices in the Kronecker representation
    (no explicit generation or storage of aggregated system)
  - Extremely efficient solvers for most models scalable to huge systems

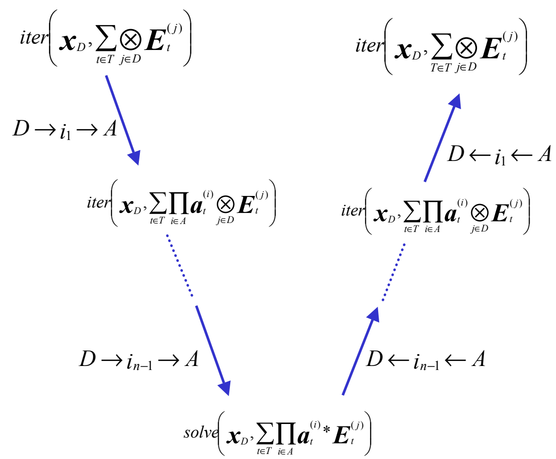
Structured Analysis (Numerical Techniques)

Shematic description of multi-level for one macro state
(extension to multiple macro states work similar)



Structured Analysis (Numerical Techniques)

Natural integration in Kronecker representation



- Examples shows V-cycle other cycles from multigrid can as well be realized
- Different realizations
  - different iteration methods
  - different stopping criteria
- Efficient implementation requires some effort

Structured Analysis (Numerical Techniques)

<u>Larger models </u>(not solvable with sparse matrix techniques):

10 different models with

- between 358,560 and 2,945,880 states
- between 1,871,004 and 26,172,344 transitions

Hierarchical Kronecker representation with

- between 1 and 1774 macro states/blocks
- between 370 and 11,480 non-zero elements

- 4 models for communication systems/protocols (msmq1/2, courier1/2)
- 4 models from the manufacturing area (kanban1/2/3, fms)
- 1 model from computer system modeling (ncd,qh-realcontrol)
- 2 models with ncd property (ncd, kanban3)

Structured Analysis (Numerical Techniques)

Experiment conditions

- Iterations are stopped if
  - The maximum norm of the residual vector is less than 10^{-8} or
  - 7000 seconds of CPU time elapsed
- Methods are ranked for each example and the mean rank for each method is determined

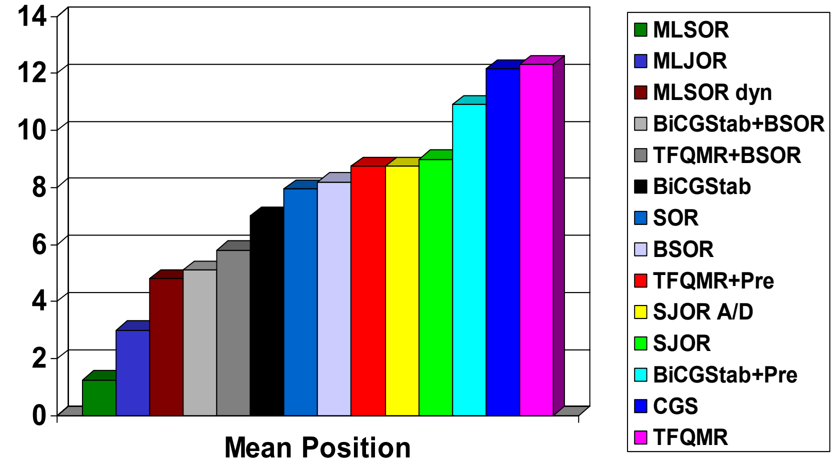
Structured Analysis (Numerical Techniques)

Empirical comparison using 14 different structured solution methods:

- JOR, SJOR, SOR and SJOR A/D
- BSOR
- BiCGStab without/with BSOR- or separable-precond.
- TFQMR with and without BSOR- or separable-precond.
- CGS
- Multi-Level with JOR or SOR and fixed number of iterations
- Multi-level with SOR and dynamic number of iterations

Structured Analysis (Numerical Techniques)

- Comparison of the methods concerning their rank



Structured Analysis (Numerical Techniques)

Some observations

- Advanced methods outperform standard methods clearly
- ML-methods scale better than BSOR
  MLSOR is the clear winner
  - Solution of all models with MLSOR in at most 12 minutes
- No difference between NCD and non-NCD examples
- Projection methods without BSOR-preconditioner show convergence problems on several examples

Advanced solution techniques are helpful

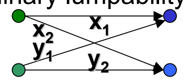
- **they reduce solution times (reduction factor model dependent but often > 5)**
- **they do not increase the size of solvable model**

Structured Analysis (Exact Aggregation)

Reduction of the state space by exact aggregation:

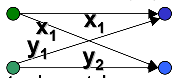
Reduction of components
(i.e. substitution of component matrices by smaller aggregate matrices yielding exact results)

- Ordinary lumpability (performance bisimulation...)



lumpable if $x_1 = y_1$ and $x_2 = y_2$

- Exact lumpability (inverse performance bisimulation)



lumpable if $x_1 = y_1$ and $x_2 = y_2$

aggregated matrices can be computed with an an effort that grows with the size of the component

- Symmetric components
  (harder to integrate in Kronecker representations, but still possible (see e.g., Derisavi, Kemper, Sanders NSMC 03)

Structured Analysis (Other Approaches)

Many practically relevant models are still much too large to be solved numerically
but generator matrix *Q* can be represented in compact form (see the examples in the first part of this talk!)

Can we exploit knowledge of ***Q*** without storing all state probabilities ?

Yes, different possibilities exist!

- bounding methods
- hybrid simulation – numerical analysis
- approximation techniques

will be considered in this tutorial

Structured Analysis (Hybrid Simulation)

Numerical Analysis

- generation of the state space
- computation of all state probabilities
- results are exact up to numerical precision (no stochastic fluctation)
- state space explosion

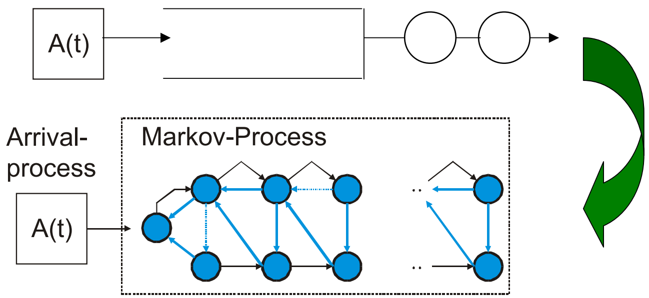
Discrete Event Simulation

- observation of some sample path
- statistical evaluation of result measures
- results depend on the observed behavior (as part of the complete behavior)
- applicable independently of the state space size

How to combine both approaches?

Structured Analysis (Hybrid Simulation)

Example G/PH/1/K Queue



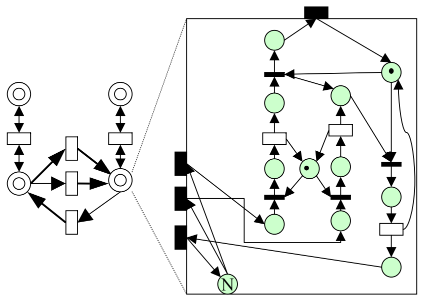
Generation of the CTMC for the service process

Analysis idea:

1. Sample next arrival at time $t+\Delta$
2. Compute probability distribution at the server at time $t+\Delta$ with knowledge of the distribution at time t
3. Compute distribution after the arrival
4. Set $t = t + \Delta$ and continue with 1.

Structured Analysis (Hybrid Simulation)

Generalization for GSPNs-QNs:



Hybrid Algorithm

- Determine potential firing times of synchronized transitions via simulation
- Compute local distribution between firing points numerically (via randomization)
- Decide (via simulation) whether the synchronized transition firing is real or pseudo
- Compute local distributions after real/pseudo firing of synchronized transition

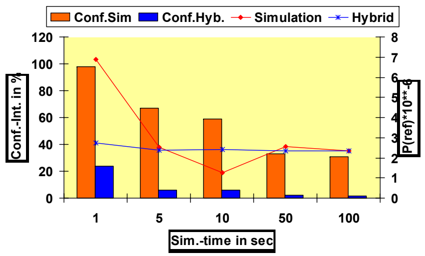
Applicable for transient and stationary analysis!

Structured Analysis (Hybrid Simulation)

Experiences
- Efficient for several models with loose coupling and/or small probabilities
  - smaller confidence intervals than simulation
  - larger model class than numerical analysis
  - no approximation due to decomposition

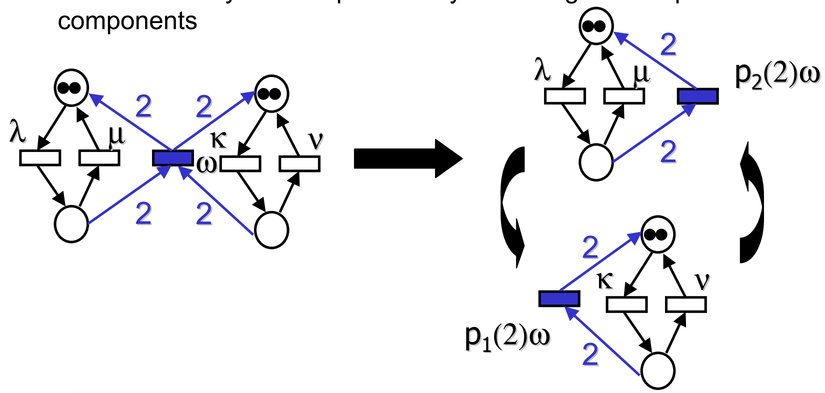
Example results for a model of a mobile communication system

- Loose coupling between cells due to small handover probabilities
- Computation of (small) blocking probabilities



Structured Analysis (Approximation)

Many approximation approaches are based on fixed point computations
- Decompose the system into parts
- Define relations among parts (in form of input-output parameters)
- Solve iteratively the components by assuming fixed exports from other components



Structured Analysis (Approximation)

- Approaches of this type are used for a long time
- May be first formalization by Ciardo, Trivedi 1991
- Still a lot of open questions
  - Do we have convergence?
  - Is convergence unique?
  - How good are the results?
  - …
- In general non-zero probabilities can be assigned to non-reachable states!

⇒ Combine fixed point ideas with compact matrix representation to avoid unreachable states!

Structured Analysis (Approximation)

- Ciardo, Donatelli, Miner Sigmetrics 2000
  - Integration with matrix diagrams
  - Mapping of states onto state space of one component
  - Computation of probabilities only for reachable state
- Buchholz ATPN 1998
  - Aggregation of other components due to bisimulation (functional, nearly lumpable, lumpable)
  - Computation of results for aggregated system considers only reachable states
  - Varying degree of approximation
- Buchholz DSN 2002
  - Representations of vectors as Kronecker products
  - Simultaneous computation of the fixed point in the reachable state space
  - Varying degree of approximation

Structured Analysis (Approximation)

Compact representation of vectors by Kronecker products:

$$p[\tilde{x}] \approx \Pr(x) \bigotimes_{j=1}^{J} \bar{p}_j[\tilde{x}(j)]$$

component vectors

vector decomposition per macro state for the hierarchical representation (similar representation for matrix diagrams are possible)

where

- $\Pr(x) = p[\tilde{x}]e^T$ probability of macro state
- $\bar{p}_j[\tilde{x}(j)]$ vector of conditional state probabilities

representation is compact but only an approximation

Structured Analysis (Approximation)

Operations on Kronecker representations of vectors:

Transformation:

$$p[\tilde{x}] \rightarrow \Pr(x) \bigotimes_{j=1}^{J} \bar{p}_j[\tilde{x}(j)] \quad \textit{and} \quad p[\tilde{x}] \leftarrow \Pr(x) \bigotimes_{j=1}^{J} \bar{p}_j[\tilde{x}(j)]$$

Vector sum:

$$\Pr(x) \bigotimes_{j=1}^{J} \bar{p}_j[\tilde{x}(j)] \leftarrow \sum_{k=1}^{K} \Pr_k(x) \bigotimes_{j=1}^{J} \bar{p}_j^k[\tilde{x}(j)]$$

Vector matrix product:

$$\Pr(x) \bigotimes_{j=1}^{J} \bar{p}_j[\tilde{x}(j)] \leftarrow \Pr(x) \bigotimes_{j=1}^{J} \bar{p}_j[\tilde{x}(j)] \bigotimes_{j=1}^{J} E_j = \Pr(x) \bigotimes_{j=1}^{J} \bar{p}_j[\tilde{x}(j)]E_j$$

Structured Analysis (Approximation)

Operations

- can be naturally and efficiently implemented based on Kronecker operations
- require no allocation and deallocation of memory
- are numerically stable

Realization of the Power method (for transient analysis see paper at QEST):

$$\Pr^{(k+1)}(\tilde{x}) \bigotimes_{j=1}^{J} \overline{\boldsymbol{p}}_j^{(k+1)}[\tilde{x}] = \Pr^{(k)}(\tilde{x}) \bigotimes_{j=1}^{J} \overline{\boldsymbol{p}}_j^{(k)}[\tilde{x}] + \frac{1}{\alpha} \sum_{\tilde{y} \in MS} \sum_{t} W(t) \Pr^{(k)}(\tilde{y}) \Big(\bigotimes_{j=1}^{J} \overline{\boldsymbol{p}}_j^{(k)}[\tilde{y}] \boldsymbol{E}_t^j[\tilde{y}(j), \tilde{x}(j)] - \bigotimes_{j=1}^{J} \overline{\boldsymbol{p}}_j^{(k)}[\tilde{y}] \boldsymbol{D}_t^j[\tilde{y}(j), \tilde{x}(j)] \Big)$$

Structured Analysis (Approximation)

Memory requirements and effort per iteration in $O(\sum_{j=1}^{J} n_j)$

Approximate solution due to approximate vector representation

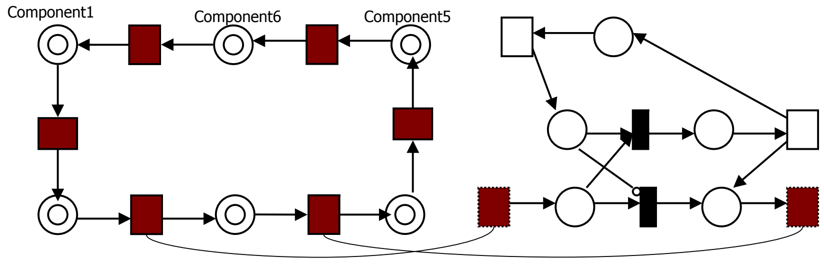
Source of approximation: $a \otimes b \approx (v \otimes w) + (x \otimes y)$

Optimal approximations exist in numerical analysis !

Quality of the approximation:

- exact results for product form models
- provably good approximation for loosely coupled systems
- often good results for general models (fixed point approach which often has been analyzed experimentally)

Example of a SGSPN specified via hierarchical nets (MSMQ):



Components are not loosely coupled

⇒ not a „best case“ for the approach

Results for the MSMQ example

- Power method with Kronecker matrix and detailed vector between 900 and 2500 seconds solution time, memory requirements about 18 Mbytes
- Power method with Kronecker matrix and Kronecker vector between 10 and 16 seconds solution time, memory requirements about 30 Kbytes

Relative errors when using Kronecker representation for vectors:

- server throughput less than 1.6%
- mean buffer population less than 3.2%
- blocking probability less than 15%

Results are satisfactory, but can they be improved?

Kronecker and detailed representation can be transformed

➡ **different representations for different subvectors!**

Operations to be performed:

- multiplication detailed vector with Kronecker matrix
- multiplication Kronecker vector with Kronecker matrix
- sum of detailed vectors
- sum of Kronecker vectors
- transformation detailed vector in Kronecker vector
- transformation Kronecker vector in detailed vector

➡ **any combination of normal and Kronecker representation can be realized**

Example 3×3 block matrix (here with 2 components):

$$Q = \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix}$$

*Pr(1)(**p**$_1$[1]⊗**p**$_2$[1])* (compact)

***p**[2]* (detailed)

*Pr(3)(**p**$_1$[3]⊗**p**$_2$[3])* (compact)

<u>Examples</u>

Multiplication with $Q_{12} = W(t)\left(E_t^1[0,1] \otimes E_t^2[2,1]\right)$

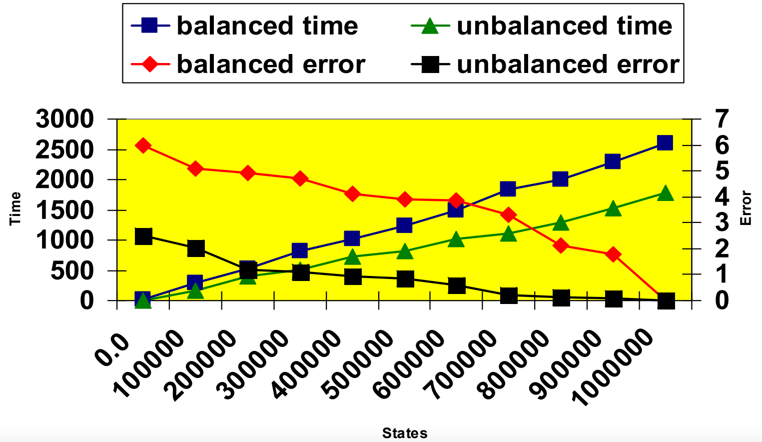
$$Pr(1)W(t)\left(p_1[1]E_t^1[0,1] \otimes p_2[1]E_t^2[2,1]\right) \to Pr'(1)\left(p'_1[2] \otimes p'_2[2]\right) \to p'[2]$$

Multiplication with $Q_{21} = W(t)\left(E_t^1[1,0] \otimes E_t^2[1,2]\right)$

$$p[2] \to Pr(2)\left(p'_1[2] \otimes p'_2[2]\right) \to$$
$$Pr(2)W(t)\left(p'_1[2]E_t^1[1,0] \otimes p'_2[2]E_t^2[2,1]\right) \to Pr'(1)\left(p'_1[1] \otimes p'_2[1]\right)$$

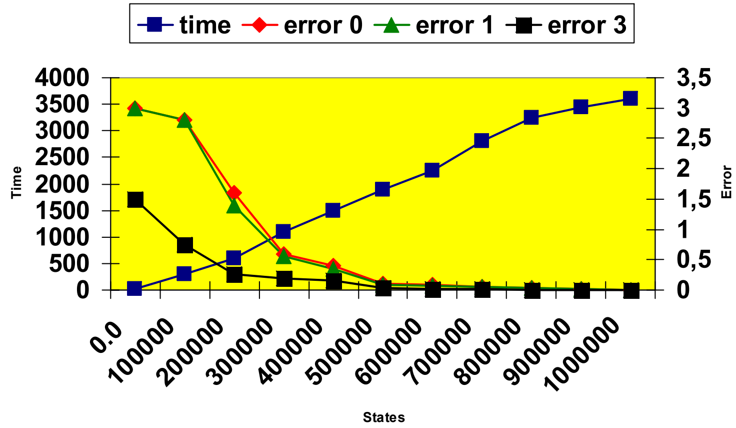
Structured Analysis (Approximation)

A symmetric MSMQ system with balanced and unbalanced load and at most one server per queue



Structured Analysis (Approximation)

An asymmetric MSMQ system with a heavily loaded station which can occupy all servers



Structured Analysis (Conclusions)

- Compact matrix representations exist for a large class of models
  - Problem of unreachable states in earlier approaches has been solved
    - Reinvention of a Kronecker representation for model class xyz without an implementation is not necessary and useless!
  - Matrix generation and functional analysis are extremely efficient and huge transition system can be generated and represented on contemporary PCs
    - parallel state space generation is not necessary!
  - Efficient implementations of vector matrix products are possible, but require a sophisticated design and implementation of data structures and algorithms

Structured Analysis (Conclusions)

- Model structure can be used to speed up solutions
- Although no black box method exist in general, multilevel methods seem to be a good candidates for reliable and efficient solvers
  - But still a lot of work remains to be done in this area
- On current PCs models with up to 10^7 states usually can be solved in a reasonable time
  - Out-of-core methods with sparse matrix representation are usually not competitive
  - Out-of-core methods with compact matrix representations and vectors partially stored on fast disks could be candidates to analyze larger models
- Development of approximation and bounding methods for structured matrix representations is still an important research topic