Conditioned Belief Propagation Revisited (Extended Version)

Thomas Geier, Felix Richter, Susanne Biundo

Ulmer Informatik-Berichte

Nr. 2014-03
Juni 2014

Conditioned Belief Propagation Revisited (Extended Version)

Thomas Geier, Felix Richter, Susanne Biundo
Ulm University, Germany
{thomas.geier, felix.richter, susanne.biundo}@uni-ulm.de

Belief Propagation (BP) applied to cyclic problems is a well known approximate inference scheme for probabilistic graphical models. To improve its accuracy, Conditioned Belief Propagation (CBP) has been proposed, which splits a problem into subproblems by conditioning on variables, applies BP to subproblems, and merges the results to produce an answer to the original problem. In this work, we propose a reformulated version of CBP that exhibits anytime behavior and allows for more specific tuning by formalizing a further aspect of the algorithm through the use of a leaf selection heuristic. We propose several simple and easy to compute heuristics and demonstrate their performance using an empirical evaluation on randomly generated problems.

1 Introduction

Belief Propagation (BP) (Pearl, 1986) works by sending messages between variables along the edges of the dependency graph (Bayesian network or Markov network). For acyclic problems the algorithm terminates after a number of message-passing steps that is linear in the size of the graph, producing exact results for the marginal probabilities of all variables at once. But when the graph contains loops, BP is no longer guaranteed to converge, and the results produced are no longer exact. However, during the 1990s, it became apparent empirically that BP is a very well performing approximate algorithm even for cyclic problems (Weiss, 1997). Some years later Yedidia et al. (2001a) improved our understanding of BP on a theoretical level by recognizing that the fixed points of the BP update equations are exactly the stationary points of a variational approximation problem long known in the physics literature. The work by Yedidia et al. also enables us to compute estimates of the partition function from the BP messages.

In this extended report[1] we describe a simple method of improving the approximation quality of BP. The basic idea was already formulated in Pearl (1986), who proposed to condition on variables to break loops. But instead of aiming to break all loops, we apply BP again to a now slightly less cyclic problem. This very idea was picked up in Eaton

[1] There exists a short version of this essay (Geier et al., 2014).

and Ghahramani (2009), who introduced the term Conditioned Belief Propagation (CBP) for it. They describe a very elaborate method of picking variables to condition on. In this paper, we formalize a version of CBP with an additional choice point we call "leaf selection", together with simpler, well performing heuristics.

Related to CBP are collapsed sampling methods (Koller and Friedman, 2009, p. 526, p. 645), which sample assignments to a subset of variables while solving the conditioned problem exactly (see for example Cycle-Cutset sampling (Bidyuk and Dechter, 2003)). In contrast to this, CBP systematically explores conditions, while solving the remaining problems approximately. Further, CBP can be regarded as a mixture model. See for example Jaakkola and Jordan (1998) for using mixtures of Mean field approximations for probabilistic inference. Closer to the idea of conditioning instead of mixing, while still using variational approximations, is the work by Bouchard and Zoeter (2009), who are using the approach for approximating integrals.

The reason why CBP appears to be an interesting inference algorithm is the way in which it combines two different approaches to inference. At its base it uses BP, which works very well on "smooth" problems, i.e. problems with a high entropy. But BP fails on low entropy portions of problems with strong to deterministic dependencies (Weiss, 1997), (Koller and Friedman, 2009, p. 428/429), and Figure 5. In contrast to this, systematic exploration by conditioning is the de facto approach for purely deterministic inference (SAT and CSP). Under the presence of determinism, conditioning enables us to prune portions of the search space (unit propagation) and reveals context-specific independence (Boutilier et al., 1996; Zhang and Poole, 1999). In combination with BP, conditioning can be used to eliminate low entropy portions of the problem. This leaves a high-entropy remainder that can be solved efficiently with BP.

We will go on with formalizing our notion of CBP. We suggest some simple heuristics (much simpler than the heuristic described in Eaton and Ghahramani (2009)), and provide a thorough empirical evaluation on generated problems. We finish with a discussion of further research directions that also includes an argument on why CBP, as it exists now, cannot readily be used for large problem instances.

2 Preliminaries

We denote sets of variables using bold letters ($\mathbf{X}, \mathbf{Y}, \dots$), and single variables using normal style ($X, Y, \dots$). The set of all variables is $\mathcal{X}$. For a set of variables $\mathbf{X}$, let $\mathrm{V}(\mathbf{X})$ be the set of all its *assignments*, which are functions mapping a variable $X \in \mathbf{X}$ to one of its finitely many values $\mathrm{Dom}(X)$, and let $\widetilde{\mathrm{V}}(\mathbf{X})$ be the set of partial assignments to $\mathbf{X}$. We denote (partial) assignments with lower case (Greek) letters. The set of all partial assignments is $\mathcal{A} := \widetilde{\mathrm{V}}(\mathcal{X})$. A *factor* $\phi : \mathrm{V}(\mathbf{X}_\phi) \to \mathbb{R}^+$ over a finite set of variables $\mathbf{X}_\phi$ maps an assignment to its variables to the non-negative reals. An inference *problem* Φ over variables $\mathbf{X}_\Phi$ is given by a finite set of factors. It maps an assignments $\mathbf{x} \in \mathrm{V}(\mathbf{X}_\Phi)$ to the non-negative reals by

$$\Phi(\mathbf{x}) := \prod_{\phi \in \Phi} \phi(\mathbf{x}_\phi). \tag{1}$$

Here $\mathbf{x}_\phi$ denotes the restriction of $\mathbf{x}$ to the variables $\mathbf{X}_\phi$ of factor ϕ. The set of all problems is $\mathcal{P}$. An inference problem defines an unnormalized distribution over the discrete random variables $\mathbf{X}_\phi$. The *partition function* Z_Φ of a problem Φ, is defined by

$$Z_\Phi \coloneqq \sum_{\mathbf{X}_\Phi} \Phi \coloneqq \sum_{\mathbf{x} \in \mathrm{V}(\mathbf{X}_\Phi)} \Phi_{\mathbf{x}}. \tag{2}$$

In this summation notation we sum over all assignments $\mathbf{x} \in \mathrm{V}(\mathbf{X}_\Phi)$ as argument to Φ, without explicitly naming them.

The *factor graph* of a problem Φ is a bipartite graph $G_\Phi \coloneqq \langle \mathbf{X}_\Phi \cup \Phi, E\rangle$ where variables and factors are vertices and there exists an edge between X and ϕ, if and only if $X \in \mathbf{X}_\phi$, i.e., factor ϕ depends on variable X.

Belief Propagation

In its interpretation as message passing on factor graphs (Kschischang et al., 2001), BP sends messages between factors and variables until convergence, which is not guaranteed. By doing so it attempts to minimize the Kullback-Leibler divergence between the Gibbs distribution of the factor product and a class of distributions parameterized by message values (Yedidia et al., 2001a). If the factor graph contains no loops, BP converges to the exact result in linear time. We will now go on and describe BP in more detail.

With each edge $\{X, \phi\}$ of the factor graph G_Φ, there are associated two messages, $\delta_{X\to\phi}$ and $\delta_{\phi\to X}$. Both messages are factors over the variable X. Belief Propagation then tries to calibrate these messages to achieve the following conditions for all pairs of variables and factors $\{X, \phi\}$ with $X \in \mathbf{X}_\phi$:

$$\delta_{\phi\to X} \propto \sum_{\mathbf{X}_\phi \setminus \{X\}} \phi \cdot \prod_{Y \in \mathbf{X}_\phi; Y \neq X} \delta_{Y\to\phi} \tag{3}$$

$$\delta_{X\to\phi} \propto \prod_{\psi \in \mathrm{Nb}(X), \psi \neq \phi} \delta_{\psi\to X} \tag{4}$$

Here, $\mathrm{Nb}(X)$ returns adjacent objects of a vertex X in G_Φ.

BP calibration works by iteratively updating the message values according to these equations, followed by normalizing the messages. The schedule for these updates plays an important role in the ability to achieve convergence (Koller and Friedman, 2009, pp. 407-411).

When BP converges and equations 3 and 4 hold, we can obtain estimates for the variable marginals $\beta_X \approx Z_\Phi^{-1}\Phi(X) = Z_\Phi^{-1} \sum_{\mathbf{X}_\Phi \setminus \{X\}} \Phi$, which we call variable beliefs; in a similar way we can obtain factor beliefs β_ϕ:

$$\beta_X \propto \prod_{\phi \in \mathrm{Nb}(X)} \delta_{\phi\to X} \tag{5}$$

$$\beta_\phi \propto \prod_{X \in \mathbf{X}_\phi} \phi \cdot \delta_{X\to\phi} \tag{6}$$

It is also possible to obtain an estimate on the partition function $Z_\Phi^{\mathrm{BP}} \approx Z_\Phi$ via the Bethe free energy approximation (Yedidia et al., 2001a), (Koller and Friedman, 2009, p. 414):

$$\ln Z_\Phi^{\mathrm{BP}} := \sum_{\phi\in\Phi} \mathbb{E}_{\beta_\phi}[\ln\phi] + \sum_{\phi\in\Phi} \mathbb{H}_{\beta_\phi} - \sum_{X\in\mathbf{X}_\Phi} (|\mathrm{Nb}(X)| - 1)\cdot \mathbb{H}_{\beta_X} \tag{7}$$

Here, $\mathbb{E}_{\beta_\phi}[\ln\phi]$ is the expected value of factor ϕ using its factor belief as probability measure and $\mathbb{H}_{\beta_\phi}$ and $\mathbb{H}_{\beta_X}$ are the entropies of the factor beliefs and variable beliefs, respectively.

3 Conditioned Belief Propagation

CBP is an inference algorithm for undirected graphical models over categorical random variables that yields an approximation to the partition function. CBP recursively applies conditioning to produce smaller subproblems on which BP hopefully performs better. The results to these subproblems can then be aggregated to obtain estimates of the partition function (and variable beliefs) of the original problem. If we decompose a problem Φ on some variable $X \in \mathbf{X}_\Phi$, it holds that

$$Z_\Phi = \sum_{\mathbf{X}_\Phi} \Phi = \sum_X \sum_{\mathbf{X}_\Phi\setminus\{X\}} \Phi = \sum_X Z_{\Phi[x]}, \tag{8}$$

where $\Phi[x]$ is the problem obtained by conditioning all factors in Φ on the assignment $x \in \mathrm{V}(X)$. This equation essentially constitutes the justification of the correctness of one step of conditioning. In a similar way we can also compute an estimate of the marginal probabilities using the variable beliefs of the conditioned problems.

The CBP algorithm decomposes a given problem Φ step by step. This forms a tree of partial assignments with the empty assignment as the root. For each inner node $\xi \in \widetilde{\mathrm{V}}(\mathbf{X}_\Phi)$, its children are all the assignments obtained by extending ξ by some assignments to a select variable X. Because at each stage of the algorithm only the leaf nodes of this tree are relevant, we capture the state of the computation by a set of partial assignments $\Xi \subseteq \widetilde{\mathrm{V}}(\mathbf{X}_\Phi)$. Ξ always implies a partition of all assignments $\mathrm{V}(\mathbf{X}_\Phi)$. The function $\mathtt{refine}_{L,V} : \mathcal{P} \times 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ applies one refinement step to a set of leaves Ξ, using leaf selection heuristic $L : \mathcal{P} \times 2^{\mathcal{A}} \to \mathcal{A}$ and variable selection heuristic $V : \mathcal{P} \times \mathcal{A} \to \mathcal{X}$. Letting $\xi := L(\Phi, \Xi)$, and $X := V(\Phi, \xi)$,

$$\mathtt{refine}_{L,V}(\Phi, \Xi) := (\Xi \setminus \{\xi\}) \cup \{\{\xi \cup \{X \mapsto x_i\}\} \mid x_i \in \mathrm{Dom}(X)\}. \tag{9}$$

To obtain an estimate of the partition function, we sum over the estimates $Z_{\Phi[\xi]}^{BP}$ obtained from applying BP to the problem Φ conditioned on partial assignment ξ. We define the function $\mathtt{sum} : \mathcal{P} \times 2^{\mathcal{A}} \to \mathbb{R}^+$ as

$$\mathtt{sum}(\Phi, \Xi) := \sum_{\xi\in\Xi} Z_{\Phi[\xi]}^{\mathrm{BP}}. \tag{10}$$

Then $\mathtt{CBP}_{L,V} : \mathcal{P} \times \mathbb{N}^+ \to \mathbb{R}^+$ estimates the partition function using n steps of CBP by

$$\mathtt{CBP}_{L,V}(\Phi, n) := \mathtt{sum}(\Phi, \mathtt{refine}_{L,V}^n(\Phi, \{\emptyset\})). \tag{11}$$

Here, $\texttt{refine}^n_{L,V}$ means the n-fold recursive application of $\texttt{refine}_{L,V}$ in its second argument: $\texttt{refine}^n_{L,V}(\Phi, \Xi) \coloneqq \texttt{refine}_{L,V}(\Phi, \texttt{refine}^{n-1}_{L,V}(\Phi, \Xi))$ and $\texttt{refine}^0_{L,V}(\Phi, \Xi) \coloneqq \Xi$. Also note that this formalization of CBP is agnostic to the used inference algorithm, and every other way of calculating an approximate partition function can be used where $Z^{BP}_{\Phi[\xi]}$ appear.

Since BP yields exact results on tree-structured problems, one can stop the decomposition of a leaf once it contains no loops, or use other exact methods to solve the leaf earlier. But anyway CBP converges to the exact solution, since it becomes equivalent to summing over all assignments once all variables are conditioned in all leaves. Also note that the algorithm terminates after finitely many steps.

Proposition 1. *For all factor products Φ, all leaf selection heuristics L and variable selection heuristics V*

$$\lim_{n\to\infty} \mathit{CBP}_{L,V}(\Phi, n) = Z_\Phi.$$

In theory a run of BP only has to be performed on leaves once the final result is computed. But most of the proposed heuristics draw their information from a run of BP, making it necessary to run BP on every intermediate conditioned problem. Note that these problems also become smaller over time, and the computational effort required by running BP on each new leaf should decrease.

In its formulation given by us, CBP can be implemented as an anytime algorithm, because one can compute additional applications of `refine` until time runs out. This anytime behavior is in contrast to the original definition of CBP (Eaton and Ghahramani, 2009), which was recursive and required the provision of a stopping criterion such as the maximum recursion depth or some threshold value for the leaves' partition function estimates.

4 Heuristics for CBP

The variable selection heuristic V in $\texttt{refine}_{L,V}$ is given an assignment and the original problem (thus the equivalent to a conditioned problem), and picks a target variable for conditioning next. We go on to discuss the properties such a selection scheme should fulfill and follow this with proposing some simple heuristics that try to achieve these goals.

Viewing the problem of picking a variable to condition on from the perspective of the used approximative algorithm, we hope to obtain more accurate results from BP applied to the subproblems than from BP applied to the original problem. One of the main reasons BP performs unsatisfactorily are near deterministic factors, which can induce long distance dependences between variables. Short loops can also lead to oscillation and prevent BP from producing a good approximation (Koller and Friedman, 2009, pp. 428-429). We should thus seek to condition on variables that participate in such problematic structures.

From the perspective of using conditioning to decompose a problem we want to exercise the strengths of conditioning and choose variables for branching that allow this. Conditioning lets us exploit certain kinds of structure. First, we may encounter partial assignments that let us evaluate early some factors that might yield zero, and thus we can avoid examining all further extensions and prune our search early. Picking variables that yield some zero probability assignments to their value is thus favorable. This concept is very similar to unit propagation in the field of SAT solving.

A different structural property that is exploitable when conditioning is *context-specific independence*. Some variables X, Y may become independent of each other in Φ when conditioning on some context $\xi \in \mathrm{V}(\mathbf{C})$, i.e. they are situated in different components of $G_{\Phi[\xi]}$. This might not be true for a different context $\xi' \in \mathrm{V}(\mathbf{C})$ and thus is not revealed in the graphical structure of G_Φ. By preferring assignments that induce more independences, we can thus steer towards sub-problems of lower structural complexity. This can, for example, be measured by their respective tree width. We know that BP applied to tree-structured problems is exact, so we assume that sparser problems lead to better results when performing BP, although this might not be true in general. For future work, it might also be of interest to examine variable selection schemes employed in SAT and CSP solving.

After having discussed the aims we try to fulfill when selecting a branching variable, let us now look at some concrete heuristics.

Variable Selection Heuristics

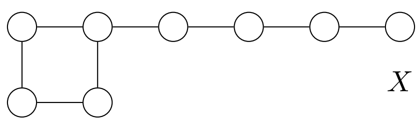
All variable selection heuristics that are described in the following section are given a problem as input (as opposed to both the original problem and a partial assignment). This argument problem is defined to be the conditioned problem $V(\Phi') := V(\Phi, \xi)$. We deal analogously with the later described leaf selection heuristic functions.

Time To Convergence. We hypothesize that those variables that take long to converge in a run of BP are also those variables located in areas where BP yields a bad approximation. This claim is also supported by the theoretical result of Weiss (1997) for networks with a single loop. For this purpose, let us define the function U_Φ that maps directed versions of edges in the factor graph of Φ to $\mathbb{N}^+$ in the following way. Assuming that the used BP schedule only recomputes a single message each step, $U_\Phi(X \to \phi)$ returns the number of the BP iteration in which $\delta_{X \to \phi}$ was changed last time. This requires that the BP implementation can detect the convergence of messages and does not further update them.

A first possible heuristic NTTC (Naive Time To Convergence) picks the variable that participated in the last message update before convergence:

$$\mathrm{NTTC}(\Phi) := \underset{X \in \mathbf{X}_\Phi}{\arg\max} \max_{\phi \in \mathrm{Nb}(X)} \max\left(U_\Phi(X \to \phi), U_\Phi(\phi \to X)\right). \tag{12}$$

But there are good reasons to assume that a variable selected by NTTC is not part of the problematic region. For example imagine a problem with one loop and a long tail, like the one depicted below.



The reason of BP converging slowly and yielding a wrong result will be messages oscillating in the loop. Once the loop has settled, the tail needs to adjust to the messages inside the loop. Picking a variable that received the last update, as NTTC does, will result branching on variable X. And that will yield no improvement to our approximation.

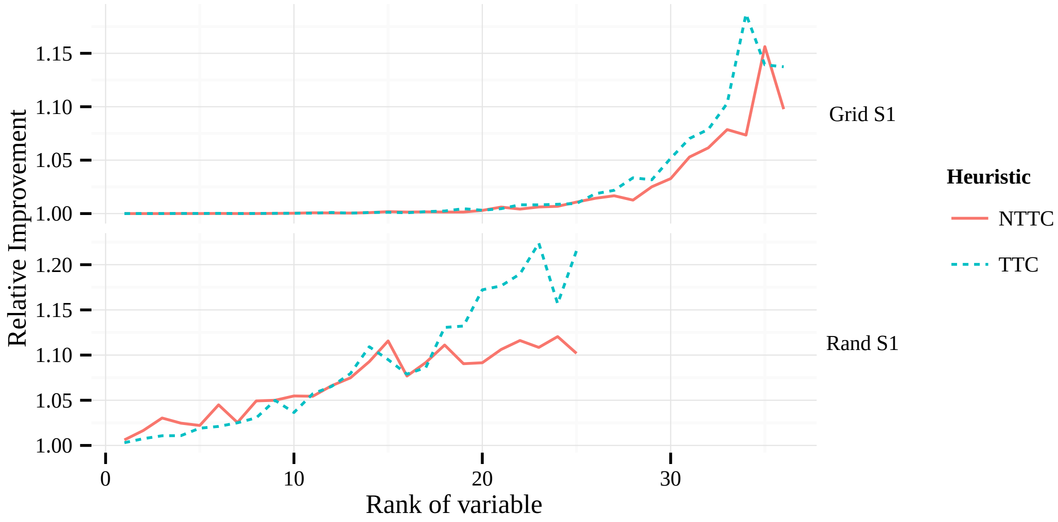


Figure 1: Median of the relative improvement $\Delta\delta_{*,V}(\Phi, 1)$ (see Equation 17) of one step of CBP over 1000 random problem instances. Problems are 6×6 grids and random problems with 25 variables. Variable selection is based on the ranking of variables according to the values inside the arg max of Equations 12 and 13. The result of conditioning on higher ranked variables is on the right. The plots show that conditioning on variables that score higher gives a larger improvement. Also TTC yields a better improvement than NTTC.

Therefore, we propose a second heuristic TTC. This heuristic selects a variable next to the edge that participated in the last *bidirectional* update. Formally, TTC is defined by the following equation:

$$\mathrm{TTC}(\Phi) := \underset{X \in \mathbf{X}_\Phi}{\arg\max} \max_{\phi \in \mathrm{Nb}(X)} \min\left(U_\Phi(X \to \phi), U_\Phi(\phi \to X)\right) \tag{13}$$

We empirically analyzed the suitability of NTTC and TTC by a small scale experiment (Figure 1). We can observe that our hypothesis is supported, since the relative improvement is larger for conditioning on the highly ranked variables. We can also observe that TTC appears to perform slightly better than NTTC, which also follows our intuition.

Min Entropy. The *Min Entropy* heuristic selects the variable with the lowest entropy:

$$V_{\min \mathbb{H}}(\Phi) := \underset{X \in \mathbf{D}_\Phi}{\arg\min} \, \mathbb{H}_{\beta_X} \tag{14}$$

This heuristic favors extreme or even deterministic variable beliefs and thus might lead to some children with a very low probability, which might be later exploited by focusing on the high probability children. In addition, such extreme beliefs might appear close to extreme factors, and by eliminating them we hope to obtain a smoother problem.

Max Degree. By choosing a variable that appears in many factors, the *Max Degree* heuristic tries to reduce dependencies and results in structurally simpler children:

$$V_{\max D}(\Phi) \coloneqq \arg\max_{X \in \mathbf{X}_\Phi} |\mathrm{Nb}(X)| \tag{15}$$

Tree Width. The *Tree Width* heuristic, just like the *Max Degree* heuristic tries to structurally simplify the problem. But it does so in a more elaborate way. It randomly selects a variable, which appears in the largest clique of a constructed junction tree, we obtain using the *min-degree* (also called *min-neighbors*) heuristic (Koller and Friedman, 2009, p. 314).

BBP. Variable selection is the main heuristic for the original CBP (Eaton and Ghahramani, 2009). Eaton's hypothesis about which variable should be conditioned on focuses on the idea to capture long ranging correlations. He tries to implement such a selection based on calculating the derivative of some value V with respect to the marginal beliefs $P_\Phi(X)$. He shows that this calculation can be performed efficiently using *back-belief-propagation* (BBP). BBP is then used to find variables that "push the model's beliefs in a certain direction". This direction is defined by a sample drawn by Gibbs sampling, which is hopefully a representative of one of the modes of Φ.

We want to make one general remark about the described heuristics here. In contrast to the original CBP algorithm, the algorithm as stated here always branches on all values of the selected variable. While this allows us to use variable selection heuristics that are oblivious to the variable's values, like the structural heuristics *Max Degree* and *Tree Width*, this approach is possibly inferior when applied to problems with large variable domains. While our empirical evaluation focuses on problems with binary variables only, we like to note that at least the *TTC* and *Min Entropy* heuristics can easily be adapted to be value-specific.

Leaf Selection Heuristics

The CBP algorithm, as described in this essay, selects one leaf to further condition in each iteration. In contrast to the original CBP algorithm, this allows for a further parametrization by choosing among different heuristics to pick the next leaf. If we were only performing one step of refinement, we would want to select a subproblem for further conditioning, for which one step of CBP yields the maximal reduction in approximation error. Obviously this reduction depends largely on the choice of variable selection heuristic and is thus difficult to analyze in isolation. In the following paragraphs we propose some basic heuristics which we expect to perform well.

Max Z. The first leaf selection heuristic we propose chooses the leaf $\xi \in \Xi$ with the highest $Z^{\mathrm{BP}}_{\Phi[\xi]}$. The idea is to focus on a leaf that has a high impact on the final result. In addition, the selection of the most probable leaf does well, since it also fights the accumulation of error caused by sub-problems for which BP overestimates the true partition function.

Time To Convergence. In a similar manner as the *Last Update* heuristic for variable selection, we select the leaf that took the longest for BP to converge on. This heuristic's intent is to identify problems that are likely to have inaccurate approximations.

Min Depth. This heuristic chooses a leaf that has a minimal number of variables conditioned. This approach mimics the original recursive CBP algorithm. It also has one desirable property: it guarantees that a leaf *will* be picked sooner or later. This can be beneficial because it allows CBP to fix grossly wrong approximations that might not be selected otherwise; e.g., when a leaf with a significant weight gets largely underestimated, then the *Max Z* heuristic will not touch it again and it remains as a source of error.

5 Evaluation

We evaluated the proposed heuristics on randomly generated problems with different topologies and different methods for generating potentials. We focus on the accuracy of inferring the partition function Z_Φ. To measure the total approximation error, we report the *relative error* of the inferred log partition function

$$\delta_{L,V}(\Phi, n) := \left| \frac{\log \mathtt{CBP}_{L,V}(\Phi, n) - \log Z_\Phi}{\log Z_\Phi} \right|. \tag{16}$$

The value $\delta_{*,*}(\Phi, 0)$ is the result of running ordinary BP on the original problem and can serve as a baseline. The *relative improvement* is the relative error of CBP compared to the relative error of BP:

$$\Delta\delta_{L,V}(\Phi, n) := \frac{\delta_{*,*}(\Phi, 0)}{\delta_{L,V}(\Phi, n)} \tag{17}$$

Note that the relative improvement is larger for better heuristics.

We generate problems using two different graph topologies, and binary random variables only. The topologies are two dimensional grids (Grid), and random graphs with 25 variables and 50 factors over three randomly selected variables each (Rand). We use two methods to generate values for factors. They are either sampled from an exponentiated normal distribution $\exp(\mathcal{N}(0, \sigma))$ with standard deviation σ (denoted by SX for $\sigma = X$). Or they are generated by starting with a neutral factor and changing just one value by sampling it from an exponentiated normal distribution with a given standard deviation (denoted by CX for $\sigma = X$). The CX potentials simulate structured factors (or features), like the ones obtained from grounding Markov Logic Networks (Richardson and Domingos, 2006). These factors are basically a soft clause and as such they exhibit context-specific independence, since such a factor reduces to a neutral factor as soon as a variable is assigned in contradiction to the special assignment.

To obtain an overview over the performance of CBP with various heuristics, we generated 500 instances from each problem class and applied 64 steps of CBP, implemented in our own framework. We used the available implementation of the BBP heuristic available in libDAI (Mooij, 2010). Since that implementation does not have a leaf selection heuristic, we assign the MIN DEPTH heuristic to it, which is equivalent when the number of steps is a power of two. We did not include the NTTC variable selection heuristic in the plots. Its performance was always slightly below the performance of TTC.

The relative errors for some leaf selection and variable selection heuristics are given in Figure 2. One notices that the approximation error of BBP from libDAI is lower for the first iteration on some problem classes (Rand S2, Rand C2). Our investigation revealed that our BP implementation and that from libDAI disagree on problems with larger errors,

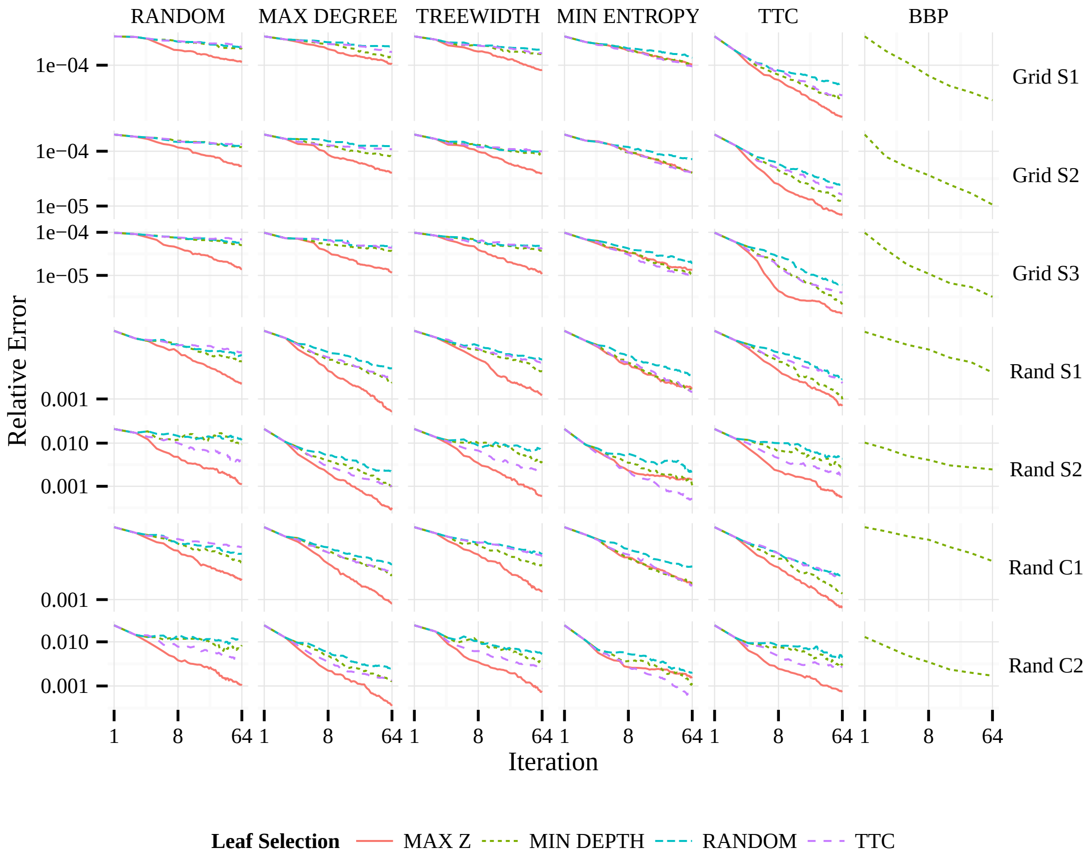


Figure 2: Median of the relative error $\delta_{L,V}(\Phi, n)$ over 500 problems plotted over the number of CBP steps n for different combinations of leaf selection heuristic L and variable selection heuristic V. Plot columns show different V; plot rows show problem classes; X-axis and Y-axis are logarithmic; lower values are better.

with libDAI yielding a better result more often than not. This is caused by failures to converge. The message schedule in libDAI always updates all messages in each step, while our implementation only updates if a significant change would occur. It appears that the more aggressive updating of libDAI improves BP convergence. Only the problem configurations Rand S2 and Rand C2 contain cases where BP did not converge.

Analyzing the results, we can notice that the median decrease in relative error appears linear on the log-log plots for all heuristic combinations[2]. This means that the benefit of CBP only increases logarithmically with the number of leaves in the tree. This is in accordance with a theoretical result about the related mixture of mean field approximation stated by Jaakkola and Jordan (1998), and follows the intuition that later on, the importance of a single leaf decreases, and a correction applied to it has a smaller influence on the final result.

[2]The mean shows the same relationship, but is less stable.

Figure 3: Median of the relative improvement $\Delta\delta_{L,V}(\Phi, 64)$ after 64 steps of CBP over 500 problems for different combinations of leaf selection heuristic L and variable selection heuristic V. Plot columns show different V; plot rows show problem classes; Y-axis is logarithmic; higher values are better.

For better comparison we also provide the relative improvement after 64 steps of CBP in Figure 3, which is basically equivalent to the slope of the curve in Figure 2. The improvement CBP yields over plain BP is very good for the examined problem classes, yielding a decrease in error of nearly two magnitudes after 64 steps for some configurations. Also all examined heuristics perform better than the random heuristics.

Concerning the influence of the used leaf selection heuristic, the MAX Z heuristic dominates all configurations. This was except for the MIN ENTROPY variable selection heuristic which yielded the best results when used with TTC for the Rand S2 and Rand C2 problems. The superior performance of the MAX Z leaf selection heuristic supports our intuition that focusing on the most important subproblems is a good strategy. We expect that a randomized mixture of MAX Z with MIN DEPTH may perform even better, because this mitigates neglecting underestimated subproblems.

When looking at the performance of the various variable selection heuristics, we see that

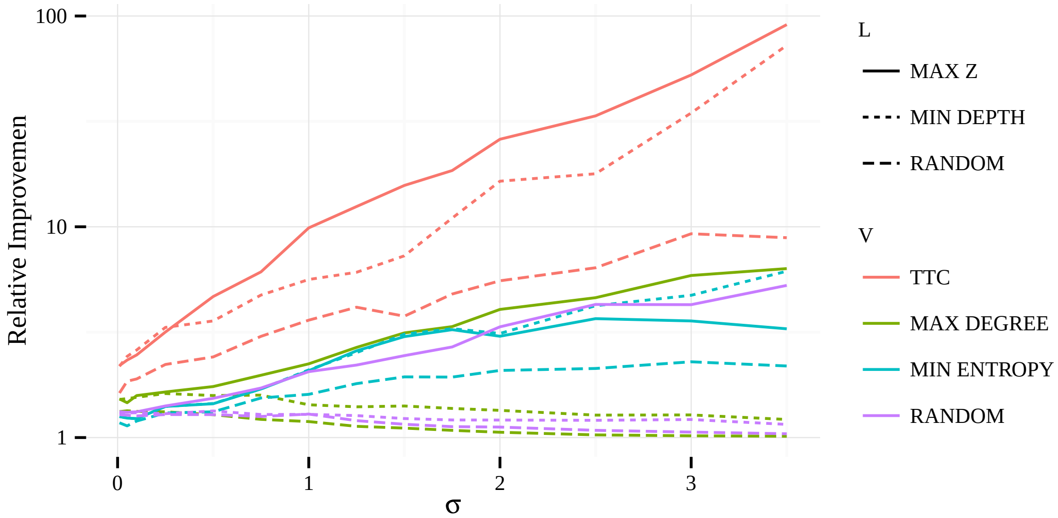


Figure 4: Median of the relative improvement $\Delta\delta_{L,V}(\Phi, 64)$ after 64 steps of CBP with $L = \text{MAX Z}, V = \text{TTC}$. Evaluation uses 250 instances of 8×8 grids with varying strength of factors each: standard deviation σ is plotted along x-axis. The plot shows how the benefit of CBP improves with tighter coupling.

TTC comes out as the best heuristic on the grid problems, basically tying with the much more complicated BBP when focusing only on the MIN DEPTH leaf selection. These two variable selection heuristics are the only ones that perform well on grid problems. On the randomly structured problems, it seems that all heuristics deliver at least a decent performance. This effect might also be attributed to the lower number of variables in these problems compared to the grid problems. For the randomly structured problems we observe that the structure-oriented heuristic MAX DEGREE performs best.

When looking at the strength of the factor values, we can also recognize that the improvement in accuracy offered by the CBP approach is better for the non-smooth potentials, despite BP seems to provide about the same initial approximation for all values of sigma. A possible explanation is that with increasing sigma, the probability mass is concentrated in fewer modes, and CBP manages to concentrate on those regions. We had a closer look at this phenomenon with a dedicated experiment (Figure 4), focusing only on grid problems and the best-performing heuristic for those (MAX Z/TTC). The results show that the improvement CBP delivers over ordinary BP increases very consistently with the strength of the dependencies between variables. For higher values of σ, this improvement cannot be attributed solely to the growing degradation of the BP approximation for low entropy distributions, as our experiments revealed that the relative error of BP maxes around $\sigma = 1$ (Figure 5), at least for the range of σ we examined.

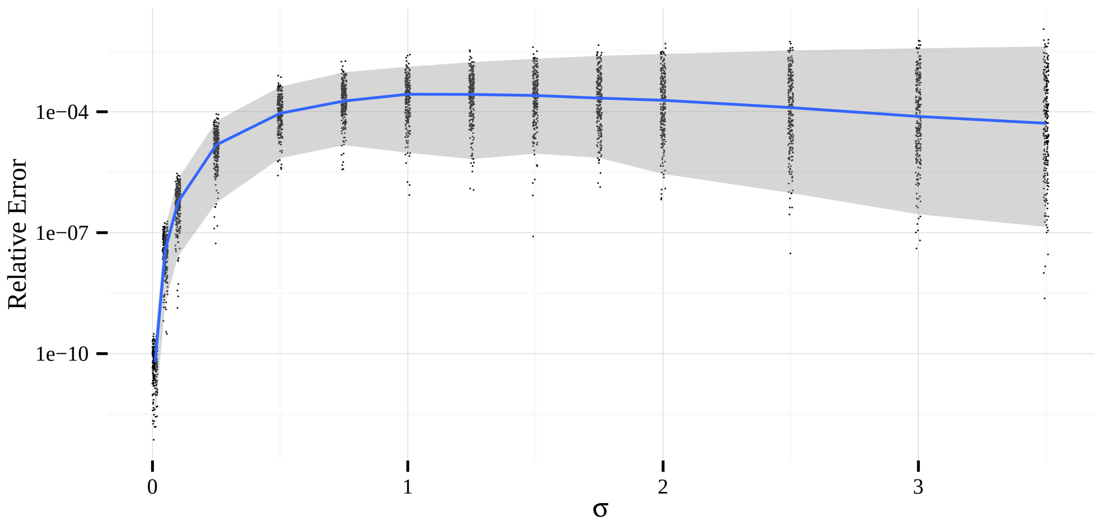


Figure 5: Median and $0.05, 0.95$ quantiles of relative error for ordinary BP on grid problems with varying amount of coupling. Evaluation uses 250 instances of 8×8 grids with varying strength of factors each: standard deviation σ is plotted along x-axis.

6 Discussion

CBP offers a simple means to improve the accuracy of BP. Our formulation can be cast as an anytime algorithm, and allows to trade in time and space for improved accuracy. Since CBP solves partially conditioned problems, it is also able to reveal and exploit context-specific independence. Further, it can exploit deterministic dependencies when those become inconsistent with the current condition. Then it is possible to evaluate the current leaf to zero. In this way CBP is an algorithm that has facilities to solve both high entropy parts of problems (BP), as well as low entropy parts (conditioning). This is a perfect combination, as BP is weak on low entropy problems (i.e. problems with very strong dependencies), and conditioning fails under the presence of many equal choices.

Despite the apparent benefits of CBP, we would also like to point out a major shortcoming that has to be solved before CBP can be used as a true general-purpose inference algorithm. As stated before, the accuracy of CBP improves only with the logarithm of the number of steps. This is intuitive, since with the progression of CBP the error contribution of each leaf decreases with its weight, and thus each further decomposition step has a lesser impact on the final result. In addition, the relative improvement per step will be much smaller for problems with more variables, as the absolute improvement that can be gained by conditioning on one variable stays the same. This means that the computational cost of CBP required to achieve the same relative improvement grows exponentially with the problem size, and this is clearly impractical. To prove this, we conducted an experiment that shows the relative improvement of a fixed number of CBP steps over an increasing problem size in (Figure 6). To make CBP a viable choice, we have to develop a way to exploit the independence between the conditioning effects of variables that are largely unrelated to each other.

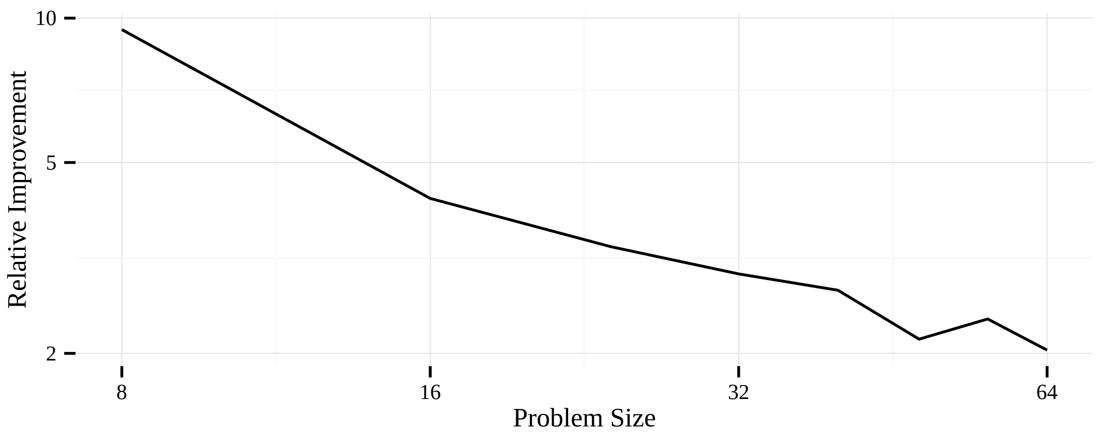


Figure 6: Median of relative improvement $\Delta\delta_{L,V}(\Phi, 64)$ of 64 steps of CBP on 500 instances of $8 \times (8 \cdot \text{size})$ grid problems ($\sigma = 1$); $L = \text{MAX Z}, V = \text{TTC}$. One can observe how the benefit of a fixed number of CBP steps diminishes with increasing problem size.

This work focuses on finding good heuristics for improving the BP approximation on the conditioned problems. There remain many opportunities to improve CBP on the decomposition side by using concepts from the CSP community, just as SampleSearch (Gogate and Dechter, 2011) does. Unit-Propagation and clause learning are two prominent candidates that could greatly improve the performance on problems containing deterministic constraints.

Any serious implementation of CPB should also examine leaves for the possibility of solving them exactly. This could mean applying a Junction Tree algorithm (Koller and Friedman, 2009) as soon as the tree width drops below some threshold value. Heuristic tests for tree width can be very cheap. It is also possible to update an existing tree-decomposition on each conditioning step, which can practically eliminate the cost of this test.

What appears as another possible improvement deals with the possibility of subproblems decomposing into independent parts after conditioning on some values; a concept also known as Cutset-Conditioning (Pearl, 1988, pp. 204-210). It appears tempting to decompose subproblems multiplicatively and solve these independently of each other, but we have to keep in mind that BP already exploits factorization which manifests in the graphical structure of the problem.

7 Conclusion

We have proposed a reformulated, iterative version of CBP that allows CBP to be used as an anytime algorithm and allows better tuning via the use of a leaf selection heuristic. We discussed the fundamental goals that both kinds of heuristics try to achieve, and proposed a set of interesting candidates. In an empirical evaluation we could demonstrate that the revised CBP algorithm using the proposed heuristics outperforms the original heuristic in terms of accuracy. The new heuristics are both simpler to implement, computationally less demanding, and yield more exact results.

Overall CBP can serve as a simple method to improve the accuracy of Belief Propagation and extends readily to other message passing algorithms, such as Generalized Belief Propagation (Yedidia et al., 2001b). Since the improvement offered by CBP grows only logarithmically with the number of leaf problems, its use remains limited. In this regard, a method that lifts this limitation by reusing computations across leaf problems is conceivable. In any case, CBP is not only another probabilistic inference method, but can also serve as a tool to gain insights into the behavior of BP.

Acknowledgements

This work is done within the Transregional Collaborative Research Centre SFB/ TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

References

Bidyuk, B. and Dechter, R. (2003). Cycle-cutset sampling for Bayesian networks. In *Advances in Artificial Intelligence*, pages 297–312. Springer.

Bouchard, G. and Zoeter, O. (2009). Split variational inference. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 57–64. ACM.

Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in bayesian networks. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 115–123. Morgan Kaufmann Publishers Inc.

Eaton, F. and Ghahramani, Z. (2009). Choosing a variable to clamp: Approximate inference using conditioned belief propagation. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 145–152.

Geier, T., Richter, F., and Biundo, S. (2014). Conditioned Beleif Propagation Revisited. In *European Conference on Artificial Intelligence*.

Gogate, V. and Dechter, R. (2011). Samplesearch: Importance sampling in presence of determinism. *Artificial Intelligence*, 175(2):694–729.

Jaakkola, T. S. and Jordan, M. I. (1998). Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*, pages 163–173. Springer.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519.

Mooij, J. M. (2010). libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3):241–288.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine learning*, 62(1-2):107–136.

Weiss, Y. (1997). Belief propagation and revision in networks with loops. AI Memo 1616 (CBCL Paper 155), MIT. Presented in NIPS 97 workshop on graphical models.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001a). Bethe free energy, kikuchi approximations, and belief propagation algorithms. *Advances in neural information processing systems*, 13.

Yedidia, J. S., Freeman, W. T., Weiss, Y., et al. (2001b). Generalized belief propagation. *Advances in neural information processing systems*, pages 689–695.

Zhang, N. L. and Poole, D. (1999). On the role of context-specific independence in probabilistic inference.

Liste der bisher erschienenen Ulmer Informatik-Berichte
Einige davon sind per FTP von `ftp.informatik.uni-ulm.de` erhältlich
Die mit * markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm
Some of them are available by FTP from `ftp.informatik.uni-ulm.de`
Reports marked with * are out of print

91-01 *Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe*
Instance Complexity

*91-02** *K. Gladitz, H. Fassbender, H. Vogler*
Compiler-Based Implementation of Syntax-Directed Functional Programming

*91-03** *Alfons Geser*
Relative Termination

*91-04** *J. Köbler, U. Schöning, J. Toran*
Graph Isomorphism is low for PP

91-05 *Johannes Köbler, Thomas Thierauf*
Complexity Restricted Advice Functions

*91-06** *Uwe Schöning*
Recent Highlights in Structural Complexity Theory

*91-07** *F. Green, J. Köbler, J. Toran*
The Power of Middle Bit

*91-08** *V.Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf*
Reductions for Sets of Low Information Content

*92-01** *Vikraman Arvind, Johannes Köbler, Martin Mundhenk*
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

*92-02** *Thomas Noll, Heiko Vogler*
Top-down Parsing with Simulataneous Evaluation of Noncircular Attribute Grammars

92-03 *Fakultät für Informatik*
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

*92-04** *V. Arvind, J. Köbler, M. Mundhenk*
Lowness and the Complexity of Sparse and Tally Descriptions

*92-05** *Johannes Köbler*
Locating P/poly Optimally in the Extended Low Hierarchy

*92-06** *Armin Kühnemann, Heiko Vogler*
Synthesized and inherited functions -a new computational model for syntax-directed semantics

*92-07** *Heinz Fassbender, Heiko Vogler*
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

*92-08** *Uwe Schöning*
On Random Reductions from Sparse Sets to Tally Sets

*92-09** *Hermann von Hasseln, Laura Martignon*
Consistency in Stochastic Network

92-10 *Michael Schmitt*
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function

92-11 *Johannes Köbler, Seinosuke Toda*
On the Power of Generalized MOD-Classes

92-12 *V. Arvind, J. Köbler, M. Mundhenk*
Reliable Reductions, High Sets and Low Sets

92-13 *Alfons Geser*
On a monotonic semantic path ordering

*92-14** *Joost Engelfriet, Heiko Vogler*
The Translation Power of Top-Down Tree-To-Graph Transducers

93-01 *Alfred Lupper, Konrad Froitzheim*
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager

93-02 *M.H. Scholl, C. Laasch, C. Rich, H.-J. Schek, M. Tresch*
The COCOON Object Model

93-03 *Thomas Thierauf, Seinosuke Toda, Osamu Watanabe*
On Sets Bounded Truth-Table Reducible to P-selective Sets

93-04 *Jin-Yi Cai, Frederic Green, Thomas Thierauf*
On the Correlation of Symmetric Functions

93-05 *K.Kuhn, M.Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam*
A Conceptual Approach to an Open Hospital Information System

93-06 *Klaus Gaßner*
Rechnerunterstützung für die konzeptuelle Modellierung

93-07 *Ullrich Keßler, Peter Dadam*
Towards Customizable, Flexible Storage Structures for Complex Objects

94-01 *Michael Schmitt*
On the Complexity of Consistency Problems for Neurons with Binary Weights

94-02 *Armin Kühnemann, Heiko Vogler*
A Pumping Lemma for Output Languages of Attributed Tree Transducers

94-03 *Harry Buhrman, Jim Kadin, Thomas Thierauf*
On Functions Computable with Nonadaptive Queries to NP

94-04 *Heinz Faßbender, Heiko Vogler, Andrea Wedel*
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers

94-05 *V. Arvind, J. Köbler, R. Schuler*
On Helping and Interactive Proof Systems

94-06 *Christian Kalus, Peter Dadam*
Incorporating record subtyping into a relational data model

94-07 *Markus Tresch, Marc H. Scholl*
A Classification of Multi-Database Languages

94-08 *Friedrich von Henke, Harald Rueß*
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge

94-09 *F.W. von Henke, A. Dold, H. Rueß, D. Schwier, M. Strecker*
Construction and Deduction Methods for the Formal Development of Software

94-10 *Axel Dold*
Formalisierung schematischer Algorithmen

94-11 *Johannes Köbler, Osamu Watanabe*
New Collapse Consequences of NP Having Small Circuits

94-12 *Rainer Schuler*
On Average Polynomial Time

94-13 *Rainer Schuler, Osamu Watanabe*
Towards Average-Case Complexity Analysis of NP Optimization Problems

94-14 *Wolfram Schulte, Ton Vullinghs*
Linking Reactive Software to the X-Window System

94-15 *Alfred Lupper*
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen

94-16 *Robert Regn*
Verteilte Unix-Betriebssysteme

94-17 *Helmuth Partsch*
Again on Recognition and Parsing of Context-Free Grammars:
Two Exercises in Transformational Programming

94-18 *Helmuth Partsch*
Transformational Development of Data-Parallel Algorithms: an Example

95-01 *Oleg Verbitsky*
On the Largest Common Subgraph Problem

95-02 *Uwe Schöning*
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension

95-03 *Harry Buhrman,Thomas Thierauf*
The Complexity of Generating and Checking Proofs of Membership

95-04 *Rainer Schuler, Tomoyuki Yamakami*
Structural Average Case Complexity

95-05 *Klaus Achatz, Wolfram Schulte*
Architecture Indepentent Massive Parallelization of Divide-And-Conquer Algorithms

95-06 *Christoph Karg, Rainer Schuler*
Structure in Average Case Complexity

95-07 *P. Dadam, K. Kuhn, M. Reichert, T. Beuter, M. Nathe*
ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen

95-08 *Jürgen Kehrer, Peter Schulthess*
Aufbereitung von gescannten Röntgenbildern zur filmlosen Diagnostik

95-09 *Hans-Jörg Burtschick, Wolfgang Lindner*
On Sets Turing Reducible to P-Selective Sets

95-10 *Boris Hartmann*
Berücksichtigung lokaler Randbedingung bei globaler Zieloptimierung mit neuronalen Netzen am Beispiel Truck Backer-Upper

95-11 *Thomas Beuter, Peter Dadam*:
Prinzipien der Replikationskontrolle in verteilten Systemen

95-12 *Klaus Achatz, Wolfram Schulte*
Massive Parallelization of Divide-and-Conquer Algorithms over Powerlists

95-13 *Andrea Mößle, Heiko Vogler*
Efficient Call-by-value Evaluation Strategy of Primitive Recursive Program Schemes

95-14 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
A Generic Specification for Verifying Peephole Optimizations

96-01 *Ercüment Canver, Jan-Tecker Gayen, Adam Moik*
Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE

96-02 *Bernhard Nebel*
Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class

96-03 *Ton Vullinghs, Wolfram Schulte, Thilo Schwinn*
An Introduction to TkGofer

96-04 *Thomas Beuter, Peter Dadam*
Anwendungsspezifische Anforderungen an Workflow-Mangement-Systeme am Beispiel der Domäne Concurrent-Engineering

96-05 *Gerhard Schellhorn, Wolfgang Ahrendt*
Verification of a Prolog Compiler - First Steps with KIV

96-06 *Manindra Agrawal, Thomas Thierauf*
Satisfiability Problems

96-07 *Vikraman Arvind, Jacobo Torán*
A nonadaptive NC Checker for Permutation Group Intersection

96-08 *David Cyrluk, Oliver Möller, Harald Rueß*
An Efficient Decision Procedure for a Theory of Fix-Sized Bitvectors with Composition and Extraction

96-09 *Bernd Biechele, Dietmar Ernst, Frank Houdek, Joachim Schmid, Wolfram Schulte*
Erfahrungen bei der Modellierung eingebetteter Systeme mit verschiedenen SA/RT–Ansätzen

96-10 *Falk Bartels, Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Formalizing Fixed-Point Theory in PVS

96-11 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Mechanized Semantics of Simple Imperative Programming Constructs

96-12 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Generic Compilation Schemes for Simple Programming Constructs

96-13 *Klaus Achatz, Helmuth Partsch*
From Descriptive Specifications to Operational ones: A Powerful Transformation Rule, its Applications and Variants

97-01 *Jochen Messner*
Pattern Matching in Trace Monoids

97-02 *Wolfgang Lindner, Rainer Schuler*
A Small Span Theorem within P

97-03 *Thomas Bauer, Peter Dadam*
A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration

97-04 *Christian Heinlein, Peter Dadam*
Interaction Expressions - A Powerful Formalism for Describing Inter-Workflow Dependencies

97-05 *Vikraman Arvind, Johannes Köbler*
On Pseudorandomness and Resource-Bounded Measure

97-06 *Gerhard Partsch*
Punkt-zu-Punkt- und Mehrpunkt-basierende LAN-Integrationsstrategien für den digitalen Mobilfunkstandard DECT

97-07 *Manfred Reichert, Peter Dadam*
$ADEPT_{flex}$ - Supporting Dynamic Changes of Workflows Without Loosing Control

97-08 *Hans Braxmeier, Dietmar Ernst, Andrea Mößle, Heiko Vogler*
The Project NoName - A functional programming language with its development environment

97-09 *Christian Heinlein*
Grundlagen von Interaktionsausdrücken

97-10 *Christian Heinlein*
Graphische Repräsentation von Interaktionsausdrücken

97-11 *Christian Heinlein*
Sprachtheoretische Semantik von Interaktionsausdrücken

97-12 *Gerhard Schellhorn, Wolfgang Reif*
Proving Properties of Finite Enumerations: A Problem Set for Automated Theorem Provers

97-13 *Dietmar Ernst, Frank Houdek, Wolfram Schulte, Thilo Schwinn*
Experimenteller Vergleich statischer und dynamischer Softwareprüfung für eingebettete Systeme

97-14 *Wolfgang Reif, Gerhard Schellhorn*
Theorem Proving in Large Theories

97-15 *Thomas Wennekers*
Asymptotik rekurrenter neuronaler Netze mit zufälligen Kopplungen

97-16 *Peter Dadam, Klaus Kuhn, Manfred Reichert*
Clinical Workflows - The Killer Application for Process-oriented Information Systems?

97-17 *Mohammad Ali Livani, Jörg Kaiser*
EDF Consensus on CAN Bus Access in Dynamic Real-Time Applications

97-18 *Johannes Köbler,Rainer Schuler*
Using Efficient Average-Case Algorithms to Collapse Worst-Case Complexity Classes

98-01 *Daniela Damm, Lutz Claes, Friedrich W. von Henke, Alexander Seitz, Adelinde Uhrmacher, Steffen Wolf*
Ein fallbasiertes System für die Interpretation von Literatur zur Knochenheilung

98-02 *Thomas Bauer, Peter Dadam*
Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse

98-03 *Marko Luther, Martin Strecker*
A guided tour through *Typelab*

98-04 *Heiko Neumann, Luiz Pessoa*
Visual Filling-in and Surface Property Reconstruction

98-05 *Ercüment Canver*
Formal Verification of a Coordinated Atomic Action Based Design

98-06 *Andreas Küchler*
On the Correspondence between Neural Folding Architectures and Tree Automata

98-07 *Heiko Neumann, Thorsten Hansen, Luiz Pessoa*
Interaction of ON and OFF Pathways for Visual Contrast Measurement

98-08 *Thomas Wennekers*
Synfire Graphs: From Spike Patterns to Automata of Spiking Neurons

98-09 *Thomas Bauer, Peter Dadam*
Variable Migration von Workflows in *ADEPT*

98-10 *Heiko Neumann, Wolfgang Sepp*
Recurrent V1 – V2 Interaction in Early Visual Boundary Processing

98-11 *Frank Houdek, Dietmar Ernst, Thilo Schwinn*
Prüfen von C–Code und Statmate/Matlab–Spezifikationen: Ein Experiment

98-12 *Gerhard Schellhorn*
Proving Properties of Directed Graphs: A Problem Set for Automated Theorem Provers

98-13 *Gerhard Schellhorn, Wolfgang Reif*
Theorems from Compiler Verification: A Problem Set for Automated Theorem Provers

98-14 *Mohammad Ali Livani*
SHARE: A Transparent Mechanism for Reliable Broadcast Delivery in CAN

98-15 *Mohammad Ali Livani, Jörg Kaiser*
Predictable Atomic Multicast in the Controller Area Network (CAN)

99-01 *Susanne Boll, Wolfgang Klas, Utz Westermann*
A Comparison of Multimedia Document Models Concerning Advanced Requirements

99-02 *Thomas Bauer, Peter Dadam*
Verteilungsmodelle für Workflow-Management-Systeme - Klassifikation und Simulation

99-03 *Uwe Schöning*
On the Complexity of Constraint Satisfaction

99-04 *Ercument Canver*
Model-Checking zur Analyse von Message Sequence Charts über Statecharts

99-05 *Johannes Köbler, Wolfgang Lindner, Rainer Schuler*
Derandomizing RP if Boolean Circuits are not Learnable

99-06 *Utz Westermann, Wolfgang Klas*
Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets

99-07 *Peter Dadam, Manfred Reichert*
Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications. Paderborn, Germany, October 6, 1999, GI–Workshop Proceedings, Informatik ’99

99-08 *Vikraman Arvind, Johannes Köbler*
Graph Isomorphism is Low for ZPP^{NP} and other Lowness results

99-09 *Thomas Bauer, Peter Dadam*
Efficient Distributed Workflow Management Based on Variable Server Assignments

2000-02 *Thomas Bauer, Peter Dadam*
Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT

2000-03 *Gregory Baratoff, Christian Toepfer, Heiko Neumann*
Combined space-variant maps for optical flow based navigation

2000-04 *Wolfgang Gehring*
Ein Rahmenwerk zur Einführung von Leistungspunktsystemen

2000-05 *Susanne Boll, Christian Heinlein, Wolfgang Klas, Jochen Wandel*
Intelligent Prefetching and Buffering for Interactive Streaming of MPEG Videos

2000-06 *Wolfgang Reif, Gerhard Schellhorn, Andreas Thums*
Fehlersuche in Formalen Spezifikationen

2000-07 *Gerhard Schellhorn, Wolfgang Reif (eds.)*
FM-Tools 2000: The 4th Workshop on Tools for System Design and Verification

2000-08 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Effiziente Durchführung von Prozessmigrationen in verteilten Workflow-Management-Systemen

2000-09 *Thomas Bauer, Peter Dadam*
Vermeidung von Überlastsituationen durch Replikation von Workflow-Servern in ADEPT

2000-10 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Adaptives und verteiltes Workflow-Management

2000-11 *Christian Heinlein*
Workflow and Process Synchronization with Interaction Expressions and Graphs

2001-01 *Hubert Hug, Rainer Schuler*
DNA-based parallel computation of simple arithmetic

2001-02 *Friedhelm Schwenker, Hans A. Kestler, Günther Palm*
3-D Visual Object Classification with Hierarchical Radial Basis Function Networks

2001-03 *Hans A. Kestler, Friedhelm Schwenker, Günther Palm*
RBF network classification of ECGs as a potential marker for sudden cardiac death

2001-04 *Christian Dietrich, Friedhelm Schwenker, Klaus Riede, Günther Palm*
Classification of Bioacoustic Time Series Utilizing Pulse Detection, Time and Frequency Features and Data Fusion

2002-01 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata

2002-02 *Walter Guttmann*
Deriving an Applicative Heapsort Algorithm

2002-03 *Axel Dold, Friedrich W. von Henke, Vincent Vialard, Wolfgang Goerigk*
A Mechanically Verified Compiling Specification for a Realistic Compiler

2003-01 *Manfred Reichert, Stefanie Rinderle, Peter Dadam*
A Formal Framework for Workflow Type and Instance Changes Under Correctness Checks

2003-02 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Supporting Workflow Schema Evolution By Efficient Compliance Checks

2003-03 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values

2003-04 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
On Dealing With Semantically Conflicting Business Process Changes.

2003-05 *Christian Heinlein*
Dynamic Class Methods in Java

2003-06 *Christian Heinlein*
Vertical, Horizontal, and Behavioural Extensibility of Software Systems

2003-07 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values (Corrected Version)

2003-08 *Changling Liu, Jörg Kaiser*
Survey of Mobile Ad Hoc Network Routing Protocols)

2004-01 *Thom Frühwirth, Marc Meister (eds.)*
First Workshop on Constraint Handling Rules

2004-02 *Christian Heinlein*
Concept and Implementation of C+++, an Extension of C++ to Support User-Defined Operator Symbols and Control Structures

2004-03 *Susanne Biundo, Thom Frühwirth, Günther Palm(eds.)*
Poster Proceedings of the 27th Annual German Conference on Artificial Intelligence

2005-01 *Armin Wolf, Thom Frühwirth, Marc Meister (eds.)*
19th Workshop on (Constraint) Logic Programming

2005-02 *Wolfgang Lindner (Hg.), Universität Ulm , Christopher Wolf (Hg.) KU Leuven*
2. Krypto-Tag – Workshop über Kryptographie, Universität Ulm

2005-03 *Walter Guttmann, Markus Maucher*
Constrained Ordering

2006-01 *Stefan Sarstedt*
Model-Driven Development with ACTIVECHARTS, Tutorial

2006-02 *Alexander Raschke, Ramin Tavakoli Kolagari*
Ein experimenteller Vergleich zwischen einer plan-getriebenen und einer leichtgewichtigen Entwicklungsmethode zur Spezifikation von eingebetteten Systemen

2006-03 *Jens Kohlmeyer, Alexander Raschke, Ramin Tavakoli Kolagari*
Eine qualitative Untersuchung zur Produktlinien-Integration über Organisationsgrenzen hinweg

2006-04 *Thorsten Liebig*
Reasoning with OWL - System Support and Insights –

2008-01 *H.A. Kestler, J. Messner, A. Müller, R. Schuler*
On the complexity of intersecting multiple circles for graphical display

2008-02 *Manfred Reichert, Peter Dadam, Martin Jurisch,l Ulrich Kreher, Kevin Göser, Markus Lauer*
Architectural Design of Flexible Process Management Technology

2008-03 *Frank Raiser*
Semi-Automatic Generation of CHR Solvers from Global Constraint Automata

2008-04 *Ramin Tavakoli Kolagari, Alexander Raschke, Matthias Schneiderhan, Ian Alexander*
Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für produktlinien-basierte Entwicklungsprozesse

2008-05 *Markus Kalb, Claudia Dittrich, Peter Dadam*
Support of Relationships Among Moving Objects on Networks

2008-06 *Matthias Frank, Frank Kargl, Burkhard Stiller (Hg.)*
WMAN 2008 – KuVS Fachgespräch über Mobile Ad-hoc Netzwerke

2008-07 *M. Maucher, U. Schöning, H.A. Kestler*
An empirical assessment of local and population based search methods with different degrees of pseudorandomness

2008-08 *Henning Wunderlich*
Covers have structure

2008-09 *Karl-Heinz Niggl, Henning Wunderlich*
Implicit characterization of FPTIME and NC revisited

2008-10 *Henning Wunderlich*
On span-P^{cc} and related classes in structural communication complexity

2008-11 *M. Maucher, U. Schöning, H.A. Kestler*
On the different notions of pseudorandomness

2008-12 *Henning Wunderlich*
On Toda's Theorem in structural communication complexity

2008-13 *Manfred Reichert, Peter Dadam*
Realizing Adaptive Process-aware Information Systems with ADEPT2

2009-01 *Peter Dadam, Manfred Reichert*
The ADEPT Project: A Decade of Research and Development for Robust and Fexible Process Support
Challenges and Achievements

2009-02 *Peter Dadam, Manfred Reichert, Stefanie Rinderle-Ma, Kevin Göser, Ulrich Kreher, Martin Jurisch*
Von ADEPT zur AristaFlow® BPM Suite – Eine Vision wird Realität "Correctness by Construction" und flexible, robuste Ausführung von Unternehmensprozessen

2009-03	*Alena Hallerbach, Thomas Bauer, Manfred Reichert* Correct Configuration of Process Variants in Provop
2009-04	*Martin Bader* On Reversal and Transposition Medians
2009-05	*Barbara Weber, Andreas Lanz, Manfred Reichert* Time Patterns for Process-aware Information Systems: A Pattern-based Analysis
2009-06	*Stefanie Rinderle-Ma, Manfred Reichert* Adjustment Strategies for Non-Compliant Process Instances
2009-07	*H.A. Kestler, B. Lausen, H. Binder H.-P. Klenk. F. Leisch, M. Schmid* Statistical Computing 2009 – Abstracts der 41. Arbeitstagung
2009-08	*Ulrich Kreher, Manfred Reichert, Stefanie Rinderle-Ma, Peter Dadam* Effiziente Repräsentation von Vorlagen- und Instanzdaten in Prozess-Management-Systemen
2009-09	*Dammertz, Holger, Alexander Keller, Hendrik P.A. Lensch* Progressive Point-Light-Based Global Illumination
2009-10	*Dao Zhou, Christoph Müssel, Ludwig Lausser, Martin Hopfensitz, Michael Kühl, Hans A. Kestler* Boolean networks for modeling and analysis of gene regulation
2009-11	*J. Hanika, H.P.A. Lensch, A. Keller* Two-Level Ray Tracing with Recordering for Highly Complex Scenes
2009-12	*Stephan Buchwald, Thomas Bauer, Manfred Reichert* Durchgängige Modellierung von Geschäftsprozessen durch Einführung eines Abbildungsmodells: Ansätze, Konzepte, Notationen
2010-01	*Hariolf Betz, Frank Raiser, Thom Frühwirth* A Complete and Terminating Execution Model for Constraint Handling Rules
2010-02	*Ulrich Kreher, Manfred Reichert* Speichereffiziente Repräsentation instanzspezifischer Änderungen in Prozess-Management-Systemen
2010-03	*Patrick Frey* Case Study: Engine Control Application
2010-04	*Matthias Lohrmann und Manfred Reichert* Basic Considerations on Business Process Quality
2010-05	*HA Kestler, H Binder, B Lausen, H-P Klenk, M Schmid, F Leisch (eds):* Statistical Computing 2010 - Abstracts der 42. Arbeitstagung
2010-06	*Vera Künzle, Barbara Weber, Manfred Reichert* Object-aware Business Processes: Properties, Requirements, Existing Approaches

2011-01 *Stephan Buchwald, Thomas Bauer, Manfred Reichert*
Flexibilisierung Service-orientierter Architekturen

2011-02 *Johannes Hanika, Holger Dammertz, Hendrik Lensch*
Edge-Optimized À-Trous Wavelets for Local Contrast Enhancement with Robust Denoising

2011-03 *Stefanie Kaiser, Manfred Reichert*
Datenflussvarianten in Prozessmodellen: Szenarien, Herausforderungen, Ansätze

2011-04 *Hans A. Kestler, Harald Binder, Matthias Schmid, Friedrich Leisch, Johann M. Kraus (eds):*
Statistical Computing 2011 - Abstracts der 43. Arbeitstagung

2011-05 *Vera Künzle, Manfred Reichert*
PHILharmonicFlows: Research and Design Methodology

2011-06 *David Knuplesch, Manfred Reichert*
Ensuring Business Process Compliance Along the Process Life Cycle

2011-07 *Marcel Dausend*
Towards a UML Profile on Formal Semantics for Modeling Multimodal Interactive Systems

2011-08 *Dominik Gessenharter*
Model-Driven Software Development with ACTIVECHARTS - A Case Study

2012-01 *Andreas Steigmiller, Thorsten Liebig, Birte Glimm*
Extended Caching, Backjumping and Merging for Expressive Description Logics

2012-02 *Hans A. Kestler, Harald Binder, Matthias Schmid, Johann M. Kraus (eds):*
Statistical Computing 2012 - Abstracts der 44. Arbeitstagung

2012-03 *Felix Schüssel, Frank Honold, Michael Weber*
Influencing Factors on Multimodal Interaction at Selection Tasks

2012-04 *Jens Kolb, Paul Hübner, Manfred Reichert*
Model-Driven User Interface Generation and Adaption in Process-Aware Information Systems

2012-05 *Matthias Lohrmann, Manfred Reichert*
Formalizing Concepts for Efficacy-aware Business Process Modeling

2012-06 *David Knuplesch, Rüdiger Pryss, Manfred Reichert*
A Formal Framework for Data-Aware Process Interaction Models

2012-07 *Clara Ayora, Victoria Torres, Barbara Weber, Manfred Reichert, Vicente Pelechano*
Dealing with Variability in Process-Aware Information Systems: Language Requirements, Features, and Existing Proposals

2013-01 *Frank Kargl*
Abstract Proceedings of the 7th Workshop on Wireless and Mobile Ad-Hoc Networks (WMAN 2013)

2013-02	*Andreas Lanz, Manfred Reichert, Barbara Weber* A Formal Semantics of Time Patterns for Process-aware Information Systems
2013-03	*Matthias Lohrmann, Manfred Reichert* Demonstrating the Effectiveness of Process Improvement Patterns with Mining Results
2013-04	*Semra Catalkaya, David Knuplesch, Manfred Reichert* Bringing More Semantics to XOR-Split Gateways in Business Process Models Based on Decision Rules
2013-05	*David Knuplesch, Manfred Reichert, Linh Thao Ly, Akhil Kumar, Stefanie Rinderle-Ma* On the Formal Semantics of the Extended Compliance Rule Graph
2013-06	*Andreas Steigmiller, Birte Glimm* Nominal Schema Absorption
2013-07	*Hans A. Kestler, Matthias Schmid, Florian Schmid, Dr. Markus Maucher, Johann M. Kraus (eds)* Statistical Computing 2013 - Abstracts der 45. Arbeitstagung
2013-08	*Daniel Ott, Dr. Alexander Raschke* Evaluating Benefits of Requirement Categorization in Natural Language Specifications for Review Improvements
2013-09	*Philip Geiger, Rüdiger Pryss, Marc Schickler, Manfred Reichert* Engineering an Advanced Location-Based Augmented Reality Engine for Smart Mobile Devices
2014-01	*Andreas Lanz, Manfred Reichert* Analyzing the Impact of Process Change Operations on Time-Aware Processes
2014-02	*Andreas Steigmiller, Birte Glimm, and Thorsten Liebig* Coupling Tableau Algorithms for the DL SROIQ with Completion-based Saturation Procedures
2014-03	*Thomas Geier, Felix Richter, Susanne Biundo* Conditioned Belief Propagation Revisited: Extended Version

Ulmer Informatik-Berichte
ISSN 0939-5091

Herausgeber:
Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
89069 Ulm