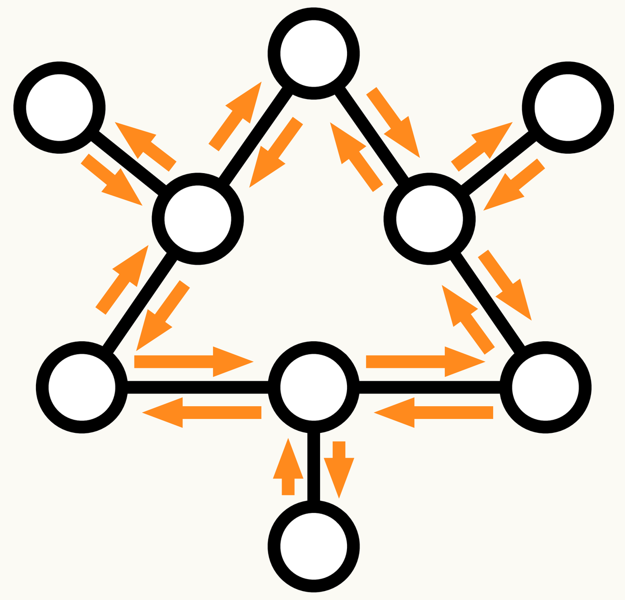
Relaxed Scheduling for Scalable Belief Propagation

Vitaly Aksenov · ITMO University
Dan Alistarh · IST Austria
Janne H. Korhonen · IST Austria

NEURAL INFORMATION PROCESSING SYSTEMS



Summary. We show how to efficiently parallelise priority-based belief propagation – such as residual belief propagation – in a shared-memory parallel setting. Our parallel algorithms are based on scalable relaxed priority queues (schedulers), and have state-of-the-art parallel scaling performance

Belief propagation

Given a graphical model over variables $X_1, X_2, \ldots, X_N$ with domain D defining a joint probability distribution

$$\Pr[X = x] \propto \prod_i \Psi_i(x_i) \prod_{ij} \Psi_{ij}(x_i, x_j)$$

the *inference problem* asks for computation of the marginal probabilities $\Pr[X_i = x_i]$ for some variables X_i. This is a NP-hard task, so in practice one uses heuristics or approximation algorithms to solve inference.

Belief propagation is a heuristic algorithm for inference. In belief propagation, messages $\mu_{i \to j} \in \mathbb{R}^{|D|}$ are associated with the edges of the graphical model, and an update rule

$$\mu_{i \to j}(x_j) \propto \sum_{x_i \in D} \psi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus \{j\}} \mu_{k \to i}(x_i)$$

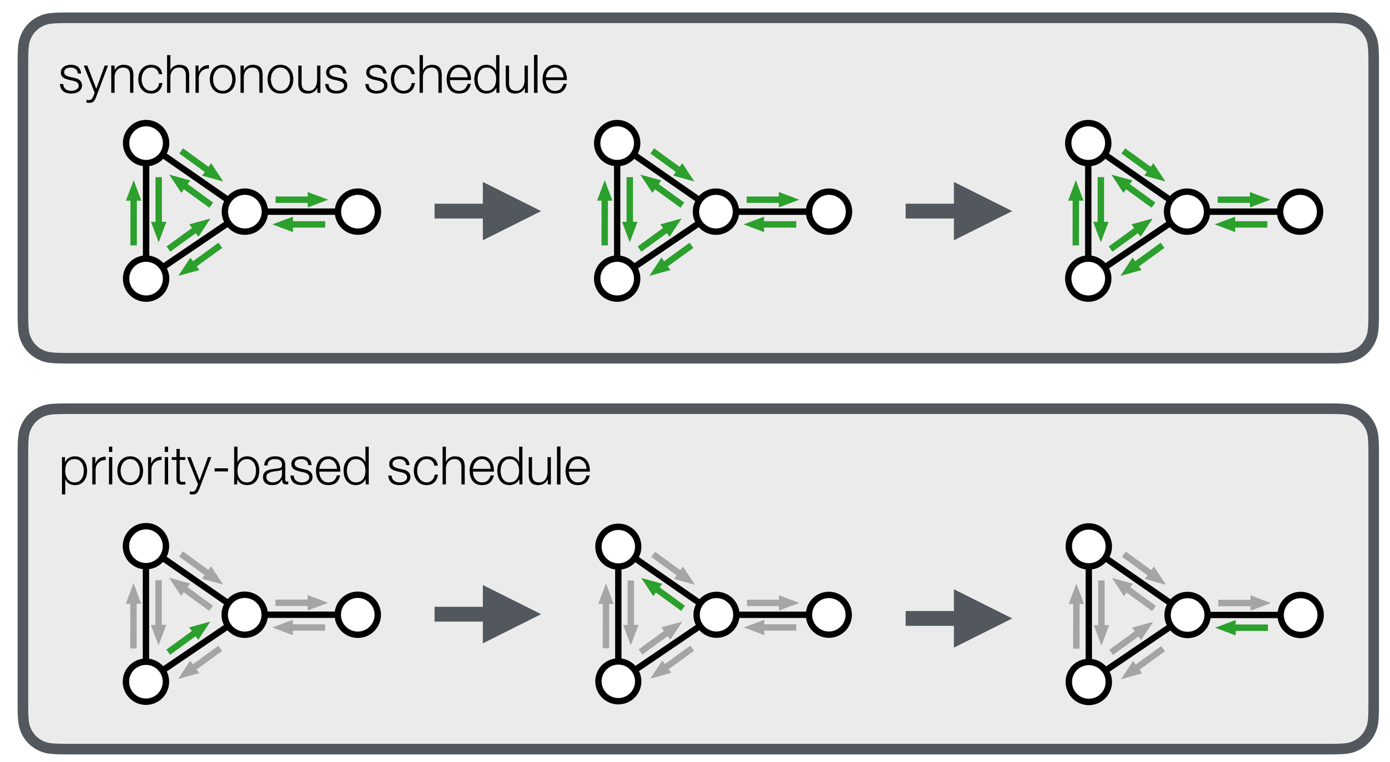
is applied to the messages until convergence to a fixed point. The marginals are then estimated as

$$\Pr[X_i = x_i] \propto \psi_i(x_i) \prod_{j \in N(i)} \mu_{j \to i}(x_i)\,.$$

In practice, belief propagation can be sped up using a *priority-based schedule* for updates – e.g. residual belief propagation [Elidan et al, UAI 2006] – instead of the obvious *synchronous* schedule. A priority function r is used decide which message is updated next. Concretely, this can be implemented by storing the messages in a priority queue Q, and iterating the following procedure:

1. Pop the top element for Q to obtain the message $\mu_{i \to j}$ with highest priority $r(\mu_{i \to j})$.
2. Update message $\mu_{i \to j}$.
3. Update the priorities in Q for messages affected by the update.

However, this template does not easily lend itself to efficient parallelisation, as the priority queue Q becomes a contention bottleneck.



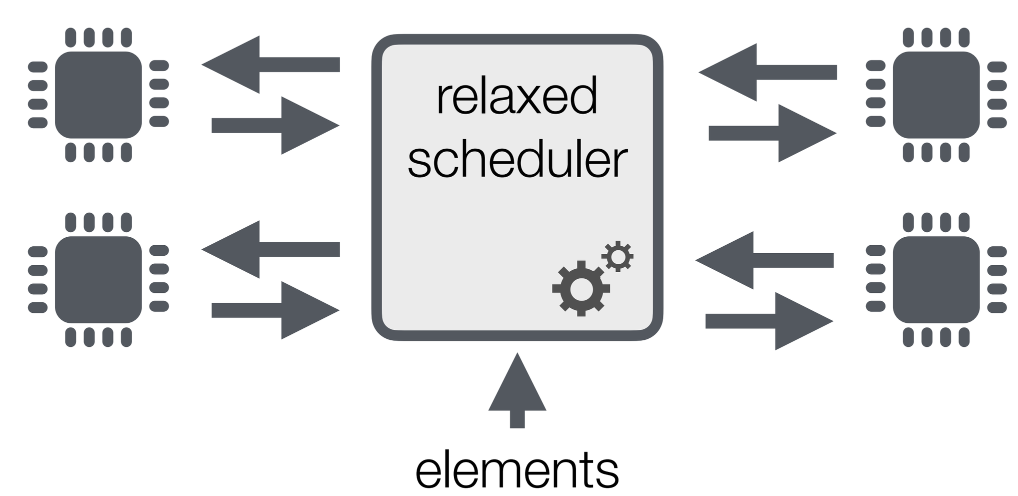
Relaxed schedulers

A *relaxed priority queue* (*scheduler*) is a concurrent data structure with semantics similar to a priority queue. Specifically, a relaxed priority queue Q is a data structure supporting `Insert(<key, priority>)` and `IncreaseKey(<key, priority>)` with the usual semantics, and an `ApproxDeleteMin()` operation with the following guarantees:

(a) `ApproxDeleteMin()` returns one of the top q elements in priority order.

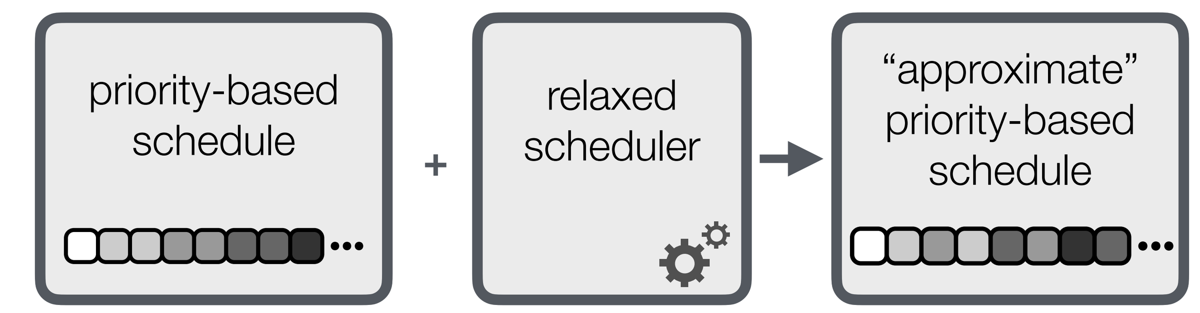
(b) Any element can be passed over in favour of an element with lower priority at most q times before it must be returned.

Unlike standard priority queues, relaxed schedulers can be implemented very efficiently in shared-memory parallel setting, and they have been used to efficiently parallelise many iterative algorithms.



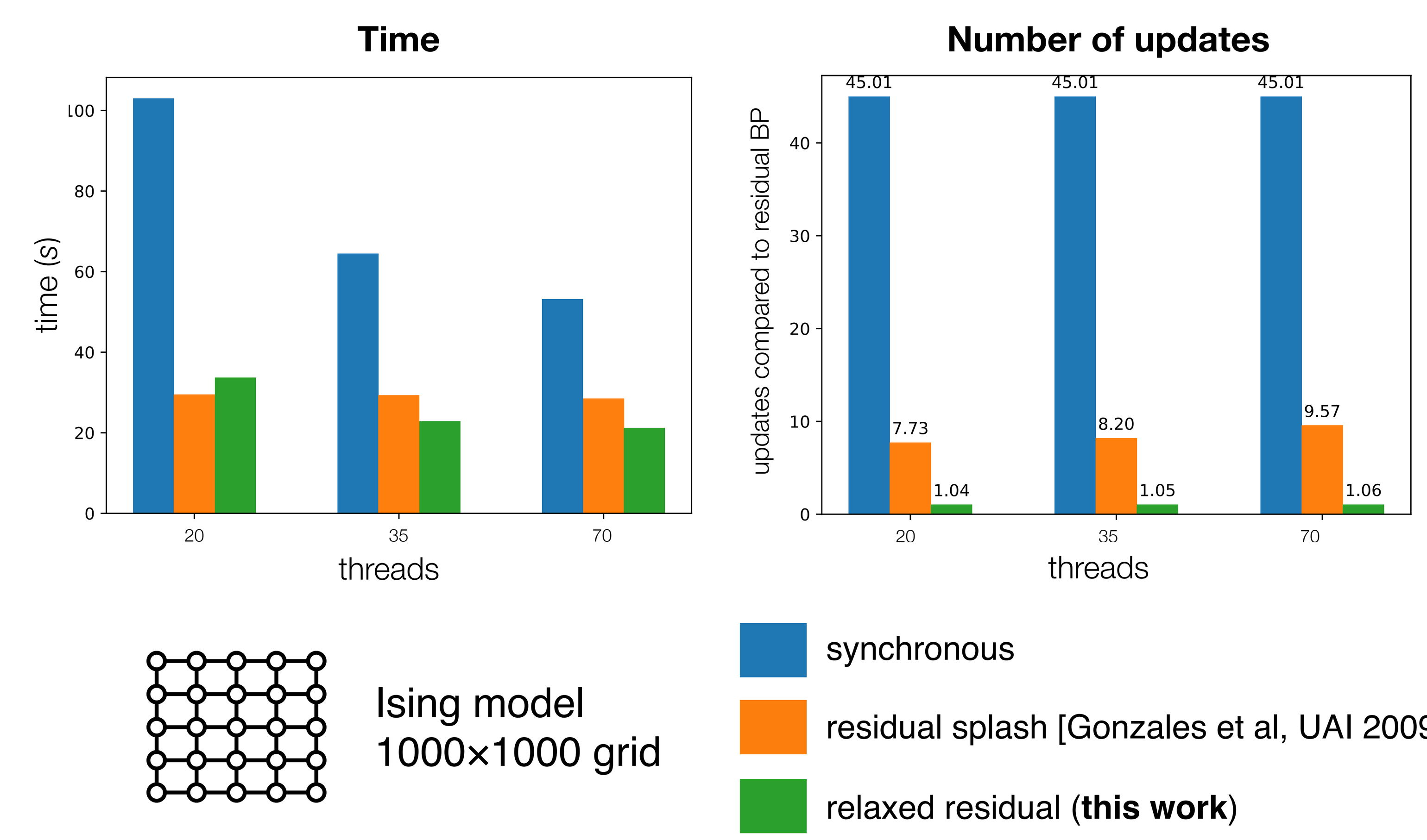
Relaxed priority-based belief propagation

A priority-based schedule for belief propagation can be implemented efficiently in shared-memory parallel setting by replacing the priority queue with a relaxed scheduler. The resulting schedule can be seen as an approximate version of the original, with similar dynamics.



Experiments

We implement our relaxed priority-based scheduling framework with a Multiqueue data structure [Rihani et al, SPAA 2015]. In the benchmarks, we show that this framework gives state-of-the-art parallel scalability on a wide variety of Markov random field models. As expected, the relaxed priority-based schedules require slightly more message updates than their exact counterparts, but this performance overhead is offset by their better scalability.



Above: Running time and number of updates until convergence on a 1000×1000 Ising grid model on $p \in \{20, 35, 70\}$ processes. Included algorithms are synchronous belief propagation, residual splash belief propagation of Gonzalez et al. [UAI 2009] with splash size 10, and our relaxed residual belief propagation. The number of updates is relative to sequential residual belief propagation.

For extended benchmarks with more algorithms and graphical models, please refer to the paper.

- https://arxiv.org/abs/2002.11505
- https://github.com/IST-DASLab/relaxed-bp