Quang Minh Ngo, Taha Erkoc

Applied Combinatorial Optimization – Exercise 3

(Block) Iterated Conditional Nodes Algorithm:

(i) What type of algorithm is it?

Block-coordinate descent method in the primal space. It operates on blocks of variables (subproblems) which are derived from the row/column decomposition of the grid graph. Each block gets optimized independently while treating the neighboring nodes outside the block as fixed.

(ii) Does the algorithm provide guarantees?

The algorithm converges to a local minimum of the energy function, however it does not guarantee convergence to the global minimum due to the discrete nature of the problem.

(iii) Give a rough estimate of the time complexity per iteration

It depends on the number of blocks and the size of each block

If each block contains b nodes, the cost of optimizing a block is $O(b*L^2)$ where L is the number of labels per node.

So, the total complexity per iteration would be $O(B*b*L^2)$ with B being the number of blocks.

(iv) Describe the quality of the output

The algorithm produces a labeling that is a local minimum of the energy function. While it doesn't guarantee the global optimum, the block-based updates generally result in better solutions than node-wise ICM.

Also, the row/column decomposition can exploit the structure of grid graphs effectively, leading to faster convergence and better-quality results compared to node-wise methods.

Subgradient Algorithm:

(i) What type of algorithm is it?

Dual optimization algorithm where it optimizes the dual objective of our graph model by iteratively updating dual variables. The algorithm is based on the subgradient method for convex optimization (non-differentiable)

(ii) Does the algorithm provide guarantees?

The algorithm provides convergence guarantees for convex problems where the duality gap diminishes over time if the step size is chosen correctly. However, the primal rounding solution may not always be optimal and depends on the rounding method.

(iii) Give a rough estimate of the time complexity per iteration

Let L be the number of labels per node, n the number of nodes and m the number of of edges. The cost per edge is $O(L^2)$ and for all edges $O(m*L^2)$

(iv) Describe the quality of the output

It heavily depends on the chosen step size. It is effective for larger-scale problems due to its simplicity and scalability and can provide good approximation for convex graphical models. However, it does not converge as fast, especially if the step size is chosen poorly.

MinSum-Diffusion inference Algorithm:

(i) What type of algorithm is it?

Dual block-coordinate ascent algorithm that iteratively reparametrizes the model by adjusting edge costs.

(ii) Does the algorithm provide guarantees?

It guarantees convergence to the global optimum for convex energy functions as well as acyclic graphs, but for general graphical models, it is not guaranteed.

(iii) Give a rough estimate of the time complexity per iteration

Let L be the number of labels per node, n the number of nodes and m the number of of edges. We have for each node the accumulation $O(\deg(u) * L^2)$ and for all nodes, this becomes $O(m * L^2)$. Same applies for the distribution phase $O(m*L^2)$
Thus the total complexity per iteration is: $O(m*L^2)$

(iv) Describe the quality of the output