Applied Combinatorial Optimization

Exercise sheet 3

Dr. Bogdan Savchynskyy <bogdan.savchynskyy@iwr.uni-heidelberg.de>
Sebastian Stricker <sebastian.stricker@iwr.uni-heidelberg.de>

⋆

General Notes The goal of this exercise is to implement the individual inference techniques that have been discussed in the lecture. In addition to the implementation you should answer the following questions for *each* of the following exercises:

(i) What type of algorithm is it?[1]

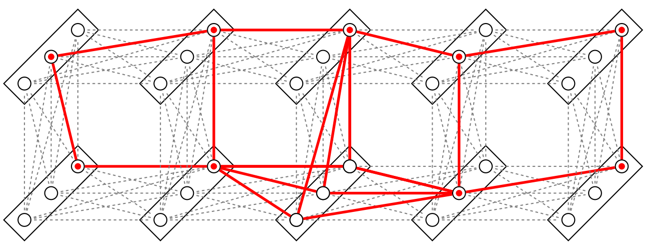
(ii) Does the algorithm provide guarantees?[2]

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

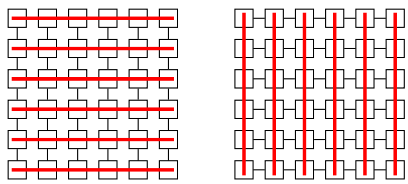
(iv) Describe the quality of the output.

The implementation should be fully functional on the provided *tsukuba* model. You are allowed to assume that your input is a grid graph, as this makes the implementation more straight-forward. Output the bounds of the objective after each iteration (when applicable). Plot the progress of the dual/primal objective with respect to the number of iterations and elapsed runtime. Include the output and the results in your submission. We provide helper functions for some technical subtasks (parsing of input files, visualization of labeling, fast grid graph access).

Exercise 3.1 [2 points] Implement the iterated conditional modes algorithm with the speedup discussed in the lecture. Extend it to block-ICM where you use a row/column-decomposition. Suggest and implement a generalization of the above speedup.



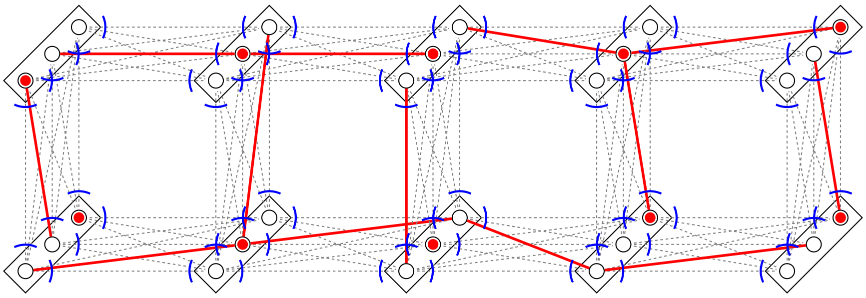
$$\min_{y_i} E(\theta, [y_1, \ldots, y_i, \ldots, y_n])$$



row / column decomposition

[1] e.g. primal/dual, subgradient/block-coordinate based, etc.
[2] e.g. convergence, approximation/exact optimization, etc.

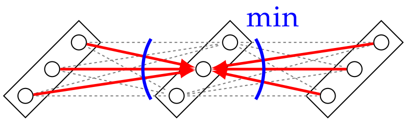
Exercise 3.2 [1 point] Implement the subgradient algorithm for graphical models. Use naïve rounding to obtain the label assignment.



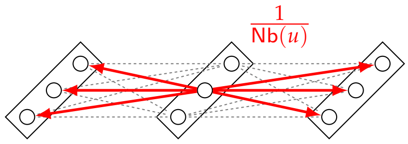
For the step-size selection implement the rule described in Remark 6.9 (Practical step-size rules), the third option given by Eq. (6.14)[3]

$$\hat{f}^t := \max\{\widetilde{f}^t, m^t/(1+\gamma^t)\}\,. \tag{1}$$

Exercise 3.3 [1 point] Implement the MinSum-Diffusion inference algorithm.



(1) accumulation



(2) distribution

Use the rounding scheme as discussed in the lecture draft.

[3]Since the lecture draft is constantly updated, the exact numbering may differ from the given one.