Graph-based Simplex Method for Pairwise Energy Minimization with Binary Variables

Daniel Průša
Faculty of Electrical Engineering, Czech Technical University in Prague.

Energy minimization is a well known NP-hard combinatorial problem which arises in MAP inference in graphical models. The simplest variant with pairwise interactions and binary variables has a prominent role. It is utilized by many approximation and reduction methods for multi-label instances or binary instances with high-order interactions. It can be expressed as quadratic pseudo-boolean optimization (QPBO) [1] and max-flow/min-cut algorithms can be applied to find a partial optimal solution, having some variables undecided. A complete optimal solution is always found by QPBO e.g. for submodular instances.

The key prerequisite for solving the problem efficiently has been so far to come up with a max-flow algorithm working well on vision instances. The popular algorithm by Boykov and Kolmogorov [2] fulfills this role. A recent empirical comparison of max-flow algorithms by Verma and Batra [5] reveals that some other contemporary implementations are more suitable for instances with dense graphs.

We show in this paper that there is a different principle which can be also turned into an efficient solver. The problem has a natural linear programming (LP) relaxation which is known to be half integral in the case of binary variables. This means that all components of each optimal solution are in $\{0, \frac{1}{2}, 1\}$. Moreover, the solution coincides with the result of QPBO since undecided variables are indicated by value $\frac{1}{2}$. An example of LP relaxation variables and constraints is given in Figure 1.

We present how the simplex algorithm can be tailored to solve the LP relaxation very efficiently in linear space. A special structure formed by basic and nonbasic variables in each stage of the algorithm is identified and utilized to perform the whole iterative process combinatorially over an input graph $G = (V, E)$ rather than algebraically over the simplex tableau. Note that customized versions of the simplex method with similar properties have already been proposed for *transportation*, *assignment*, *minimum cost-flow* or even max-flow problems. They are known as the *network simplex* algorithms [3].

To outline the main idea, consider a linear program

$$\min\{\langle \mathbf{c}, \mathbf{x}\rangle \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $m \leq n$, $\operatorname{rank}(\mathbf{A}) = m$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$. Let $\mathcal{B}$ be a basis, i.e., a set of m indices of linearly independent columns of $\mathbf{A}$. Let $\mathcal{N}$ be the set of remaining $n-m$ nonbasic indices. The constraint equations can be rewritten as $\mathbf{B}\mathbf{x}_\mathcal{B} + \mathbf{N}\mathbf{x}_\mathcal{N} = \mathbf{b}$ where $\mathbf{B}$ is invertible and $\mathbf{x}_\mathcal{B}$, $\mathbf{x}_\mathcal{N}$ are vectors of basic and nonbasic variables, respectively. This further gives $\mathbf{I}\mathbf{x}_\mathcal{B} + \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_\mathcal{N} = \mathbf{B}^{-1}\mathbf{b}$ which is the form essential for the simplex method as the simplex tableau is composed of elements of $\mathbf{I}$, $\mathbf{B}^{-1}\mathbf{N}$ and $\mathbf{B}^{-1}\mathbf{b}$.

Each basic variable x_i, $i \in \mathcal{B}$ can thus be expressed as a linear combinations of nonbasic variables:

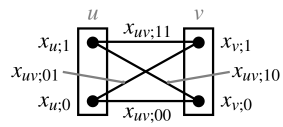
$$x_i = \beta_i - \sum_{j\in\mathcal{N}} \alpha_{ij} x_j .$$

We prove that, given a basis $\mathcal{B}$ and a basic variable x_i of the LP relaxation, β_i and nonzero coefficients α_{ij} can be retrieved easily from G. Assuming that x_i is a basic variable in an object $v \in V$, all nonbasic variables x_j such that $\alpha_{ij} \neq 0$ are located in objects and objects pairs that form a path starting in v which possibly closes a loop to itself. The union of all such paths creates a subgraph of G which we call a *dependency* graph. It consists of mutually disjoint *dependency* components that cover all the objects in G. Each of the components is a rooted tree or a subgraph with one cycle. An example is depicted in Figure 2. Similarly, for a basic variable in an object pair $e \in E$, all $\alpha_{ij} \neq 0$ are located in (up to two) adjacent dependency components. This theoretical characterization is utilized by the proposed graph-based simplex algorithm. We show it is sufficient to maintain the dependency graph and in each iteration to compute only a fraction of coefficients in the simplex tableau by traversing one or two dependency components. Experiments done using vision instances have revealed the following facts:

- The simplex algorithm performs $\mathcal{O}(|E| + |V|)$ iterations.
- The average size of dependency components is constant or nearly constant for nonsubmodular instances and submodular instances where unary potentials are more dominant than pairwise potentials.
- Dependency components are usually large in the case of submodular instances with dominant pairwise potentials.

We have verified the favorable problem classes using instances arising in *Decision Tree Field* (DTF), *Super Resolution*, *Deconvolution* and *Shape Fitting*. Running times of our algorithm were compared to those used by max-flow based QPBO solver by Kolmogorov. We achieved a better performance in the case of DTF and a reasonable, competitive response for the other problems.

We conclude that the proposed algorithm gives a practical benefit (due to its performance e.g. on DTF instances) and offers a good opportunity for further research since it is still relatively unexplored comparing to the max-flow problem which has been intensively studied for a long time. It also gives a hope for a generalization of the method to the LP relaxation of multi-label problems. This is a much more difficult problem, as hard as general LP [4], however, similar dependency structures are observable there. We think that the presented applicability of the simplex algorithm in the binary setting should encourage such attempts.



$$\begin{aligned} x_{t;0} + x_{t;1} &= 1 & t &\in \{u, v\} \\ x_{uv;k0} + x_{uv;k1} &= x_{u;k} & k &\in \{0, 1\} \\ x_{uv;0k} + x_{uv;1k} &= x_{v;k} & k &\in \{0, 1\} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Figure 1: An energy minimization over a graph with objects u, v forming an object pair $\{u, v\}$, together with its LP relaxation variables and constraints.

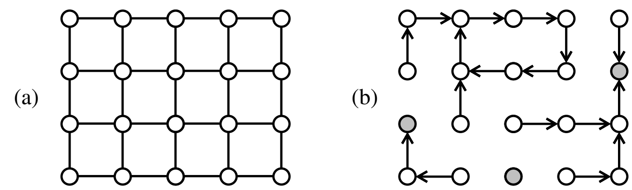


Figure 2: (a) An energy minimization graph and (b) an example of its dependency subgraph. Roots of tree components are colored in gray.

[1] Endre Boros and Peter L. Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[2] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[3] George B. Dantzig and Mukund N. Thapa. *Linear Programming 1: Introduction*. Springer, 1997.

[4] Daniel Průša and Tomáš Werner. Universality of the local marginal polytope. In *Conf. on Computer Vision and Pattern Recognition*, pages 1738–1743, 2013.

[5] Tanmay Verma and Dhruv Batra. Maxflow revisited: An empirical comparison of maxflow algorithms for dense vision problems. In *Proceedings of the British Machine Vision Conference*, pages 61.1–61.12, 2012.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.