Supplementary Material
LPQP for MAP: Putting LP Solvers to Better Use

Patrick Pletscher PLETSCHER@INF.ETHZ.CH
Sharon Wulff SHARON.WULFF@INF.ETHZ.CH
Department of Computer Science, ETH Zurich, Switzerland

A. Convergence

Proposition 1. *The convex-concave procedure in Algorithm 1 converges to a stationary point of the LPQP objective in (5) with $\rho = \rho_{final}$, the paremeter value reached when the marginals do not change further.*

Proof. It was shown in (Sriperumbudur & Lanckriet, 2009) that the CCCP with a convex constraint set converges to a stationary point of the objective. In the last DC iteration, a CCCP is solved with $\rho = \rho_{\text{final}}$. □

B. Dual Decomposition

In the dual decomposition framework the global variables, in our case $\boldsymbol{\mu}$, are replaced with local copies of the variables for each sub-problem, denoted here by $\boldsymbol{\nu}^a$. To enforce the multiple copies of the local variables to assume the same value, a designated constraint is introduced. In the literature, the sub-problems are called slave problems whereas the sum over the slave problems, subject to the unifying constraint, is referred to as the master problem.

We consider the following master problem

$$\sum_{a\in\mathcal{A}} \min_{\boldsymbol{\nu}^a\in\mathcal{L}_{\mathcal{G}_a}} s_a(\boldsymbol{\nu}^a) \tag{15}$$
$$\text{s.t.} \quad \boldsymbol{\nu}_i^a = \frac{1}{|\mathcal{A}(i)|}\sum_{a'\in\mathcal{A}(i)} \boldsymbol{\nu}_i^{a'} \quad \forall i, a\in\mathcal{A}(i)$$
$$\boldsymbol{\nu}_{ij}^a = \frac{1}{|\mathcal{A}(i,j)|}\sum_{a'\in\mathcal{A}(i,j)} \boldsymbol{\nu}_{ij}^{a'} \quad \forall (i,j), a\in\mathcal{A}(i,j).$$

Here we use the idea from Domke (2011) who formulates the constraint on the replicated marginal variables to agree with the mean. This is simpler than the constraint in (13). We can write the Lagrangian and rearrange to get

$$L(\boldsymbol{\nu}^1,\dots,\boldsymbol{\nu}^{|\mathcal{A}|},\boldsymbol{\lambda}) = \sum_{a\in\mathcal{A}} \min_{\boldsymbol{\nu}^a\in\mathcal{L}_{\mathcal{G}_a}} \left(s_a(\boldsymbol{\nu}^a) + \sum_{i\in\mathcal{V}_a} \boldsymbol{\theta}_i^a(\boldsymbol{\lambda})\boldsymbol{\nu}_i^a + \sum_{(i,j)\in\mathcal{E}_a} \boldsymbol{\theta}_{ij}^a(\boldsymbol{\lambda})\boldsymbol{\nu}_{ij}^a \right),$$

with

$$\boldsymbol{\theta}_i^a(\boldsymbol{\lambda}) = \boldsymbol{\lambda}_i^a - \frac{1}{|\mathcal{A}(i)|}\sum_{a'\in\mathcal{A}(i)} \boldsymbol{\lambda}_i^{a'}$$
$$\boldsymbol{\theta}_{ij}^a(\boldsymbol{\lambda}) = \boldsymbol{\lambda}_{ij}^a - \frac{1}{|\mathcal{A}(i,j)|}\sum_{a'\in\mathcal{A}(i,j)} \boldsymbol{\lambda}_{ij}^{a'}.$$

The Lagrange multipliers vector $\boldsymbol{\lambda}$ is of the same length as all the $\boldsymbol{\nu}$ concatenated together, where for variables that are only replicated once, the corresponding Lagrange multiplier can be dropped. We can think of the potentials as being a function of $\boldsymbol{\lambda}$ and thus the dual problem of (15) is given by

$$\max_{\boldsymbol{\lambda}} \sum_{a\in\mathcal{A}} \min_{\boldsymbol{\nu}^a\in\mathcal{L}_{\mathcal{G}_a}} s_a(\boldsymbol{\nu}^a,\boldsymbol{\lambda}).$$

Here $s_a(\boldsymbol{\nu}^a,\boldsymbol{\lambda})$ is defined by

$$s_a(\boldsymbol{\nu},\boldsymbol{\lambda}) := \sum_{i\in\mathcal{V}_a} \bar{\boldsymbol{\theta}}_i^a(\boldsymbol{\lambda})^\mathsf{T}\boldsymbol{\nu}_i + \sum_{(i,j)\in\mathcal{E}_a} \bar{\boldsymbol{\theta}}_{ij}^a(\boldsymbol{\lambda})^\mathsf{T}\boldsymbol{\nu}_{ij} - \rho\eta_a H_{\text{tree}}^a(\boldsymbol{\nu}),$$

and the modified potentials are given as:

$$\bar{\boldsymbol{\theta}}_i^a(\boldsymbol{\lambda}) = \bar{\boldsymbol{\theta}}_i + \boldsymbol{\theta}_i^a(\boldsymbol{\lambda})$$
$$\bar{\boldsymbol{\theta}}_{ij}^a(\boldsymbol{\lambda}) = \bar{\boldsymbol{\theta}}_{ij} + \boldsymbol{\theta}_{ij}^a(\boldsymbol{\lambda}).$$

All the slave computations can be carried out exactly using the sum-product algorithm. For the maximization w.r.t. $\boldsymbol{\lambda}$ we use FISTA (Beck & Teboulle, 2009), a modern variant of Nesterov's traditional fast gradient method (Nesterov, 1983). Our dual decomposition approach is very similar to (Savchynskyy et al., 2011),

with two key differences. First, the authors study only a specific choice of the decomposition for the 4-connected grid graph in which each node marginal is replicated twice and each edge is only considered once. Second, we are also interested in settings of $\rho \gg 0$, which is not meaningful in the context of the standard LP relaxation.

The algorithm we used for solving the master problem is given in Algorithm 2. It is based on FISTA descent as described in (El Ghaoui, 2012; Vandenberghe, 2012).

Algorithm 2 FISTA ascent for the master problem.

initialize $\boldsymbol{\lambda}^{(0)} = \boldsymbol{v}^{(0)}$, $k = 1$.
repeat
  $\theta_k = 2/(k+1)$.
  $\boldsymbol{y} = (1-\theta_k)\boldsymbol{\lambda}^{(k-1)} + \theta_k \boldsymbol{v}^{(k-1)}$.
  $\boldsymbol{u} = \boldsymbol{y} + t_k \nabla f(\boldsymbol{y})$, perform line-search for t_k.
  ensure ascent for $\boldsymbol{\lambda}^{(k)}$:

$$\boldsymbol{\lambda}^{(k)} = \begin{cases} \boldsymbol{u} & f(\boldsymbol{u}) \geq f(\boldsymbol{\lambda}^{(k-1)}) \\ \boldsymbol{\lambda}^{(k-1)} & \text{otherwise.} \end{cases}$$

  $\boldsymbol{v}^{(k)} = \boldsymbol{\lambda}^{(k-1)} + \frac{1}{\theta_k}(\boldsymbol{u} - \boldsymbol{\lambda}^{(k-1)})$.
  $k = k+1$.
until converged.
return $\boldsymbol{\lambda}^{(k-1)}$.

C. Additional Experiments

Figure 5 and 6 show the run time of the LPQP algorithm for different initial ρ_0 for a grid graph of size 40×40 with $K = 3$. We can see that for smaller values of ρ_0 one generally obtains better solutions. For

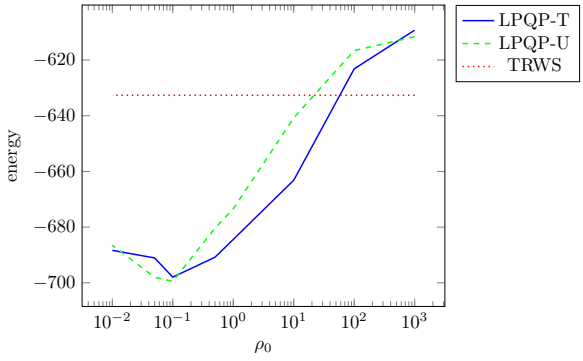

Figure 5. Energy of the solution found by the LPQP-U and LPQP-T algorithm as a function of the initial ρ_0.

larger values of ρ_0, the optimization problem becomes more and more similar to the standard QP relaxation, as violations in the pairwise marginals are strongly penalized. The run time of the algorithms however also increases substantially for smaller ρ_0, especially for LPQP-T.

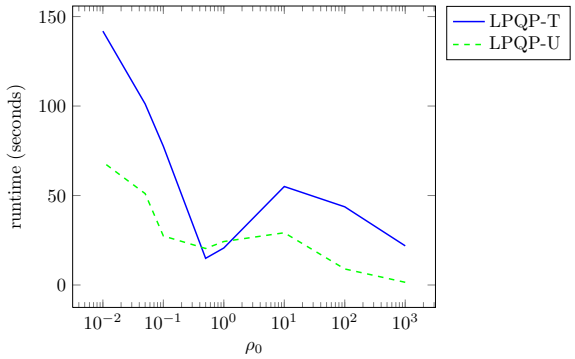

Figure 6. Run time of the two different LPQP solvers. For smaller ρ_0 the run time of LPQP-T is much worse affected than the one of LPQP-U.

Finally, Figure 7 visualizes the run time of LPQP-U, MPLP and TRWS for the protein design dataset in a box plot. The mean speedup of LPQP-U over MPLP is slightly above two, but the variance is much higher. TRWS is substantially faster than LPQP and TRWS.

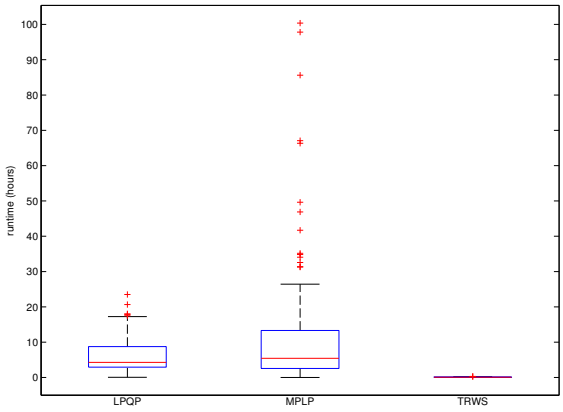

Figure 7. Run time of the three solvers for the protein design experiment. We excluded the instance where MPLP did not finish within 7 days (168 hours).

Supplement References

Beck, A and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1), 2009.

Domke, J. Dual decomposition for marginal inference. In *AAAI*, 2011.

El Ghaoui, L. Proximal gradient method, 2012. Lecture notes.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet. Math. Dokl.*, 27:372–376, 1983.

Savchynskyy, B, Kappes, J H, Schmidt, S, and Schnörr, C. A study of Nesterov's scheme for Lagrangian decomposition and MAP labeling. In *CVPR*, pp. 1817–1823, 2011.

Sriperumbudur, B. and Lanckriet, G. On the convergence of the concave-convex procedure. In *NIPS*, pp. 1759–1767. 2009.

Vandenberghe, L. Fast proximal gradient methods, 2012. Lecture notes.