COLLAPSED DUAL SOLUTION OF LARGE BINARY INTEGER LINEAR PROGRAMS

Bob Agnew, raagnew1@gmail.com, raagnew.com

We have been using the collapsed dual approach to solve large marketing optimization (generalized assignment) problems since 2004, including marketing campaigns to millions of customers at Discover Financial Services in the late 2000s. More recently, we extended this methodology to constrained k-means clustering. Here we make a natural extension to large binary integer linear programs (BILPs). In a recent publication, Xiaocheng Li, Chunlin Sun, and Yinyu Ye have advocated the collapsed dual approach, but they seem to conclude that a special algorithm is required. Actually, the collapsed dual can be solved via a simple, plain-vanilla nonlinear minimizer and the BILP can then be approximated with good precision via a simple sort.

The standard BILP is

Maximize: $\sum_{j=1}^{n} c_j \, x_j$

Subject to: $\sum_{j=1}^{n} a_{ij} \, x_j \;\le\; b_i \;,\; i \;=\; 1,\ldots,m$

$x_j \;\in\; \{0,1\} \;,\; j \;=\; 1,\ldots,n$

This BILP can be relaxed to a linear program by replacing the binary constraints with $0 \le x_j \le 1$, $j = 1,\ldots,n$. The dual of this relaxed LP is

Minimize: $\sum_{i=1}^{m} b_i \, y_i \;+\; \sum_{j=1}^{n} z_j$

Subject to: $\sum_{i=1}^{m} a_{ij} \, y_i \;+\; z_j \;\ge\; c_j \;,\; j \;=\; 1,\ldots,n$

$y_i \;,\; z_j \;\ge\; 0 \;,\; i \;=\; 1,\ldots,m \;,\; j \;=\; 1,\ldots,n$

but clearly at a minimum $z_j \;=\; max\left(\, 0,\; c_j - \sum_{i=1}^{m} a_{ij} \, y_i \right)$ leading to the collapsed dual

Minimize: $\sum_{i=1}^{m} b_i\, y_i \;+\; \sum_{j=1}^{n} max\left(\,0,\; c_j - \sum_{i=1}^{m} a_{i\,j}\, y_i\,\right)$

Subject to: $y_i \;\geq\; 0\;,\;\; i \;=\; 1,\ldots,m$

This objective function is not smooth, but it is continuous and convex, and it is solvable via a simple, plain-vanilla nonlinear minimizer like *nlminb* in R. If y^* denotes the quasi-optimal dual solution, we then rank-order $c_j - \sum_{i=1}^{m} a_{i\,j}\, y_i^*$ descending and make binary $x_j \;=\; 1$ allocations until they are no longer feasible. For large BILPs, the solutions are quasi-optimal, as guaranteed by the dual upper bound $\sum_{i=1}^{m} b_i\, y_i^* \;+\; \sum_{j=1}^{n} max\left(\,0,\; c_j - \sum_{i=1}^{m} a_{i\,j}\, y_i^*\,\right)$. We were able to solve a large BILP with 100 rows and 50,000 columns in about four minutes on a beefy Dell PC, starting with all $x_j \;=\; 0$ (assuming all $b_i \;\geq\; 0$). The resulting BILP value was within 99.96% of the dual upper bound. We provide a simple R-function *bilp.max* for this task along with corresponding R-function *bilp.min* for a minimization task, starting with all $x_j \;=\; 1$ (assuming this is feasible).

References

Xiaocheng Li, Chunlin Sun, and Yinyu Ye, “Simple and Fast Algorithm for Binary Integer and Online Linear Programming,” 34th Conference on Neural Information Processing Systems (Neur IPS 2020), Vancouver, Canada. https://proceedings.neurips.cc/paper/2020/file/6abba5d8ab1f4f32243e174beb754661-Paper.pdf.

https://communities.sas.com/t5/Mathematical-Optimization/Dual-Solution-of-Large-Marketing-Optimization-Problems-using/m-p/63284

https://www.r-bloggers.com/2011/07/marketing-optimization-using-the-nonlinear-minimization-function-nlm/

https://github.com/raagnew/r_utility_code/blob/master/mkt_opt.R

https://github.com/raagnew/Constrained-K-Means-Clustering-in-R

https://github.com/raagnew/Marketing-Optimization-in-Python