Supplementary Material for "Learning Fast Optimizers for Contextual Stochastic Integer Programs"

July 6, 2018

1 SUPPLEMENTARY MATERIAL

1.1 DETAILS OF DUAL DECOMPOSITION

We describe the dual decomposition approach in more detail. We start with the optimization problem:

$$\max_{x\in\{0,1\}^n} c_1(z)^T x + \frac{1}{N_\omega}\sum_{\omega\in D_\omega}\left[\max_{y\in Y(x,\omega,z)} c_2(\omega,z)^T y\right]$$

We first introduce independent copies of x for each scenario ω. We can state this as

$$\max_{x,\{x_\omega\}} \frac{1}{N_\omega}\sum_{\omega\in D_\omega}\left[\max_{y\in Y(x_\omega,\omega,z)} c_1(z)^T x_\omega + c_2(\omega,z)^T y\right] \quad (1)$$
$$\text{s.t } x_\omega = x \quad \forall\omega\in\Omega$$

The constraints in the problem are known as *non-anticipativity* constraiants, since the enforce that x has to be chosen independent of the value of ω (i.e., at the first stage, we do not know the precise realization of the second stage randomness). We obtain a relaxation of the above problem by dropping the non-anticipativity constraints and simply adding a Lagrangian term that tries to enforce the constraint:

$$\max \sum_{\omega\in D_\omega}\frac{1}{N_\omega}\left[\max_{\substack{y\in\\ Y(x_\omega,\omega,z)}} c_1^T x_\omega + c_2^T y + \lambda_\omega^T(x-x_\omega)\right] \quad (2)$$

where we dropped the dependce of c_1, c_2 on ω, z for brevity and the outer maximization is over $x, \{x_\omega\}$ For any choice of $\{\lambda_\omega\}_{\omega\in\Omega}$, the optimal value of this optimization problem is an upper bound on the optimal value of (1) - this follows from *weak duality* [Carøe and Schultz, 1999]. A quick way to see that this is true is as follows: Any

feasible solution for (1) satisfies $x = x_\omega$ and hence the Lagrangian terms involving λ disappear and the objective reduces to the objective of (1). Thus, every feasible solution of (1) is feasible for (2) and for every such solution the objectives of (2) and (1) coincide. Thus, the optimal value of (2) is guaranteed to be larger than or equal to the optimal value of the (1).

We can simplify (2) by observing that the maximization over x_ω can be done independently for each ω:

$$\frac{1}{N_\omega} \sum_{\omega \in D_\omega} \max_{\substack{x_\omega, y: \\ B(\omega,z)x_\omega + C(\omega,z)y \leq d(\omega,z)}} c_1^T x_\omega + c_2^T y - \lambda_\omega^T x_\omega + \max_{x \in \{0,1\}^n} \left(\frac{1}{N_\omega} \sum_{\omega \in D_\omega} \lambda_\omega \right)^T x \tag{3}$$

We denote

$$h(\omega, z, \lambda_\omega) = \max_{B(\omega,z)x_\omega + C(\omega,z)y \leq d(\omega,z)} c_1(z)^T x_\omega + c_2(\omega, z)^T y - \lambda_\omega^T x_\omega$$

Note also that

$$\frac{\partial h(\omega, z, \lambda_\omega)}{\partial \lambda_\omega} = \operatorname*{argmax}_{x: B(\omega,z)x_\omega + C(\omega,z)y \leq d(\omega,z)} c_1(z)^T x_\omega + c_2(\omega, z)^T y - \lambda_\omega^T x_\omega$$

Then (2) reduces to

$$\frac{1}{N_\omega} \sum_{\omega \in D_\omega} h(\omega, z, \lambda_\omega) + \mathbf{1}^T \max \left(\frac{1}{N_\omega} \sum_{\omega \in D_\omega} \lambda_\omega, 0 \right) \tag{4}$$

Since this quantity is an upper bound on the optimal value of (1) for every choice of λ_ω, we can optimize the upper bound leading to the *dual problem*:

$$\min_{\{\lambda_\omega\}} \frac{1}{N_\omega} \sum_{\omega \in D_\omega} h(\omega, z, \lambda_\omega) + \mathbf{1}^T \max \left(\frac{1}{N_\omega} \sum_{\omega \in D_\omega} \lambda_\omega, 0 \right) \tag{5}$$

Theorem 1 (From the results in [Carøe and Schultz, 1999])**.** *The optimal value of* (5) *is a bound on the optimal value of* (1) *and further* (5) *is a convex optimization problem.*

Proof. The bound argument was sketched informally above and a formal proof can be found in [Carøe and Schultz, 1999]. The convexity of the objective can be easily checked from the definition of h, which is simply a maximum over linear functions of λ_ω. □

1.2 TABU SEARCH

Tabu Search (Glover [1986]) makes local moves in the space of the first stage decision variable x and maintains a list of recently visited solutions ("tabu" list) to avoid cycles. The algorithms starts with a random initialization in $\{0,1\}^n$ during training. At each iteration, a minibatch of values of ω $S = \{\omega^s\}$ and a subset of dimensions $B = i_1, i_2, \ldots, i_k$ are sampled. Then, the effect of flipping each bit in the index set B is evaluated by averaging the objective over the minibatch S. The best performing flip is then accepted (if it does better than the current solution and if the flip does not lead to a solution in the tabu list). If the solution is accepted, it is added to the tabu list displacing the oldest member (if the size of the tabu list has reached its limit).

The limit on the size of the tabu list, the minibatch size and the size of the dimension set B are hyperparameters that are tuned by observing the value that give the maximum improvement in the objective over the validation set, averaged over the set of training context vectors z

Also, for each training context z, the optimal solution found by the hill climbing algorithm is stored.

At test time, given a test context z, we look at the nearest neighbor z in the training context set, retrieve the optimal solution for that context, and initalize the tabu search with that solution with the hyperparmaters chosen based on tuning on the training contexts.

1.3 PROGRESSIVE HEDGING

We implement the progressive hedging algorithm [Watson and Woodruff, 2011] that is based on the dual decomposition approach. This algorithm has been successfully applied to several stocahstic integer programming problems. The main idea in this algorithm is to fix the relaxation introduced in the dual decomposition by adding penalty terms to promote consistency between x_ω and x. Specifically, an additional penalty of the form $\rho\|x_\omega - x\|^2$ is added to the objective to promote consistency between solutions and the dual variables λ_ω are updated with a gradient step at each iteration. If all scenarios converge to the same solution, the algorithm is terminated. Otherwise, if the algorithm fails to converge, the x_ω are averaged over all the scenarios and the averaged solution is rounded to binary values and returned.

References

C. C. Carøe and R. Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45, 1999.

F. Glover. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13(5):533–549, 1986.

J.-P. Watson and D. L. Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, Nov 2011.