Review of "Fractional Decomposition Tree Algorithm: A tool for studying the integrality gap of Integer Programs"

Summary

The (Goemans-)Carr-Vempala theorem characterizes the integrality gap of an integer program as the smallest constant C such that for all $x \in \mathrm{LP}$, we can find a convex combination $\hat{x} = \theta_1\hat{x}_1 + \cdots + \theta_k\hat{x}_k$ where $x_i \in \mathrm{IP}$ for all i and $\hat{x} \leq Cx$.

The authors give a computational variant of this result. More precisely, they give an algorithm (Fractional Decomposition Tree) which for any binary integer program with bounded integrality gap g and any x in the LP relaxation, finds a convex combination $\hat{x} = \theta_1\hat{x}_1 + \cdots + \theta_k\hat{x}_k$ of integer points with $\hat{x} \leq g^{|\mathrm{supp}(x)|}x$. They also give an analogous algorithm for the (non-binary) 2EC integer program.

As part of their FDT algorithm, the authors also propose a new heuristic (Dom2IP) that finds a feasible integral solution dominating a given fractional solution for any integral program with bounded integrality gap.

The authors provide various computational results comparing the FDT algorithm to heuristics on the Vertex Cover Problem (VC), the Tree Augmentation Problem (TAP), and the Minimum-Cost 2-Edge-Connected Multi-Subgraph Problem (2EC). Lastly, they use their algorithm to show that 2EC has an integrality gap of at most $\frac{6}{5}$ on certain small instances.

Evaluation

A computational version of the Carr-Vempala theorem is of theoretical interest even if the bound of $g^{|\mathrm{supp}(x)|}$ is quite weak compared to the optimum of g. However, it should be noted that by another result of Carr and Vempala, there exists a polynomial time algorithm that finds a convex combination of integral solutions bounded by rx where r is the approximation guarantee of some polynomial time approximation algorithm relative to the LP relaxation. So, usually, if a constructive bound on the integrality gap is known, this algorithm yields a better theoretical guarantee than the FDT algorithm.

The computational results seem quite weak to me. The classic Feasibility Pump heuristic is known to produce solutions of fairly poor quality; it is designed primarily to produce feasible solutions to potentially difficult MIPs quickly. On TAP and 2EC, the authors compare to problem-specific approximation algorithms that are not particularly designed to produce high quality solutions either.

Aside from that, the paper is well-written and the proofs seem correct to me (except for some minor issues; see below). I recommend acceptance with minor revisions.

Major Comments

1. The abstract is misleading in multiple ways. First, it says that the paper will give a constructive version of the Carr-Vempala theorem but does not mention the rather large loss in quality of g to (potentially) g^n. Secondly, it says that "[Dom2IP] more quickly determines if an instance has an unbounded integrality gap" but really this algorithm will only sometimes provide proof that the integrality gap is unbounded; it is not guaranteed to certify this. Lastly, it is claimed that Christofides' algorithm is the "best previous approximation algorithm". But Christofides is only best wrt. its theoretical approximation guarantee not its practical performance which is what is compared in the paper.

2. The paper should mention the randomized metarounding algorithm by Carr and Vempala and compare it to the FDT algorithm.

3. In the proofs of Lemma 8 and Lemma 11, I do not see why it can be assumed that $\lambda_0, \lambda_1, \lambda_2 > 0$. I think one has to allow the case in which the branching step returns just 1 or (in the case of 2EC) just 2 vectors.

4. As noted before, I find the comparisons against the FP heuristic on VC and the 2-approximation for TAP rather dubious. To show that the FDT algorithm performs much better than the theoretical $g^{|\text{supp}(x)|}$ bound, just comparing to the initial fractional solution is all that is needed. On the other hand, if the goal is to show that the FDT algorithm works well as a general purpose feasibility heuristic, then comparing to a FDT to a more modern variant of FP (e.g. as in the Feasibility Pump 2.0 paper) would be more reasonable.

Minor Comments

1. The abstract says "Dom2IP" but the paper mostly uses "DomToIP".

2. In Theorem 3, I think $\mathcal{D}(S(I))$ should just be $S(I)$.

3. On page 6, "which runs in polynomial time algorithm" should be "which runs in polynomial time".

4. On page 6, the use of θ_0 and θ_1 is not consistent with how γ_0 and γ_1 are defined in Lemma 8.

5. In Theorem 6, is $E_x = \text{supp}(x)$?
6. On page 8, the wording "We use the idea of fundamental extreme point to create the hardest LP solutions to decompose." seems to suggest that this is something that will be done in this paper.
7. On page 8, you refer to Figure 2 but this figure only shows up on page 23.
8. You mention that the Dom2IP algorithm can be used to prove that an IPs integrality gap is unbounded. How effective is this?
9. Lemma 9 requires $\gamma_0 + \gamma_1 \leq 1$. Perhaps this should already be included in Lemma 8.
10. In the proof of Lemma 9, "We need to show that $\gamma_0 \hat{x}_j^0 + \gamma_1 \hat{x}_j^1 \leq g x'_j$." has an extra g in it.
11. On page 13, "nodes in level L_i" should be "nodes in level i" and "points in the leaves" should be just "leaves."
12. On page 24, "applying this tool ... can be a tool" is redundant.