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

Summary

The paper’s results can essentially be divided into two parts. In the first one, the authors present a new algorithm called “Fractional Decomposition Tree” which in polynomial time finds a feasible solution of an integer program if its integrality gap is bounded. The integrality gap $g(I)$ is defined here as the largest gap over all objective functions, i.e. $g(I) = \min_c \frac{\text{LP-OPT}(c)}{\text{ILP-OPT}(c)}$, where $c \in \mathbb{R}^n$ is an objective vector and LP-OPT(c) or ILP-OPT(c) are the values of the continuous or integer optima, respectively.

In the second part, the authors turn to specific problems such as TAP (tree augmentation problem), 2EC (finding a minimum 2-edge-connected multi-subgraph). They show some new theoretical results about these problems in which FDT is used, and they also do some experiments.

Evaluation

The results seem new and interesting to me. It’s a little difficult to judge because I do not belong to the primary audience of the paper, but I think the paper should be accepted with some revisions. The revisions would be to clarify some of the results, and, more importantly, to set the results better in the context of existing work. This is particularly necessary because the definition of integrality gap seems non-standard to me, and it gave me quite a hard time to really understand what’s going on here.

I liked the experimental results and I consider them an important contribution of the paper.

Main Comments

1. Non-standard definition of integrality gap

Let I be an LP/ILP instance, i.e., a system $Ax \leq b$, and denote by LP-OPT(c) and ILP-OPT(c) the values of the continuous and integer optima with respect to an objective function $c \in \mathbb{R}^n$, respectively. As mentioned, the authors use the following definition of integrality gap:

$$g(I) = \min_c \frac{\text{LP-OPT}(c)}{\text{ILP-OPT}(c)},$$

where $c \in \mathbb{R}^n$ is an objective function vector.

However, it appears much more common to me [1,2] to define the gap to be the worst gap for a *fixed* objective but over a *varying* family of instances. Let $\mathcal{I} \subseteq \mathbb{Z}^m$ to be a set of right-hand sides which define a family of instances, i.e., we have one instance I for each $b \in \mathcal{I}$, and this instance is precisely $\min cx$ s.t. $Ax \leq b$. Then, the more common definition of the integrality gap is

$$g(\mathcal{I}, c) = \min_{I \in \mathcal{I}} \frac{\text{LP-OPT}(I, c)}{\text{ILP-OPT}(I, c)},$$

where LP-OPT and ILP-OPT are defined naturally to be the optimum over I with the objective c.

The reason why this difference matters so much is the following. There are problems for which some upper bound on the integrality gap g is known, and it is also known how to solve the LP relaxation (this is not trivial because the LP under considetation is a Configuration LP with an exponential number of variables), but it is **not known** how to obtain ILP solutions attaining the gap [2]. In other words, sometimes we know that a good solution exists, but we don't know how to construct it. So if the present paper gave a general algorithm which for an ILP finds a feasible solution attaining the gap, it would be a big deal.

Let me take a small detour here: the authors never claim that their algorithm actually attains the gap, so maybe the catch lies here? The answer is no, the catch really is in the definition of the integrality gap. With the author's definition, it is easy to compute the LP optimum LP-OPT(c) and then add a constraint $cx \leq g(I) \cdot \text{LP-OPT}(c)$. Clearly this new augmented instance will still be integrally feasible (by the definition of $g(I)$), but any integer feasible solution is also at least as good as $g \cdot \text{LP-OPT}(c)$. It would be good to add this little claim to the paper – not only can a feasible solution be found, the gap can also be attained.

So clearly the difference in the definition of integrality gap is significant. The question is how to address it. Here are my thoughts and suggestions:

1. Is your definition of integrality gap used elsewhere? If not, try to use a different word - maybe something like "objective-oblivious integrality gap"? The point is to make it clear that yes, this is something like integrality gap, but it does NOT match anything the reader has seen before.
2. Mention the existing work on integrality gaps and compare your results to it.
3. Mention known work on obtaining integer feasible solutions. Is FDT the first algorithm which constructs a feasible solution if the instance has finite integrality gap? For which definitions of integrality gap can this be done in polynomial time, and with what quality of the resulting solution (wrt the integrality gap)?
4. Explain why FDT cannot be used to obtain a feasible solution attaining $g(\mathcal{I}, c)$ – is it because despite $g(\mathcal{I}, c)$ being finite, $g(I)$ is infinite? or would FDT find some solution, but of worse value because $g(I)$ is much larger than $g(\mathcal{I}, c)$, or can both scenarios happen?

[1] Friedrich Eisenbrand, Gennady Shmonin: Parametric Integer Programming in Fixed Dimension. Math. Oper. Res. 33(4): 839-850 (2008)

[2] Klaus Jansen, Lars Rohwedder: A note on the integrality gap of the configuration LP for restricted Santa Claus. CoRR abs/1807.03626 (2018)

2. Quality / clarity of writing

This is definitely less important than the first point, but it seemed to me that the paper is not written particularly well. Overall, I would appreciate more narrative in what is the intuitive meaning of individual statements (lemmas etc.), and to have some proof overview / plan such as "we're going to solve a particular LP and then decompose our solution and this is possible because ..." and then go into the details. Another case in point is how the authors often write things like "x can be written as $x = \sum \theta_i \dots$ where $\sum \theta_i = g$" but never mention the words "convex combination" even though that's essentially what happens (expressing x as a convex combination of the g-dilation of P). So while formally correct, the meaning is being somewhat obscured. Another level of this quality issue is on the language side, with missing articles etc.

Detailed Comments

I'm attaching a file with line numbers. I strongly recommend for any future submission using the lineno package, because it makes my job as a reviewer much easier. I will refer to pairs (page number, line number) below.

- (2,10) "we have formulated" – better to say "instances ... are known" and give a reference, e.g. to MIPLIB; or if you have some particularly hard instances in mind that YOU have formulated, you can refer to them, but this should be a fairly widely known fact.
- (2,14) "the" -> "these"?
- (2,26) "from the problem" – what problem? Maybe "using special problem structure together with the result of an LP relaxation to obtain an integer feasible solution whose value is at most C times the value of the LP optimum" or something like that.
- (3,7) you say that a separation oracle should return a most violated constraint, but I don't think this is the standard definition of e.g. Grötschel-Lovász-Schrijver – any violated constraint suffices to get polynomial convergence.
- (3,11) "it could not before" -> "they could not before"
- (3,23) "ResearcherS"
- (4,26) "We cannot hope to find..." – you've said this already.
- (5,8) "express A vector"
- (5,10) Please define $[0,1]$ which appears in $[0,1]^k$. It seems you mean the closed interval between 0 and 1, but it is very commonly also used to be the set $\{0,1\}$, so clarification is in order.
- (5,16) "of A blocking type"
- (5,23) "assuming" –> "under"

- (5,30) "$S(I) \in \{0,1\}^n$ should be $\subseteq$ instead of $\in$
- (6,1) delete "algorithm"
- (6,3) "it would be optimal" - what is "it"?
- (6,3) "we can only guarantee a factor of" I think it would be clearer to say "we can only guarantee $C \leq g(I)^{\cdots}$"
- (6,7) how does the min in $\min(Cx^*, \{0,1\}^n)$ work? Coordinate-wise, or?
- (6,27) "It's" –> "Its"
- (7,5) what is χ^{F_i}? I suppose it is the characteristic vector of F_i but you should state this clearly. Also, what is E_x? is it the number of edges of x, that is, $|\text{supp}(x)|$? Again please state clearly.
- (7,13) "A subset of U of V" – delete first "of"
- (7,18) define or comment on UG-hard
- (7,19) "feasbility" –> "feasibility"
- (7,20) "3-5x slower" use "$\times$" (\times) instead of "x"
- (7,29) Large space before "Note" - something fishy is going on here.
- (9,14) add your experimental results about VC here also
- (10,7) again, large space
- (10,14) it seems to me that the definition of the coordinates of $x^{(\ell+1)}$ other than $\ell + 1$ should appear in the algorithm too, not just here.
- (11,14-25) I'm getting tripped up at line 17. I understand the case for coordinates which have $x_j = 0$. But where do you deal with those that do not? I suppose I'm just missing something obvious, but so could any other reader. Please clarify your proof here.
- (12,10) "maintainS"
- (13,28) "Growing and Pruning THE FDT tree"
- (14,12) "It is easy to check" <– you do this quite a bit. Please go through your paper and pay attention to all the "easy to check", "obvious", "clearly" etc. If these things really are easy, then just state them and don't waste space. However, if some observation is necessary, think about the clearest and most concise way to impart this observation to the reader, and then do that instead of telling the reader that it should be "easy".
- (14,12) "THE set L'"
- (14,25) "From THE leaves"
- (16,11-17) You should discuss why the intuitive approach of binarizing an LP (just split each variable into two binary variables and copy the respective column of the constraint matrix) does not lead to the same or better result. Or maybe it does?
- (19,14) "1Mb cache" – I looked up this CPU and it has 1MB of L2 cache. Note that 1Mb $\neq$ 1MB, and "cache" is quite an ambiguous term (there are several types of CPU cache). Moreover, the amount of cache is a property of the CPU and is usually not stated, so I would just not comment on it. If you must, give a correct capacity and indicate the level of the cache.
- (19,22-26) Is there no good benchmark set for TAP? If not, please mention that as the reason why you study random instances. (It is not good practise in experimental algorithmics to study random instances unless necessary.)

- (20,22) One idea to consider and comment on: did you warm-start each computation of FDT? It seems to me that you could reuse the model that FDT solves in each iteration – basically, you only change the objective and fix the variables which corresponds to changing lower/upper bounds, so you should be able to save time on building / destroying the model. I'm not sure how much the pruning model can be reused. Either way, please comment on this idea. The idea was successful elsewhere, e.g. https://arxiv.org/abs/1911.08636
- (21,10) Christofides' algorithm is no longer the fastest https://arxiv.org/abs/2007.01409
- (21,10-11) "In the classic graph-based..." <– this is a weird sentence; you should say that you will re-cap the algorithm and then you can describe it. Right now it comes out of nowhere.