Supplement and Correction

Ziyang Men

June 28, 2022

This document aims to add some explanation for the author's thesis report and correct discovered mistakes. The bold font indicates the section where the modification/explanation focus and the italic font illustrates the specific modified position within the section.

3.1.2 Failed variable

Second paragraph, one example of the optimization method is assuming three variables

$$x_1 \implies x_2 \implies x_3 \text{ not imply } \perp \tag{1}$$

this is equivalent to saying x_1 is not a failed variable meanwhile both x_2 and x_3 will be propagated to true when $x_1 := 1$, in this case, they can be skipped in this round.

3.2.1 Bounded variable elimination

First paragraph, one example for the BVE is assuming four clauses:

$$(\ell_1 \vee \ell), \quad (\ell_2 \vee \ell), \tag{2}$$
$$(\bar{\ell} \vee \ell_3), \quad (\bar{\ell} \vee \ell_4) \tag{3}$$

perform all resolution over ℓ and remove clauses contain ℓ:

$$(\ell_1, \ell_3), \quad (\ell_1, \ell_4) \tag{4}$$
$$(\ell_2, \ell_3), \quad (\ell_2, \ell_4) \tag{5}$$
$$(\bar{\ell}, \ell_3), \quad (\bar{\ell}, \ell_4) \tag{6}$$

Since the number of clauses in equation 2 and 3 is $O(N)$, where N is the number of literals in the formula, the maximal net clauses after performing BVE is:

$$O(N^2 - N) = O(N^2) \tag{7}$$

3.2.2 Techniques based on implication graphs

- *Second paragraph*, by construction of the graph, all literals in same SCC take value true at the same time; meanwhile they must be falsified at the same time as well, e.g., in Figure 3.2, assume $x_2 := 0$ but $x_3 := 1$, by transition $x_4 := 1 \implies x_2 := 1$, contradiction.

- *Last paragraph*, set the objective coefficient $w_y := \sum_{i \in S, \sigma} w_i^\sigma$ will preserve the optimality. This is because all literals in same SCC are equivalent, meaning that we can substitute all literals to y, e.g.,: assume x_1, x_2, x_3 in same SCC, then:

$$
\begin{aligned}
& \min. \quad x_1 + 2x_2 + 3x_3 && (8) \\
\implies \quad & \min. \quad x_1 + 2x_1 + 3x_1 && (9) \\
\implies \quad & \min. \quad y + 2y + 3y && (10)
\end{aligned}
$$

3.3.2 Blocked clause elimination

As a presolving method, the map in substitution ω can be regarded as a value fix or variable substitution.

- Conditions for *equation (3.19)*, since x_W^σ are pure variables, then $x_W^{1-\sigma} \notin F$.

4.1.3 Coefficient strengthening

- *Inequality (4.9)* is sound and can be transformed from equation (4.8) by modifying the RHS of equation (4.8) as:

$$
a_i^\sigma x_i^\sigma + \sum_{j,\sigma} a_j^\sigma x_j^\sigma \leq (A - \sum_{j,\sigma} a_j^\sigma) x_i^\sigma + \sum_{j,\sigma} a_j^\sigma \tag{11}
$$

  The modification is valid since when $x_i^\sigma := 0$ the RHS reads $\sum_{j,\sigma} a_j^\sigma$ which is smaller than A by definition of d; when $x_i^\sigma := 1$ the RHS remains the same as in equation (4.8). Simplify above transformed inequality and we are done.

- Constraint (4.9) is valid because that everything remain same when $x_i^\sigma := 1$; when $x_i^\sigma := 0$, we have $\sum_{j,\sigma} a_j^\sigma x_j^\sigma \leq \sum_{j,\sigma} a_j^\sigma$ which is trivially satisfied, showing this constraint preserves same integer solution, meanwhile $\sum_{j,\sigma} a_j^\sigma x_j^\sigma \leq A - d$, which indicates the modified constraint dominates the original one.

4.1.4 Chvatal-Gomory strengthening of inequalities

- Since a_i^σ needed to be positive as defined in [Ach+20], it is convenient to restrict *constraint (4.10)* as a normalized constraint.

- *Last paragraph*, since $s \in \mathbb{R}$ while the divisor in generalized division rule is a natural number, thence multiply s on both sides does not equal to apply the generalized division rule.
- The constraint (4.13) dominates (4.10) since:
  1. Every satisfying assignment for (4.13) satisfies (4.10) and vice versa, meaning that they preserve the same integer solutions.

     Take the positive direction as an example, let $x_{i^*}^{\sigma}$ denotes those variables assigned to true in a satisfying assignment ρ, assume ρ satisfies (4.13) but falsifies (4.10), which is:

     $$\sum_{i^*,\sigma} \lceil a_{i^*}^{\sigma} \cdot s \rceil \geq \lceil A \cdot s \rceil \tag{12}$$

     $$\sum_{i^*,\sigma} a_{i^*}^{\sigma} < A \tag{13}$$

     then by inequality 13 and $a_{i^*}^{\sigma}, A \in \mathbb{Z}$ we have:

     $$\sum_{i^*,\sigma} a_{i^*}^{\sigma} < A \tag{14}$$

     $$\implies \quad \sum_{i^*,\sigma} a_{i^*}^{\sigma} \cdot s < A \cdot s \tag{15}$$

     $$\implies \quad \sum_{i^*,\sigma} \lceil a_{i^*}^{\sigma} \cdot s \rceil < \lceil A \cdot s \rceil \tag{16}$$

     contradiction.
  2. The LP relaxation becomes tighter, which is ensured by conditions (4.11) and (4.12).

     Assume constraints are in normalized form, constraints (4.10) and (4.13) can be read as

     $$\sum_{i,\sigma} (\frac{a_i^{\sigma}}{A}) x_i^{\sigma} \geq 1 \tag{17}$$

     $$\sum_{i,\sigma} (\frac{\lceil a_i^{\sigma} \cdot s \rceil}{\lceil A \cdot s \rceil}) x_i^{\sigma} \geq 1 \tag{18}$$

     two conditions can be read as:

     $$\frac{\lceil a_i^{\sigma} \cdot s \rceil}{\lceil A \cdot s \rceil} \leq \frac{a_i^{\sigma}}{A}, \quad \forall x_i^{\sigma} \in C \tag{19}$$

$$\frac{\lceil a_j^\sigma \cdot s \rceil}{\lceil A \cdot s \rceil} < \frac{a_j^\sigma}{A}, \quad \exists x_j^\sigma \in C \tag{20}$$

Clearly, the modified constraints cover smaller feasible region.

4.1.7 Doubleton equation substitution

- *Bound change (4.24)* should be rounded up and down as:

$$\lceil \frac{A - a_i^\sigma}{a_j^\sigma} \rceil \leq x_j^\sigma \leq \lfloor \frac{A}{a_j^\sigma} \rfloor \tag{21}$$

- Given a doubleton equation, we can test 4 solution patterns directly in constant time, if none of them are applied, the constraint is unsatisfiable; if exactly one pattern is applied, then we can make a variable fix; if multiple patterns are applied, such as:

$$x_1 + x_2 = 1 \tag{22}$$

  we can make substitution $x_1 = 1 - x_2$.

4.1.8 Simplify inequalities

This subsection has been *rewritten* as follows.

This approach aggressively removes the "unnecessary" variables in a constraint and derives other possible reasoning constraints [Bes+21]. Concretely, given a reverse general constraint $C \ni x_S^\sigma$ in form:

$$C \doteq \quad \sum_{i,\sigma} a_i^\sigma x_i^\sigma \leq A \tag{23}$$

where $a_i^\sigma > 0, i \in S \subseteq N, \sigma \in \{0, 1\}$. Indices set S is an ordered set sorted by the decreasing order of absolute value of a_i^σ. Denote a non-overlap partition $L \cup R = S, L \cap R = \emptyset$ and GCD $d = gcd(a_l^\sigma), l \in L$. Then variables $x_r^\sigma, r \in R$ is redundant w.r.t C if:

$$\sum_{r \in R, \sigma} a_r^\sigma \leq A - \lfloor \frac{A}{d} \rfloor \cdot d \tag{24}$$

the constraint C can then be simplified to:

$$\sum_{l \in L, \sigma} a_l^\sigma x_l^\sigma \leq \lfloor \frac{A}{d} \rfloor \cdot d \tag{25}$$

Let $X_i^\sigma, i \in L \cup R$ be solution for C, the simplified constraint 25 is valid as follows:

1. Completeness.

$$\sum_{l \in L, \sigma} a_l^\sigma X_l^\sigma \leq \sum_{l \in L, \sigma} a_l^\sigma X_l^\sigma + \sum_{r \in R, \sigma} a_r^\sigma X_r^\sigma \leq \lfloor \frac{A}{d} \rfloor \cdot d \tag{26}$$

2. Soundness.

   By condition 24 we have:

$$\sum_{r \in R, \sigma} a_r^\sigma X_r^\sigma \leq \sum_{r \in R, \sigma} a_r^\sigma \leq A - \lfloor \frac{A}{d} \rfloor \cdot d \tag{27}$$

   adding inequality 27 and 25 we have:

$$\sum_{l \in L, \sigma} a_l^\sigma X_l^\sigma + \sum_{r \in R, \sigma} a_r^\sigma X_r^\sigma \leq \lfloor \frac{A}{d} \rfloor \cdot d \leq A \tag{28}$$

If it is not possible to find such partition, then we can simplify the RHS of C as:

$$\sum_{i, \sigma} a_i^\sigma x_i^\sigma \leq \lfloor \frac{A}{d} \rfloor \cdot d \tag{29}$$

where $d = gcd(a_i^\sigma)$.

For example,

$$\begin{aligned} & 15x_1 + 15x_2 + 7x_3 + 3x_4 + x_5 \leq 26 \\ \iff \quad & 15x_1 + 15x_2 \leq 26 \\ \iff \quad & x_1 + x_2 \leq 1 \end{aligned}$$

4.2.1 Dual fixing and bound strengthening

The value fix in this section is based on greedy strategy and we need to consider both the variable and its negation or just pure variables, in this case, when we tighten bounds using dual arguments shown in the *third paragraph*:

- Variable x_i^σ can be fixed to false if this would cause all constraints in M^- redundant meanwhile fixing $x_i^{1-\sigma}$ to true causes all constraints in M^+ become redundant. Both variables are fixed at the same time.

4.3.1 Parallel and nearly parallel rows

- *Constraint (4.46)* should be:

$$\lceil sB \rceil \leq \sum_{i,\sigma} a_i^\sigma x_i^\sigma \leq A \tag{30}$$

  to ensure the modified LHS is an integer.

- The reason that we can discard constraints D in (4.41), (4.43) and (4.45) is that every satisfying assignment for C must satisfy D (within the corresponding conditions). For example, in case 2, when $s > 0$ and $A \geq sB$, we have:

$$\sum_{i,\sigma} s b_i^\sigma x_i^\sigma = \sum_{i,\sigma} a_i^\sigma x_i^\sigma = A \geq sB \tag{31}$$

- For *nearly parallel rows*, the multiplier s should be a rational number, $s \in \mathbb{Q}$, otherwise, in case 1 the substitute variable $x_j^\sigma := t x_i^\sigma + d$ will introduce non-integers.

  Still, any non-trivial bound strengthen represents a value fix here.

4.3.2 Non-zero cancellation

- Modified *constraint (4.62)* requires maintaining Pseudo-Boolean as well; in this case, the multiplier s needs to be a rational number, $s \in \mathbb{Q}$, at least. By multiplying the nominator of s on both sides of constraint (4.62) we can maintain the coefficients and degree in D' as integers.

  Another method to maintain Pseudo-Boolean is to restrict s be an integer, but we may find less s for the derivation.

- For *constraint (4.64)* we need to perform the same procedure shown above.

4.3.4 Clique merging

- *First paragraph.* A conflict graph contains an edge if and only if two variables cannot both take value 1 (not only setting one true will imply the other to false).

- *Second paragraph.* Setting arbitrary $x_i^\sigma := 1$ will force all $x_j^{1-\sigma} := 1$, which suggests there are edges from every x_i^σ to all $x_j^{1-\sigma}$ and these edges form a clique.

- The clique merging process is to extend a given set packing constraint by additional variables using the conflict graph, which is equivalent to searching for a larger clique that subsumes the initial one; then we discard the constraints which are dominated by the added one.

- The added constraint is valid since any feasible MIP solution must correspond to a stable set in the conflict graph. Thus any valid inequality for the stable set polytope on the conflict graph is also valid for the MIP [Ach+20].

4.4.1 Fix redundant penalty variables

- *First paragraph.* Intuitively, those penalty variables with large absolute objective coefficients and small constraints coefficients should be used before others. This method is also based on the greedy strategy but the target variables are singleton variables thence we don't have to consider their negation.

- The *equation (4.75)* should be:

$$\sum_{u \in [k], \sigma} a_u^\sigma \geq A_q \tag{32}$$

  meaning that setting the first k penalty variables to true is sufficient to satisfy the constraint; those more "expensive" penalty variables $x_j^\sigma, j \in U \setminus [k]$ are not needed to set to true; besides, setting arbitrary x_v^σ to true will only reduce the number of true variables needed in x_u^σ. Therefore the adaption is valid.

References

[Ach+20] Tobias Achterberg et al. "Presolve reductions in mixed integer programming". In: *INFORMS Journal on Computing* 32.2 (2020), pp. 473–506.

[Bes+21] Ksenia Bestuzheva et al. "The SCIP Optimization Suite 8.0". In: *arXiv preprint arXiv:2112.08872* (2021).