Pseudo Code for Repair Algorithm

Arthur Feng

March 24, 2019

1 Introduction

Definition 1.1. (Mixed Integer Programs)
In this report, we denote the $Mixed\,Integer\,program$ (MIP) in the follow form

$$\mathcal{Z}_{MIP} = \min\{\, c^T x : Ax \leq b, x \in \mathbb{Z}^k \,\mathrm{x}\, \mathbb{R}^l\}$$

where $m, n \in \mathbb{N}$, and $n = k + l$, A is a matrix for $A \in \mathbb{R}^{m \,\mathrm{x}\, n}$, and the vector $b \in \mathbb{R}^m$, the vector $c \in \mathbb{R}^n$.

S is a set of $feasible\,solution$ if S satisfy all the constraints in the problem. If $s \in S$, s is called a feasible solution to the problem $\mathcal{Z}_{MIP}$. The MIP is called infeasible when S is empty, other wise, the MIP is called feasible. The feasible solution s^* is called optimal when $s^* \in S$ and $c^T x_{s^*} \leq c^T x_s$ for $\forall s \in S$.

Definition 1.2. (Linear Programs and LP-relaxation)
With the definition of MIP in 1.1, there is a special case of MIP, which is called $Linear\,Problem$ (LP)

$$\mathcal{Z}_{LP} = \min\{\, c^T x : Ax \leq b, x \in \mathbb{R}^l\}$$

A LP can also be obtained by removing all integrity constraints: $x_i \in x$ where $i \in n \setminus k$, this is called LP-*relaxation*. LP-*relaxation* is the foundation of LP-based branch-and-bound technology. As the searching space is increase by removing integrity restrictions, the optimal solution in MIP could not better than LP-*relaxation*, which is $s^*_{MIP} \geq s^*_{LP}$. This means the optimal solution found in LP could provide a lower or prime bound for MIP.

Definition 1.3. (Binary Program)
With the definition of MIP in 1.1, there is a special case of MIP, which is called $Linear\,Problem$ (LP)

$$\mathcal{Z}_{LP} = \min\{\, c^T x : Ax \leq b, x \in \mathbb{R}^l\}$$

A LP can also be obtained by removing all integrity constraints: $x_i \in x$ where $i \in n \setminus k$, this is called LP-*relaxation*. LP-*relaxation* is the foundation of LP-based branch-and-bound technology. As the searching space is increase by removing integrity restrictions, the optimal solution in MIP could not better than LP-*relaxation*, which is $s^*_{MIP} \geq s^*_{LP}$. This means the optimal solution found in LP could provide a lower or prime bound for MIP.

2 Input

- A problem = $min\{c^T x \,|\, Ax \geq b,\, x_i \in \{0,1\},\, \forall i \in \mathcal{I},\, \mathcal{I} \subseteq N = \{1,2,\ldots,n\}\}$ with n binary variables x, objective function $c^T x$ (min), constraint set C^0 with an optimal solution x^0.
- A problem P^1 with n binary variables x, objective function cx (min), constraint set C^1, such that $C^0 \subsetneq C^1$.

3 Output

- An optimal solution x^1 to P^1.

4 General Idea

We propose to solve P^1 by reusing the optimal solution x^0 to P^0. In order to achieve this, we define a new problem Q with constraint set C^1 and objective function

$$\min cx + \alpha|x - x^0|,$$

where $|x - x^0| = \sum_{i=0}^{n-1} |x_i - x_i^0|$, and α is a *penalty* term for deviating from the input solution x^0. This would tentatively help the search for a good solution to P^1. However, unless x^0 is feasible for P^1, an optimal solution to Q will in general not be optimal for P^1.

To remedy thiss problem, we will instead solve a sequence of problems $Q^0, Q^1, \ldots$, where the penalty factor α will gradually decrease until it reaches 0, say at iteration k, in which case $Q^k = P^1$. This sequence of problems can be efficiently solved using a technique called *reoptimisation*, which is implemented in the MIP solver SCIP.

5 Pseudo Code

Algorithm 1: Restriction

Data: $G = (X, U)$ such that G^{tc} is an order.
Result: $G' = (X, V)$ with $V \subseteq U$ such that G'^{tc} is an interval order.

```
begin
    V ⟵ U
    S ⟵ ∅
    for x ∈ X do
        NbSuccInS(x) ⟵ 0
        NbPredInMin(x) ⟵ 0
        NbPredNotInMin(x) ⟵ |ImPred(x)|
    end
    for x ∈ X do
        if NbPredInMin(x) = 0 and NbPredNotInMin(x) = 0 then
            AppendToMin(x)
        end
    end
    while S ≠ ∅ do
        remove x from the list of T of maximal index
        while |S ∩ ImSucc(x)| ≠ |S| do
            for y ∈ S − ImSucc(x) do
                { remove from V all the arcs zy : }
                for z ∈ ImPred(y) ∩ Min do
                    remove the arc zy from V
                    NbSuccInS(z) ⟵ NbSuccInS(z) − 1
                    move z in T to the list preceding its present list
                    {i.e. If z ∈ T[k], move z from T[k] to T[k − 1]}
                end
                NbPredInMin(y) ⟵ 0
                NbPredNotInMin(y) ⟵ 0
                S ⟵ S − {y}
                AppendToMin(y)
            end
        end
        RemoveFromMin(x)
    end
end
```

Algorithm 2: How to write algorithms

```
Result: Write here the result
initialization;
while While condition do
    instructions;
    if condition then
        instructions1;
        instructions2;
    else
        instructions3;
    end
end
```

6 MIP

6.1 Mixed Integer Programming

Definition 6.1. (Mixed Integer Programming)
Let $m, n \in \mathbb{N}$, The given matrix $A \in \mathbb{R}^{m \times n}$, vectors $b \in \mathbb{R}^m$, and the vector $c \in \mathbb{R}^n$, and a set $\mathcal{I} \subseteq N = \{1, \ldots, n\}$. the problem

$$\begin{aligned} \text{(MIP)} \quad & c^* = \min c^T x \\ & s.t. \quad Ax \geq b \\ & \qquad x_i \in \mathbb{Z}_{\geq 0} \quad \forall_i \in \mathcal{I} \\ & \qquad x_j \in \mathbb{R}_{\geq 0} \quad \forall_i \in \mathcal{N} \setminus \mathcal{I} \end{aligned}$$

is called *mixedintegerprogram* with the objective function $c^T x$ and constraints $A_i x \geq b_i$ for all $i = 1, \ldots, m$.

A vector $x \in X_{MIP} = \{x \in \mathbb{R}_{\geq} 0^n \mid A_x \geq b,\, x_i \in \mathbb{Z}_{\geq 0} \forall i \in \mathcal{I}\}$ is called $feasiablesolution$ and X_{MIP} the set of feasible solutions. A feasible x^* is called optimal, if x^* satisfies $c^* = c^T x^*$.

Common special cases of MIPs are *linearprograms* (LPs) for $\mathcal{I} = \emptyset$ and *integerprogram* (IPs) for $\mathcal{I} = N$. Additional, an integer variable bounded by 0 and 1 is called *binaryvariable*. let $\mathcal{B} \subset \mathcal{I}$ denote the set of binary variables. An integer program with $\mathcal{B} = N$ is called *binaryprogram* (BP) or *mixedbinaryprogram* (MBP), if $\mathcal{B} = \mathcal{I} \subsetneq N$.

A lower or dual bound on a MIP can be computed by neglecting the intergrality constraints. the so-obtained problem is called the *LP-relaxation* of the MIP.

Definition 6.2. (LP-relaxation)

Given a MIP as introduced in Definition **??**. The LP-*relaxation* is defined as

$$\begin{aligned} \text{(MIP)} \quad & c^* = \min c^T x \\ & s.t. \quad Ax \geq b \\ & \qquad\quad x \in \mathbb{R}^n_{\geq 0} \end{aligned}$$

Analogous to X_{MIP} we can define X_{LP} as the set of feasible solutions of the LP-relaxation. A feasible solution $x^*_{LP} \in X_{LP}$ is called LP-*optimal* if $c^*_{LP} = c^T x^*_{LP}$. In general solving MIPs is NP-hard. One common method for solving MIPs is LP-$based branch-and-bound$. This method splits the problem into smaller subproblems, and procedure is repeated on these subproblems. At any point a global upper or primal bound is given by the best known solution, if existent, and a local lower bound or dual bound is given by the respective LP-relaxation.

6.2 Pseudo-Boolean optimization

In the section 2.4 we will present a special kind of a binary problem, a so-called $pseudo-Boolean problem$. For this purpose we introduce the basic definition of a $pseudo-Boolean problem$ in this section.For more detail we refer to Hammer and Rubin and Boros and Hammer and the references therein.

Let us denote by $\mathbb{B} = \{0, 1\}$ the set of binary values and let $N = \{1, \ldots, n\}$ be an index set, Reflecting to Boros and Hammer we consider functions in n binary variables $x_1, x_2 \ldots, x_n$ and denote the binary vector by $(x_1, x_2, \ldots, x_n in \mathbb{B}^n)$. A function $f : \mathbb{B}^n \to \mathbb{R}$ of the form

$$f(x_1, x_2, \ldots, x_n) = \sum_{S \subseteq N} C_S \prod_{i \in S} x_i$$

where $C_s \in \mathbb{R}$ for each $S \subseteq N$, is called $pseudo - Boolean function$. see e.g., Hannmer et al. The degree of the function $deg(f)$ is given by the seize of the largest set $S \subseteq N$ which$C_S \neq 0$.A pseudo-Boolean function f is called linear, quadratic, cubic etc. if $deg(f) \leq 1, 2, 3$ etc. repectively. Liu and Truszczynski defined a $pseudo - Boolean constraint$ as an integer inequality of the form

$$\sum_{i=1}^{n} a_i x_i \geq b$$

with $a_i, b in \mathbb{Z}$ and $xi \in \mathbb{B}$ for all $i in [n]$. A binary problem defined by a pseudo-Boolean objective function and a set of pseudo-Boolean constraints is called $pseudo - Boolean\, optimization\, problem$.