# Reinforcement Learning for (Mixed) Integer Programming: Smart Feasibility Pump

ICML | 2021
Thirty-eighth International Conference on Machine Learning

Meng Qi*, Mengxin Wang*, Zuo-Jun (Max) Shen
Department of Industrial Engineering and Operations Research, UC Berkeley
* denotes equal contribution

## Key Take-away Messages

- **A RL model for feasible solutions of MIP**: our work is the first attempt to use (deep) RL methods for seeking feasible solutions for a class of general MIPs.
- **The spirit of a successful heuristic:** Inspired by FP, we propose the smart feasibility pump model because it is empowered by deep RL models.
- **A novel CNN for constraint matrix**: We innovatively adopt a convolutional structure for the policy network to capture the structure of constraint matrix of MIPs.
- **Empirical evaluation:** The results demonstrate the significant advantages of the SFP models compared to the original FP and the representation power of CNN.

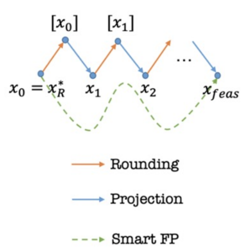
## Preliminary knowledge and background

**Mixed Integer Programming (MIP)**: We aim at finding a feasible solution of the following MIP:

$$\min \quad c^T x \qquad (1a)$$
$$s.t. \quad Ax \leq b \qquad (1b)$$
$$x_i \in \mathbb{Z}, \forall i \in S \qquad (1c)$$

Solving an MIP is computationally challenging (NP-hard in general).
Finding a feasible solution is a critical initial step for various MIP heuristics.

**Feasibility Pump (FP)**: The basic idea of the FP algorithm is to iteratively find and round a continuous relaxation solution for the MIP. The FP algorithm starts with the rounded optimal continuous relaxation solution of the MIP and then searches for the nearest points in the relaxed feasible region. It continues perturbing and rounding the new point found at each step until a feasible solution is discovered or the limit of maximum number of steps is reached. Despite being a powerful heuristic, it requires solving an optimization problem within each iteration, which becomes especially inefficient when the problem size increases or extends to nonlinear constraint cases.




## Smart Feasibility Pump: a RL Formulation

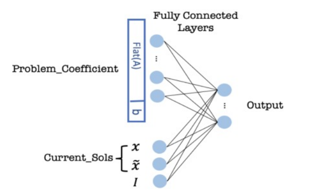
**The RL formulation**

- State Space:
  - SFP-MLP $\quad s_t = (\text{Flat}(A), b, x_t, \tilde{x}_t, I)$
  - SFP-CNN $\quad s_t = ([A, b], x_t, \tilde{x}_0, I)$
- Action Space and State Transition:
  - Action is the movement of current solution $\quad a_t \in \mathcal{A} = \mathbb{R}^n$
  - Transaction of current solution: $\quad x_{t+1} = [x_t + a_t]$
  - State transaction: $\quad s_{t+1} = (\text{Flat}(A), b, \tilde{x}_{t+1}, x_{t+1}, I)$
- Reward: violation of constraints $\quad r_t = -\|Ax_t - b\|$

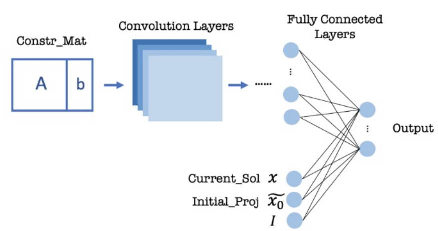
**Policy Learning**

- We use Actor-Critic with PPo for policy gradient.
- We consider two policy network structure:

**SFP-MLP**: the policy network is an MLP. The projection of current solution $\tilde{x}_t$ in the state vector improves the learning ability.




**SFP-CNN**: the policy network is a CNN. The CNN is so powerful to capture constraint structure and let us get rid of the computationally inefficient component $\tilde{x}_t$



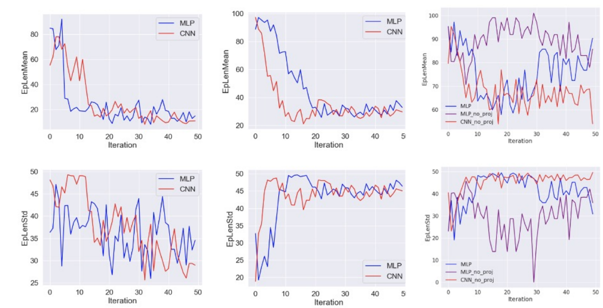

## Evaluation Metrics

- **EpLenMean & EplenStd:** the mean and the standard deviation of the number of steps the agent reach a feasible solution or the maximum number of steps (100)
- A model with higher EpLenMean and a lower EpLenStd means it steadily produce worse solutions.

## Experiments

| | $n=5, m=6$ | | | $n=7, m=9$ | | | $n=9, m=18$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | FP | MLP | CNN | FP | MLP | CNN | FP | MLP | CNN |
| EpLenMean | 43.2 | 15.0 | 11.1 | 65.5 | 32.8 | 28.2 | 90.0 | 90.3 | 54.0 |
| EpLenStd | 48.6 | 34.6 | 29.0 | 46.8 | 46.4 | 44.2 | 29.7 | 30.9 | 49.6 |
| EpLenMax | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| 90 Quant | 100.0 | 100.0 | 29.5 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| 10 Quant | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 61.2 | 51.5 | 1.0 |

Table 1: Comparison of SFP and FP




(a) 5-dim, 6 constraints (b) 7-dim, 9 constraints (c) 9-dim, 18 constraints

Figure 1: Comparison of SFP-MLP and SFP-CNN

- SFP agent finds a feasible solution to IP/MIPs faster than the FP algorithm (Table 1).
- SFP-MLP and SFP-CNN are comparable when the problem size is small (Figure 1(a))
- SFP-CNN outperforms SFP-MLP in the sense that it converges faster to a lower EpLenMean with comparable EpLenStd when the problem size becomes larger (Figures 1(b,c)).
- The performance of SFP-MLP is largely dampened without the projection information, while the performance of SFP-CNN without projection is better than that of SFP-MLP with projection.
- SFP-CNN can be more computationally efficient than SFP-MLP with larger problem scales. The representational power of the CNN structure captures hidden information in the constraint matrices.