

EFFICIENTLY SOLVING CONSTRAINT OPTIMIZATION PROBLEMS USING LEARNING-BASED TECHNIQUES

Romaissa Ghlib¹, Abdelkader Ouali¹, Jean-Luc Lamotte¹, Samir Loudni²

¹Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ, GREYC UMR 6072, F-14000 Caen, France

²IMT Atlantique, LS2N, UMR CNRS 6004, F-44307 Nantes, France

romaissa.ghlib@unicaen.fr

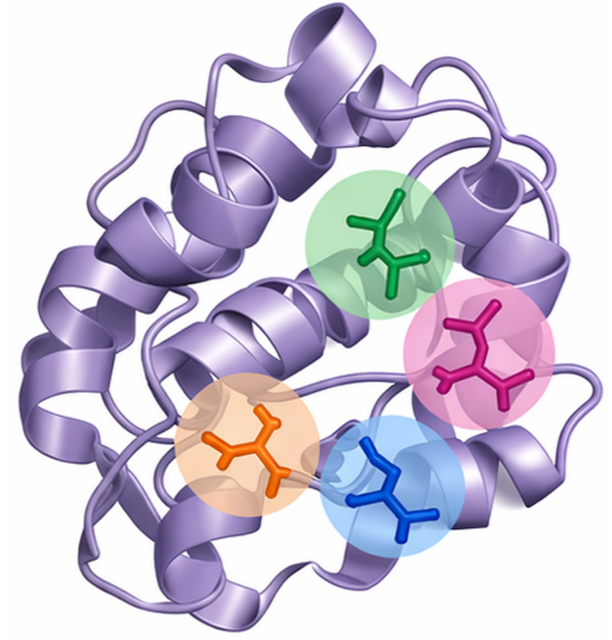


1. Context

Cost Function Networks are a specific type of graphical model used to solve optimization problems in which constraints have associated with a violation costs, where the objective is to find solution that minimizes the final cost.

Computational Protein Design (CPD)

Design an amino-acid sequence that folds into a target 3D structure with low energy.



Modeling CPD as CFN

Design an amino-acid sequence that folds into a target 3D structure with low energy.

- Variables X**
Amino-acide position / rotamer to be selected
- Domain D**
A list of side-chain types and rotamers in position i
- Cost Function w**
Energy term evaluating a subset S of positions (local or interaction)

Cost Function Network (CFNs)

CFN is defined by quadruplet (X, D, W, K) :

- Set of n variables $X = \{x_1, \dots, x_n\}$
- Set of n domains $D = \{D_1, \dots, D_n\}$ (finite domains)
- Set of e cost functions $W = \{w_1, \dots, w_e\}$
- Each w_s has a scope $S \subseteq X$
- $w_s : D^S \rightarrow [0, k]$
- $k > 0$ (penalizes forbidden assignments)

CFN Objective

Find a complete assignment of the variables (rotamers) minimizing the sum of cost functions (the total energy of protein).

$$\min_{x_1 \in D_1, \dots, x_n \in D_n} \sum_{w_s \in W} w_{S_i}(x_{S_i})$$

2. Methods for Solving CFNs

CFNs can be addressed by complementary solving strategies, balancing optimality guarantees, scalability, and structural exploitation.



Exact Methods (complete search)

Goal: Find an **optimal solution** with a **proof of optimality**.

- ▶ **Branch and Bound** with constraint filtering.
- ▶ **DFBB, AOBB** (AND/OR graphs) [Marinescu and Dechter, 2009].
- ▶ **Tree-based methods** using tree decomposition:
 - **BTD** [Terrioux and Jégou, 2003],
 - **BTD-HBFS** [Schiex2006a].
 - **AND/OR search** [Marinescu and Dechter, 2009],
 - **RDS-BTD** [Sanchez et al., 2010].



Metaheuristics (local search)

Goal: Quickly obtain a solution of **good or very good quality**, without optimality guarantee.

- ▶ **VNS** [Hansen and Mladenović, 1997].
- ▶ **VNS/LDS + CP** [Loudni and Boizumault, 2003].
- ▶ **DGVNS** [Fontaine et al., 2013].
- ▶ **CPDGVNS (HM'14), RSDGVNS, RADGVNS (ENDM'15)**.

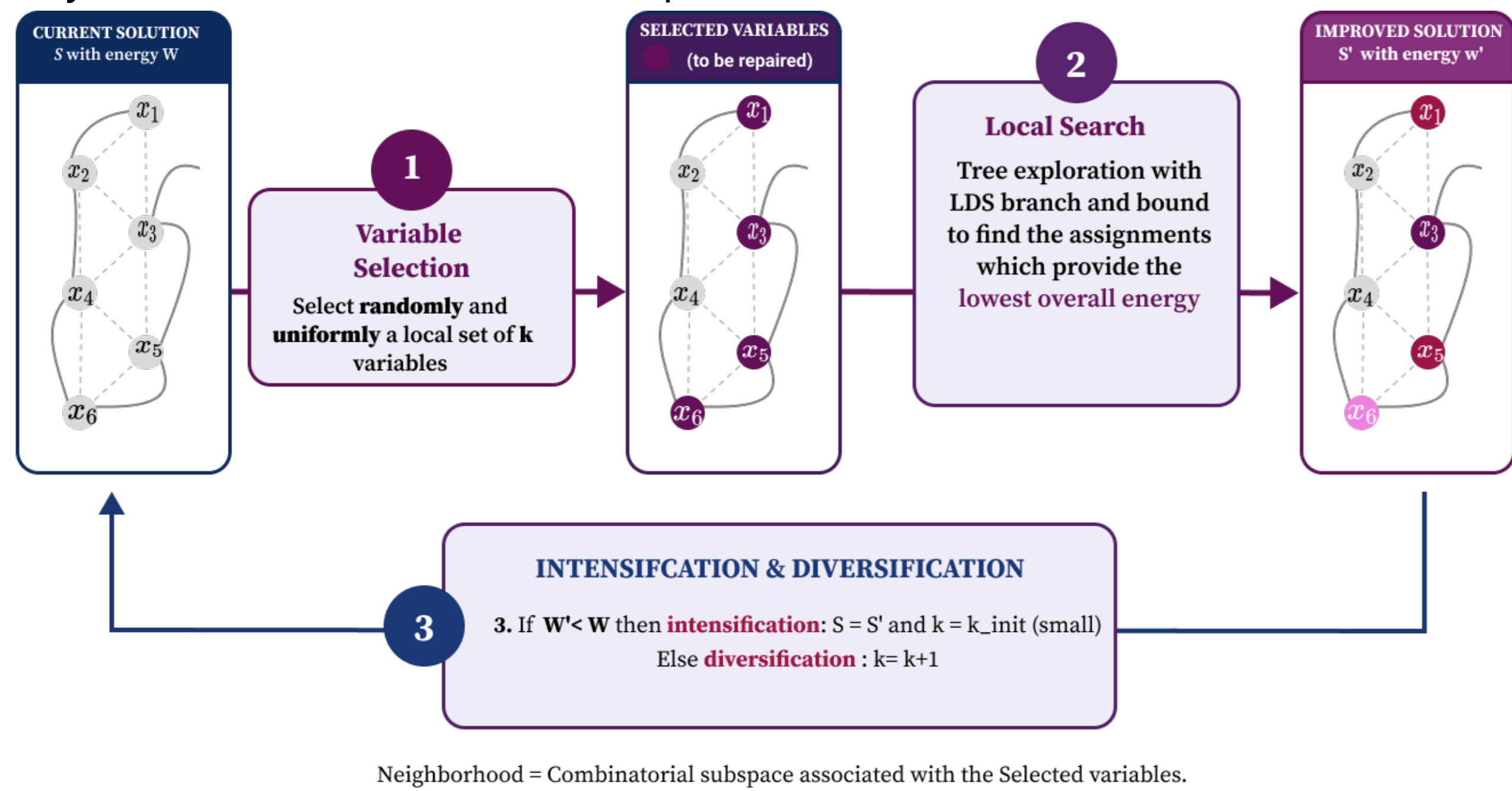


UDGVNS & UPDGVNS Unified, complete and parallel versions of DGVNS.

- [Ouali et al., 2020] (UA'17, AJ'20).

3. VNS/LDS+CP Method

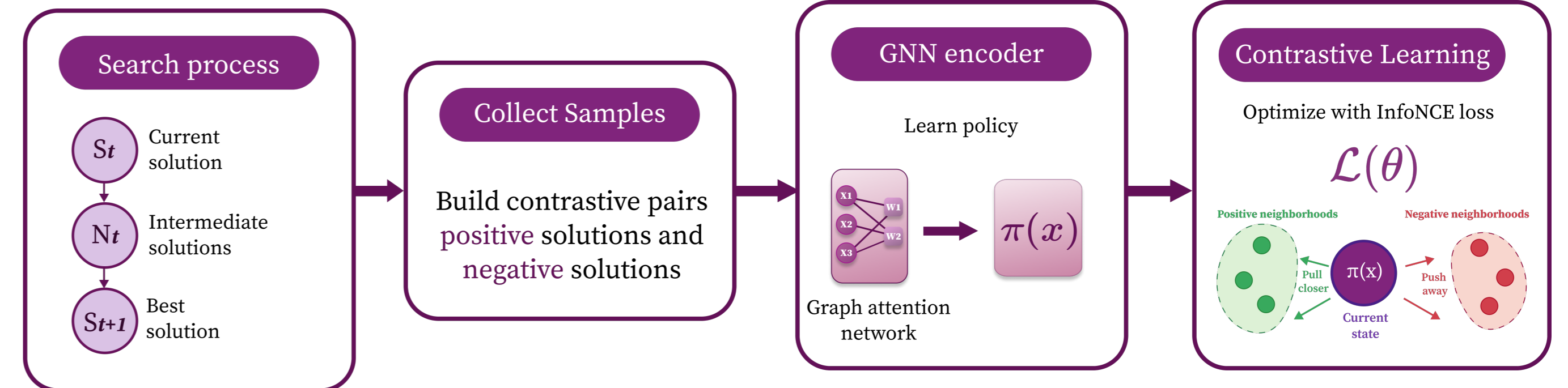
VNS explores the search space by **iteratively** changing the neighborhood size K . Its efficiency relies on the **quality of the variables** selected to be repaired.



- **Limitations :**
 - Random heuristic without exploiting past search experience
 - Insensitive to variable importance, leading to inefficient repairs
 - Limited in guiding intensification, especially on complex instances
- **Motivation :**
 - Learning from the search history using Contrastive Learning with GNNs
 - Learn a strategy to identify and select better neighborhoods.

4. Contrastive Learning Framework

The model learns a policy where improving variables are pulled closer to the current search state, while ineffective variables are pushed away.



Training loss [T. Huang et al, ICML 2023] : InfoNCE encourages similarity with improving variables and dissimilarity with non-improving ones

$$\mathcal{L}(\theta) = \sum_{(s, S_{pos}, S_{neg}) \in \mathcal{D}} \frac{-1}{|S_{pos}|} \sum_{a \in S_{pos}} \log \frac{\exp(a^T \pi(s)/\tau)}{\sum_{a' \in S_{neg} \cup \{a\}} \exp(a'^T \pi(s)/\tau)}$$

5. Proposed method CL-VNS

1. Generate training data by using VNS/LDS+CP on CFN instances

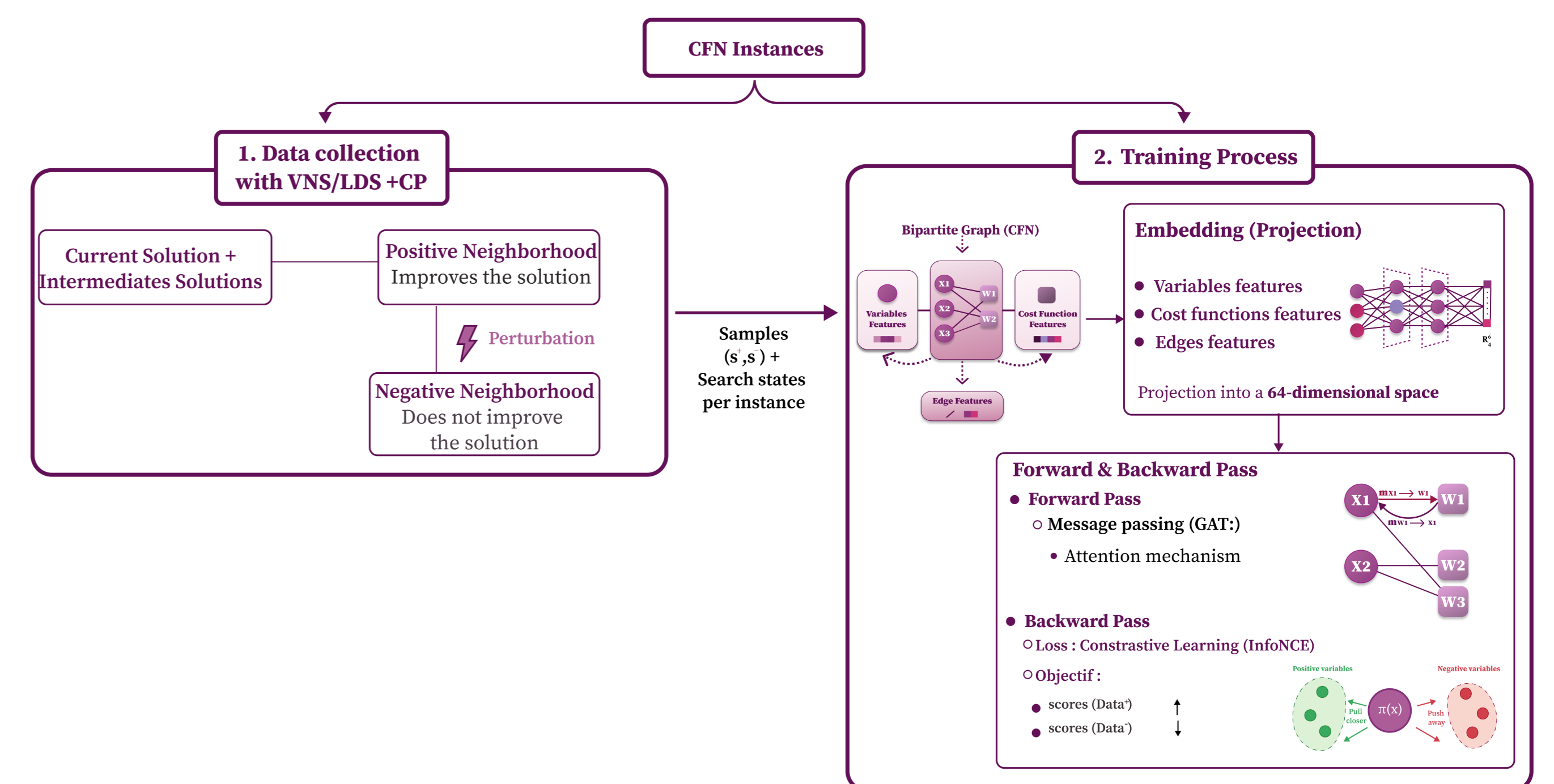
- **Positive solutions** : solutions that approach the best known solution and yield a significant cost gain
- **Negative solutions** : solutions perturbed from the neighborhood of the best solution that yield only a marginal cost gain

2. Train GNN with supervised contrastive learning to distinguish effective from ineffective variables and guide the search.

- Attention mechanism on bipartite graph (GAT)
- Embedding representation using MLP from variable/constraint features.

3. Use the learned policy to predict scores for variables

4. Perform weighted random sampling without replacement to select k variables of the neighborhood



6. Results

Instance	n	d	e	Method	Best Cost	Time (s)	Succ.
5dbl	130	384	1706	VNS/LDS+CP	202393247132	245.56	5/5
				CL-VNS	202393247132	179.22	5/5
1dvo	152	389	2587	VNS/LDS+CP	275328476364	887.676	5/5
				CL-VNS	275328476364	950.132	5/5
2gee	188	397	3715	VNS/LDS+CP	393484253099	6562.323	5/5
				CL-VNS	393484253099	4462.567	5/5
2x8x	235	407	4745	VNS/LDS+CP	508709861679	5184.334	5/5
				CL-VNS	508709861679	5107.984	5/5
1ng2	176	397	3135	VNS/LDS+CP	338460598890	1005.155	5/5
				CL-VNS	338460598890	650.237	5/5

Table 1: Comparison of VNS/LDS+CP and CL-VNS on CPD instances. n : number of variables; d : maximum domain size; e : number of cost functions. Best Cost : lowest objective value found; Time (s) : time to reach best cost; Succ. : number of runs that matched the best known solution.

7. Conclusion

- The anytime behavior of VNS/LDS+CP is adequate to collect positive solutions
- CL-VNS capture the current state of the search process to evaluate the importance of variables for the next iteration.
- **Improvements :**
 - Experimenting on larger and more diverse CFN instances
 - Extend the current CL-VNS approach to UDGVNS in order to exploit tree-based decomposition.

Acknowledgment : This thesis is funded by the ANR project GMLaS (ANR-24-CE23-3429). Project website : <https://hub.imt-atlantique.fr/gmlas/>