Instance-wise algorithm configuration with graph neural networks

Romeo Valentin†, Claudio Ferrari, Jeremy Scheurer, Andisheh Amrollahi, Chris Wendler, Max B. Paulus

ETH zürich Team: MixedInspiringLamePuns Configuration Task: Rank 3 / 15

Too long, didn't read

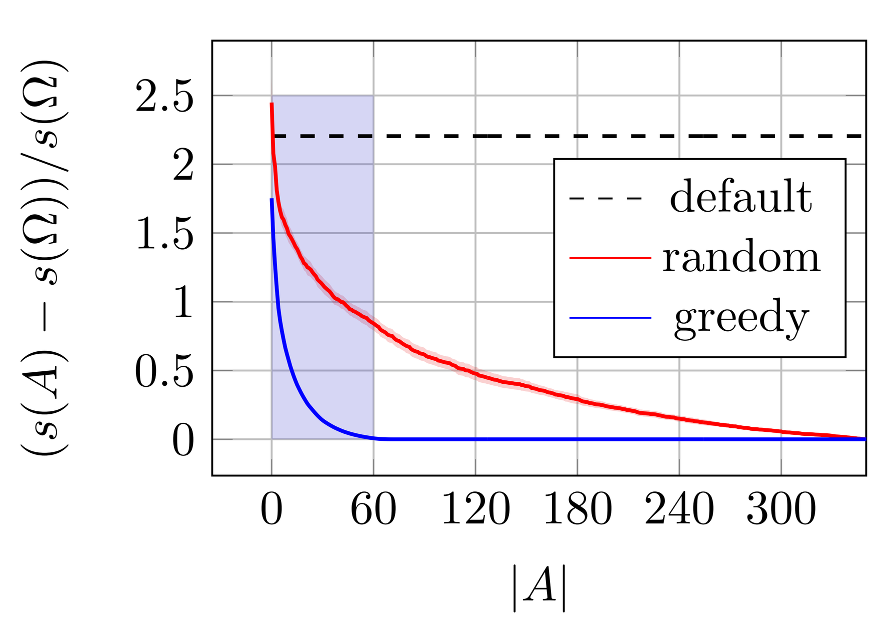
- Pose instance-wise algorithm configuration as a supervised learning problem
- Facilitate data collection over a small subset of configurations using domain knowledge and greedy search
- Use GNNs and per-instance normalization to learn a powerful model for instance-wise algorithm configuration

Configuration Space & Data Collection

- Define a reduced configuration space $\Omega \ni c$ with $|\Omega| \approx 353$ as the Cartesian product of the SCIP emphasis settings for *presolving*, *heuristics*, *separating* and *emphasis*.
- Solve a small number of training instances ($\mathcal{N} \ni i$ with $|\mathcal{N}| = 100$) with each config $c \in \Omega$ to collect the primal-dual integral γ_{ic}.
- Choose a small subset of configs $A \subset \Omega$ to approximately optimize the (submodular) score

$$s(A) = -\sum_{i \in N} \min_{c \in A} \gamma_{ic}$$

s.t. $|A| \leq k$ using the greedy algorithm [1].



- Solve all training instances with each config $c \in A$
- For item placement (above) and load balancing, we find that $s(A) \approx s(\Omega)$ for $|A| = 60$ and $|A| = 40$, respectively.
- Thus, we save more than 80% compute over collecting γ_{ic} exhaustively over entire Ω.

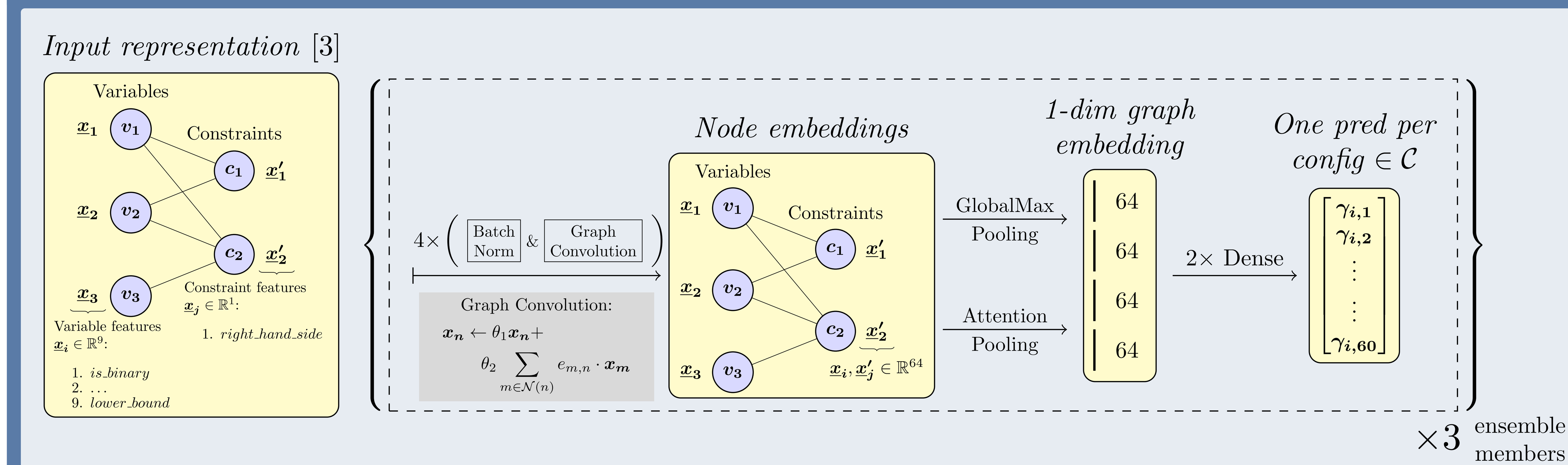
Model architecture

1. *Input:* bipartite graph [2]; *Output:* relative performance for each config.
2. We need a graph neural network operator that supports both (i) *edge weights*, and (ii) a *bipartite* structure → we use `GraphConv` [3].
3. We train a small ensemble using a simple $\mathcal{L}_2$ loss.

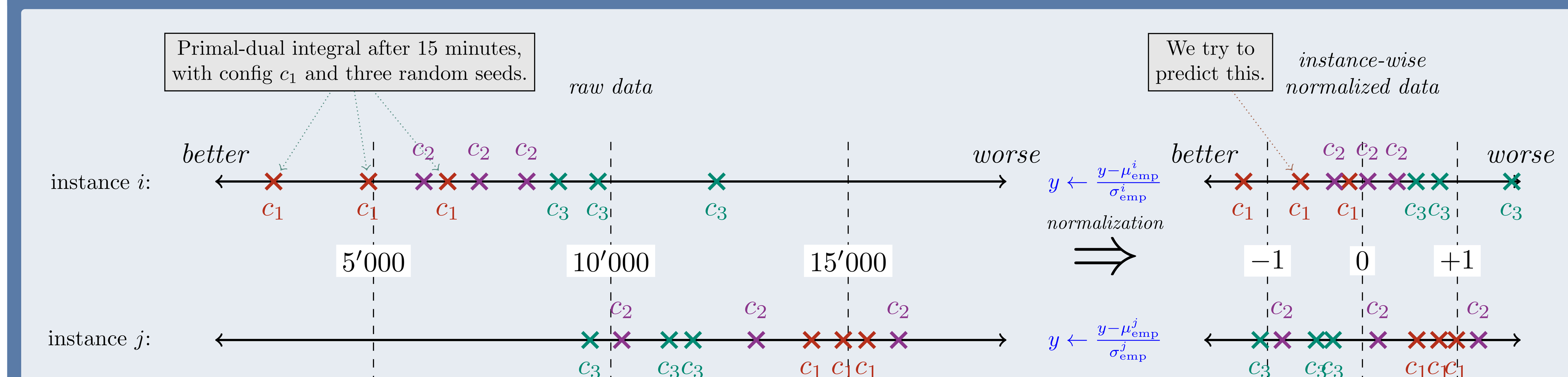
Supervised learning problem

- Label γ_{ic} is the measured primal-dual integral for an instance-config-tuple.
- Goal: Learn how configs compare for an instance.
- Instead of ranking loss, we predict performance on a *relative scale* → leverage instance-wise normalization.

Model architecture



Instance-wise data normalization



Data normalization & rationale

Compute empirical $\mu^i_{\text{emp}}, \sigma^i_{\text{emp}}$ for each instance i, then normalize instance-wise:

$$\gamma_{ic} \leftarrow \frac{\gamma_{ic} - \mu^i}{\sigma^i}.$$

⇒ The model only needs to learn *how a config compares to the other configs, not whether an instance is easy or hard.* → increase signal-to-noise ratio!

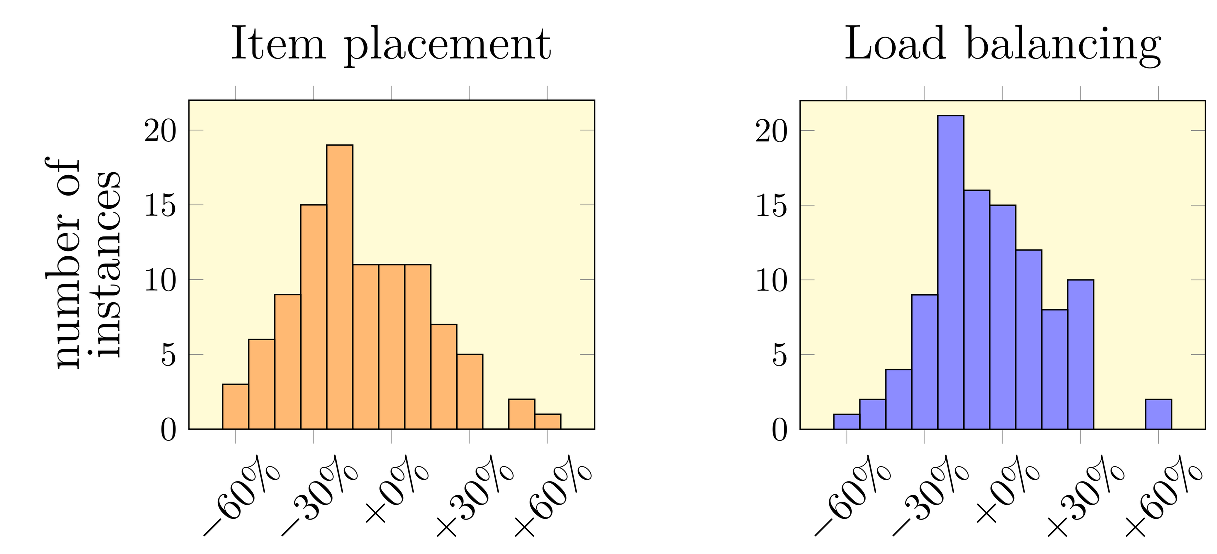
Code & contact information

`github.com/RomeoV/ml4co-competition`

Contact†: `romeo.valentin@math.ethz.ch`

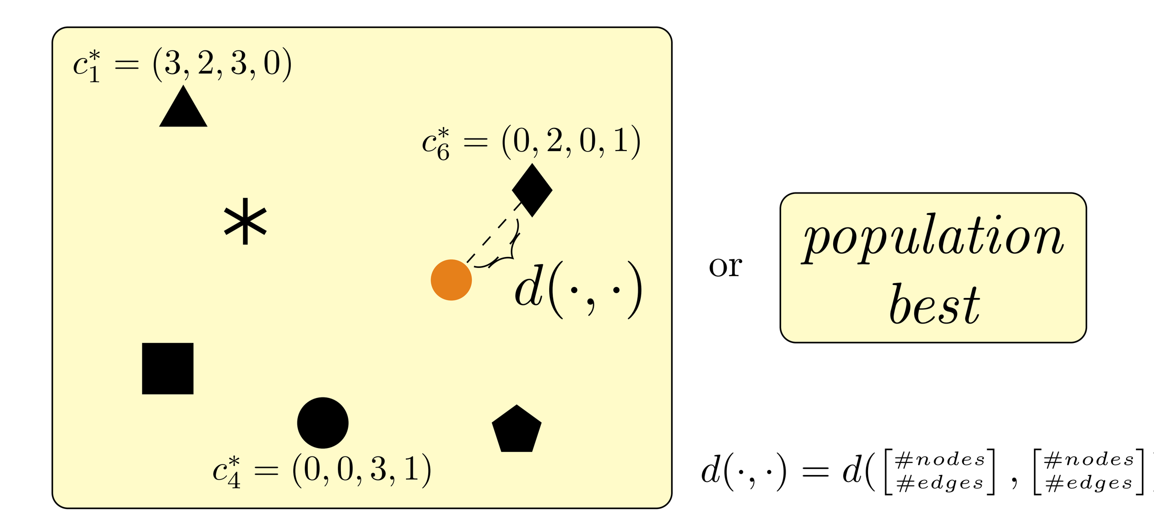
Improvements on c_{default}

Our model predicts configs c_{model} that improve in validation performance on the default c_{default}.



The mean (median) improvement is 12% (16%) for item placement and 5% (6%) for load balancing.

Anonymous Dataset



- Solve all instances $i \in \mathcal{D}$ with each config $c \in \Omega$ to collect the primal-dual integral γ_{ic}.
- Identify six clusters $\{cl_j\}_{j=1}^{6} = \mathcal{D}$ using the instances' number of constraints and variables only.
- Compute $c_j^* = \operatorname{argmin}_{c \in \Omega} \sum_{i \in cl_j} \gamma_{ic}$ for each cluster and for the population $c_0^* = \operatorname{argmin}_{c \in \Omega} \sum_{i \in \mathcal{D}} \gamma_{ic}$.
- For a test instance (▪ above), choose either c_j^* where j indexes the nearest cluster or c_0^*.

References

[1] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[2] M. Gasse, D. Chetelat, N. Ferroni, L. Charlin, and A. Lodi. Exact combinatorial optimization with graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[3] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4602–4609, Jul. 2019.