HORDA

Heuristic Optimizer using Regression-based Decomposition Algorithm

Michal Bouška

Czech Technical University in Prague
bouskmi2@fel.cvut.cz

Motivations

- Solve NP-Hard problems by Neural Networks
- Infer information from data by machine
- Utilize historical data to improve results

Problems

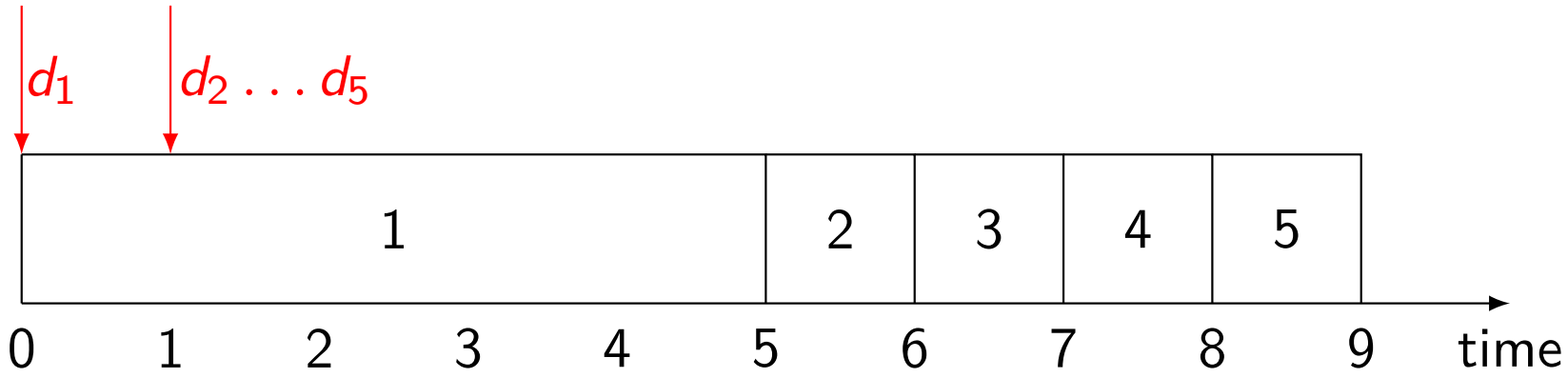
- Existing ML approaches are **inferior to state-of-the-art OR algorithms!**
  - S2V-DQN network on TSP with quality comparable to 2-opt heuristic[1]
  - Pointer network on TSP up to 50 nodes[2]
- Combination of OR and ML
- Design of Neural Network
- Generation of training data
- Training of NN, normalization of data

[1] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Advances in Neural Information Processing Systems*, 2017.

[2] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015.

Our approach: Integration of ML and OR algorithms

- Single machine total tardiness problem
  - Set of jobs $\{j_1, \ldots, j_n\}$
  - Each job j_i has processing time p_i and due date d_i
  - Job sequence $\pi : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$
  - Completion time $C_{\pi(k)} = \sum_{k'=1}^{k} p_{\pi(k')}$
  - Objective function: sum of tardiness
    $T_{\pi(k)} = \max(0, C_{\pi(k)} - d_{\pi(k)})$
  - Weakly NP-hard, admits pseudo polynomial time algorithm



State of the art OR methods

- Optimal
  - Lawler decomposition – up to 100 jobs[3]
  - SDD – up to 500 jobs[4]
  - TTBR – up to 1200 jobs[5]
- Heuristic
  - NBR[6]

---

[3] E. L. Lawler, "A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness," in *Studies in Integer Programming*, 1977.

[4] W. Szwarc, F. Della Croce, and A. Grosso, "Solution of the single machine total tardiness problem," *Journal of Scheduling*, 1999.

[5] M. Garraffa, L. Shang, F. Della Croce, and V. T'kindt, "An exact exponential branch-and-merge algorithm for the single machine total tardiness problem," *Theoretical Computer Science*, 2018.

[6] J. E. Holsenback and R. M. Russell, "A heuristic algorithm for sequencing on one machine to minimize total tardiness," *Journal of the Operational Research Society*, 1992.

Lawler decomposition

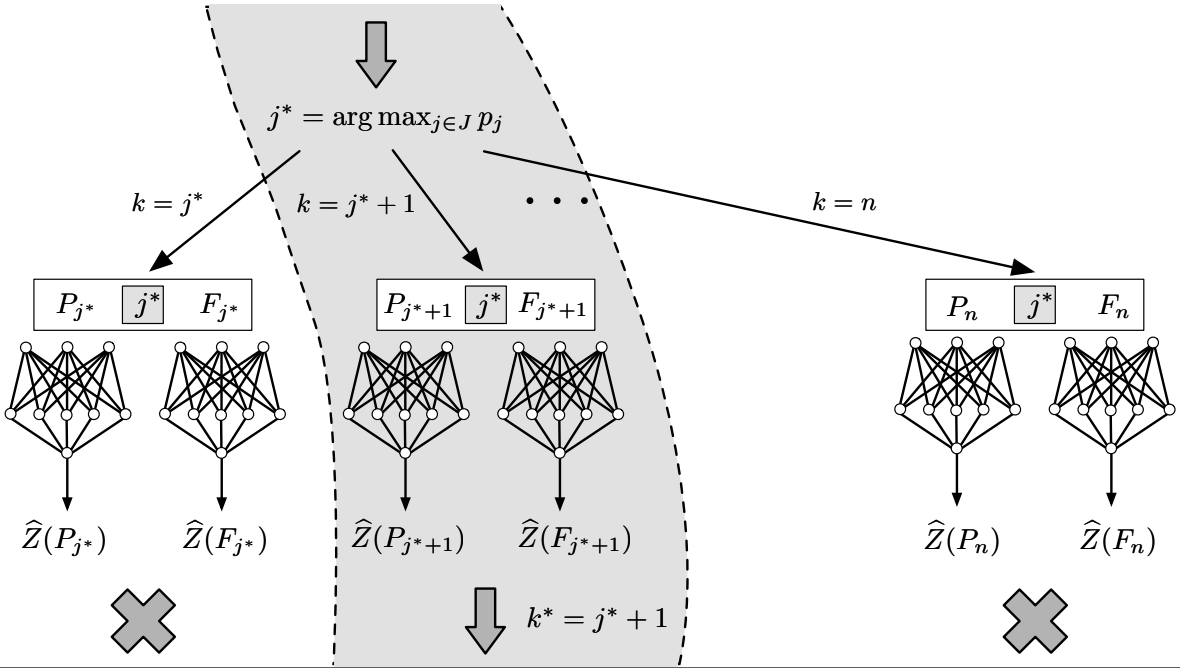
$$Z(P_k) + \max\left(0, \sum_{j\in\{1,\dots,k\}\setminus\{j^*\}} p_j - d_{j^*}\right) + Z(F_k)$$

- Jobs  (sorted in EDD)
- Job with maximal processing time 
- Possible decomposition
  - 
  - 
  - 
  - 

Proposed solution: Heuristic based on the decomposition

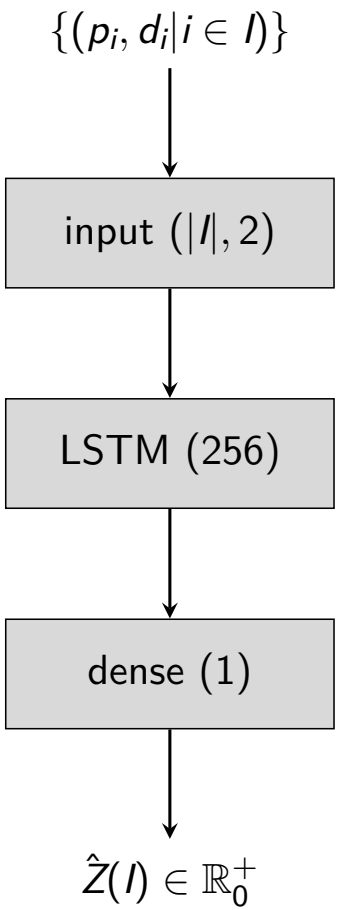
$$Z(P_k) + \max\left(0, \sum_{j\in\{1,\ldots,k\}\setminus\{j^*\}} p_j - d_{j^*}\right) + Z(F_k)$$

- Rank each possible decomposition by a regressor
- Greedy / Limited discrepancy search
- Position filtering rules adopted from TTBR algorithm



Proposed solution: Neural network as the regressor

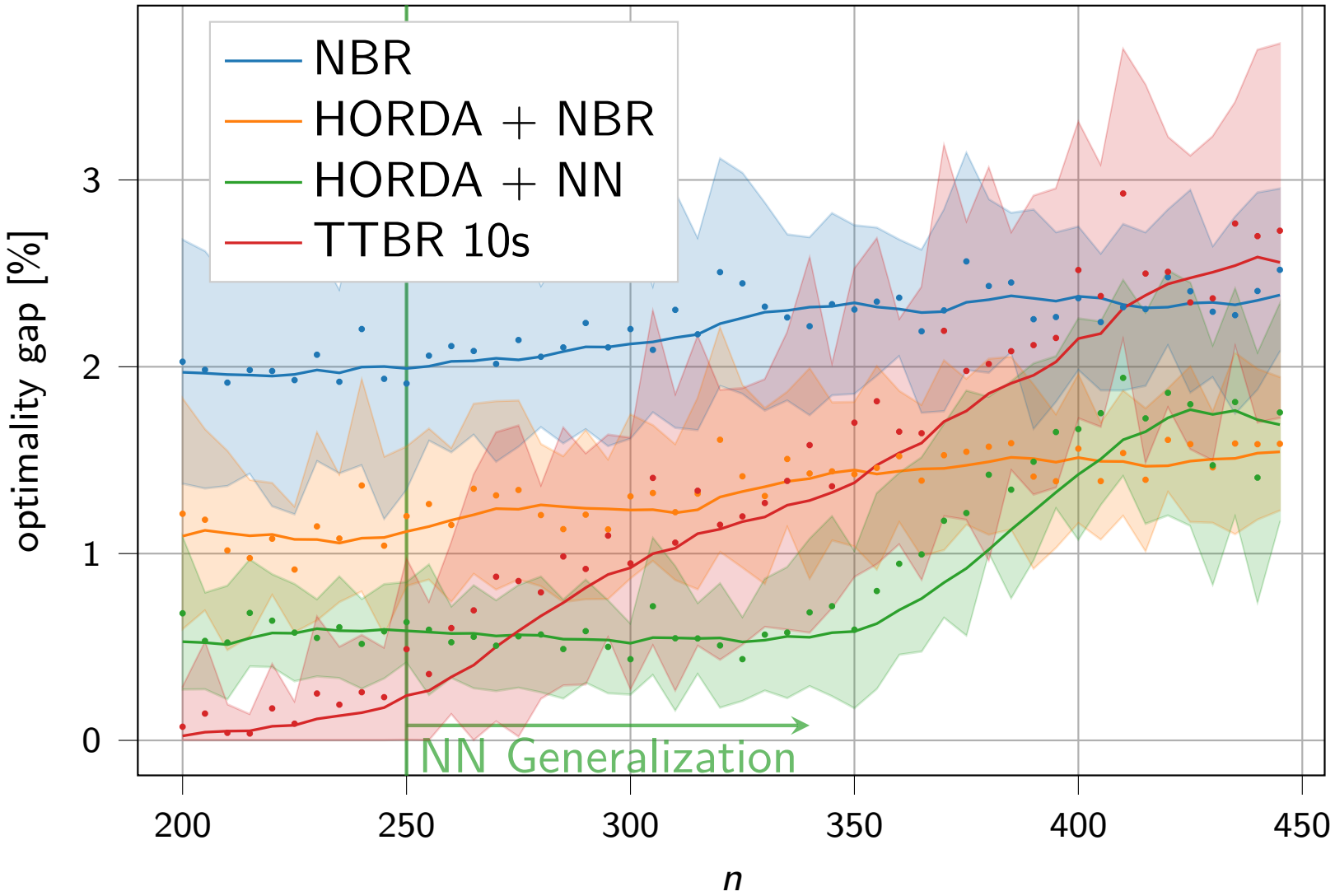
- Processing time and due date on input
- **Variable size of input**
  - LSTM recurrent layer
- Strongly supervised setting
- Optimal value as output
- Data preprocessing
  - Processing times and due dates
  - Criterion value
  - Sorting of input
  - Extension of input by a positional feature



Results: Comparison to SOTA OR heuristic

- Experiments:
  - Size of training set
  - Preprocessing of input
    - Limited discrepancy search
    - Processing times and due dates
    - Criterion value
    - Sorting of input
    - Extension of input by a positional feature
  - Training on instances created with a custom algorithm
  - Limited discrepancy search

Results: Comparison to SOTA OR heuristic



Conclusion

- The proposed solution has average optimality gap more than two times lower than SOTA NBR heuristic in decomposition.
- Lawler heuristic search with the proposed neural network as regressor has average optimality gap 0.29% on instances for n to 200.
- This work demonstrates advantages of ML and OR integration.
- Thank you for attention