

Background

General Disjunctive Program (GDP) ^{1,2}

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{Y}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) \leq 0 \\ & \bigvee_{j \in D_k} \left[h_{jk}(\mathbf{x}) \leq 0 \right], \quad \forall k \in K \\ & \Omega(\mathbf{Y}) = \text{True} \\ & \mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U] \subseteq \mathbb{R}^n \\ & Y_{jk} \in \{\text{False}, \text{True}\}, \quad \forall j \in D_k, \forall k \in K \end{aligned}$$

Assumed $h_{jk}(\mathbf{x}) = \mathbf{A}_k \mathbf{x} - \mathbf{b}_{jk}$

Big-M reformulation of constraints in disjunctions

$$\begin{aligned} \mathbf{A}_k \mathbf{x} - \mathbf{b}_{jk} &\leq \mathbf{M}_{jk}(1 - y_{jk}) \quad \forall k \in K, j \in D_k \\ \sum_{j \in D_k} y_{jk} &= 1 \quad \forall k \in K \\ \mathbf{M}_{jk} &= \max_{\mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U]} \{\mathbf{A}_k \mathbf{x} - \mathbf{b}_{jk}\} \end{aligned}$$

- Constraint enforced exactly or trivially by having a large coefficient.
- Big-M constraint estimated by computing the maximum violation of the linear constraints in the disjunction.

Comparison of Big-M and Hull Reformulations

Big-M Reformulation

- Advantage:** simplicity and low modeling.
- Disadvantage:** sensitive to M - too small violates logic, too large weakens the relaxation and causes instability.

Partial Common Structure of Mixed-integer Program Reformulation

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) \leq 0 \\ & \mathbf{D}\mathbf{y} \leq \mathbf{d} \\ & \mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U] \subseteq \mathbb{R}^n \\ & y_{jk} \in \{0, 1\}, \quad \forall j \in D_k, \forall k \in K \end{aligned}$$

Hull ³ reformulation of constraints in disjunctions

$$\begin{aligned} \mathbf{x} &= \sum_{j \in D_k} \hat{\mathbf{x}}_{jk} \quad k \in K \\ \mathbf{A}_k \hat{\mathbf{x}}_{jk} - \mathbf{b}_{jk} y_{jk} &\leq 0 \quad \forall k \in K, j \in D_k \\ \mathbf{x}^L y_{jk} \leq \hat{\mathbf{x}}_{jk} \leq \mathbf{x}^U y_{jk} &\quad \forall k \in K, j \in D_k \\ \sum_{j \in D_k} y_{jk} &= 1 \quad \forall k \in K \end{aligned}$$

Hull ³ reformulation of constraints in disjunctions

- Continuous variables are disaggregated for each disjunction with the index set.
- Disaggregated variables are coupled with binary variables.

Hull Reformulation

- Advantage:** tighter relaxation than Big-M.
- Disadvantage:** Higher computational cost from the large model size due to variable disaggregation.

Results of the Scheduling Problem

Reaggregated Hull Reformulation of the Scheduling Problem

Time-Slot Concept

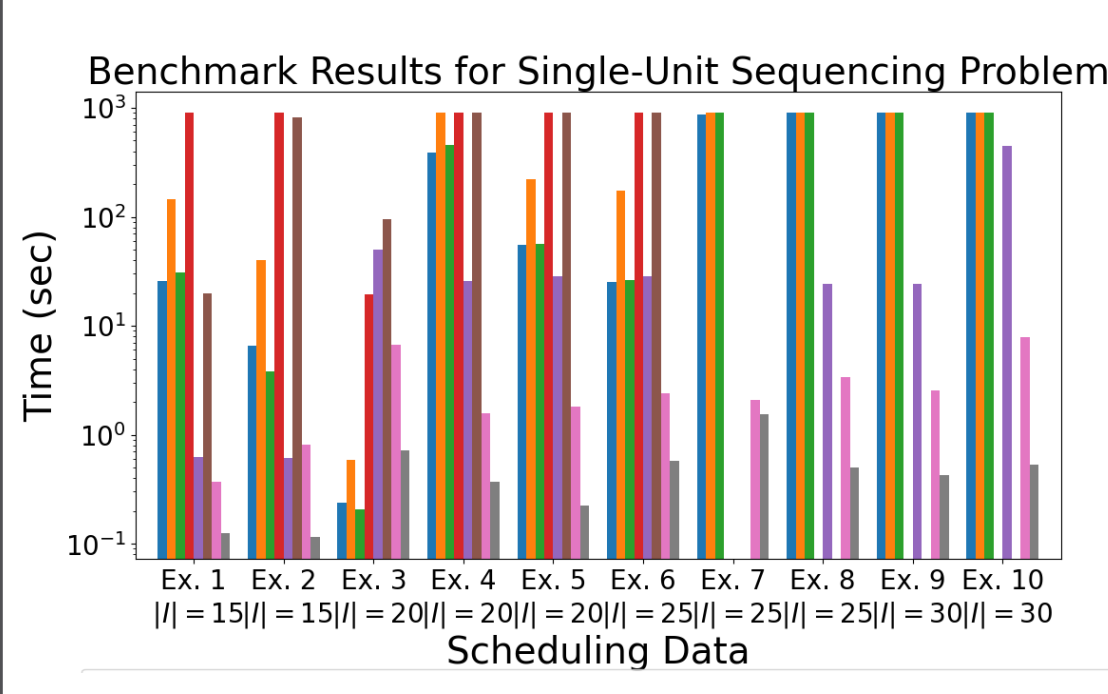
$$\begin{aligned} \min_{MS, \mathbf{x}, \mathbf{y}} \quad & MS \\ \text{s.t.} \quad & MS \lfloor t - \tau_j \rfloor + x_{t+1} \lfloor t - \tau_j \rfloor - x_t \geq \sum_{i \in I} p_i y_{it} \quad \forall t \in T \\ & x_t \geq \sum_{i \in I} r_i y_{it} \quad \forall t \in T \\ & x_t \leq \sum_{i \in I} (d_i - p_i) y_{it} \quad \forall t \in T \\ & \sum_{t \in T} y_{it} = 1 \quad \forall i \in I \\ & \sum_{i \in I} y_{it} = 1 \quad \forall t \in T \\ & y_{it} \in \{0, 1\} \quad \forall i \in I, \forall t \in T \\ & x_t \in \mathbb{R}_+ \quad \forall t \in T \\ & MS \in \mathbb{R}_+ \end{aligned}$$

General Precedence Concept

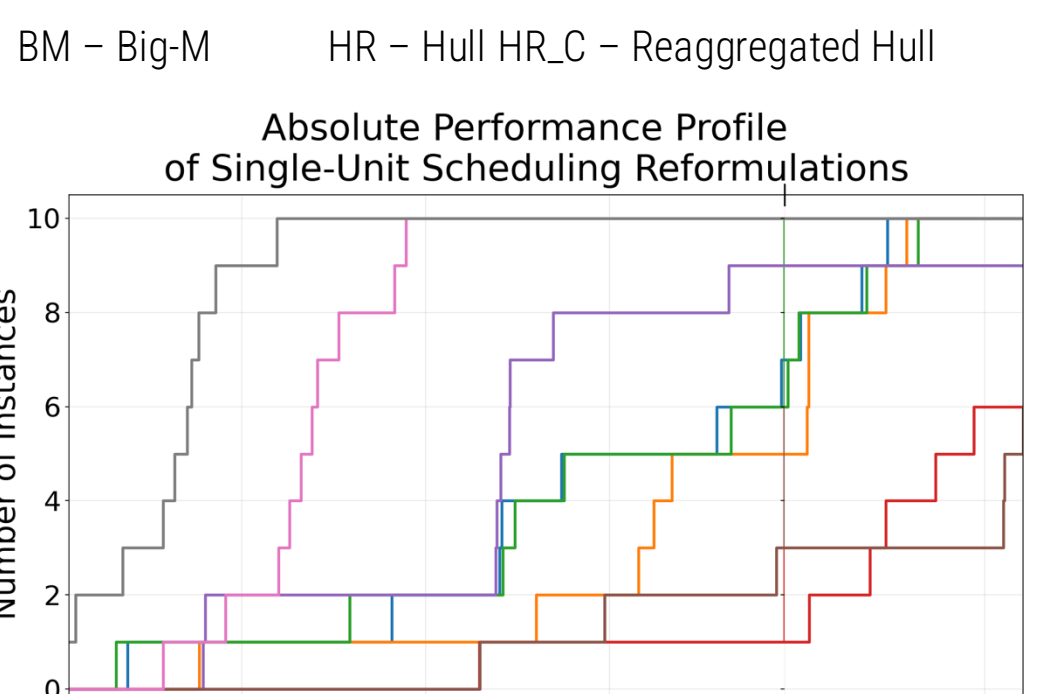
- Bounds created by algebraic constraints applied on disjunction, making the Reaggregated Hull applicable.

$$\begin{aligned} \min_{MS, \mathbf{x}, \mathbf{y}} \quad & MS \\ \text{s.t.} \quad & r_i \leq x_i \quad \forall i \in I \\ & x_i + p_i \leq d_i \quad \forall i \in I \\ & \left[r_i - (d_j - p_j) \leq x_i - x_j \leq \min(-p_i, (d_i - p_i) - r_j) \right] \vee \\ & \left[\max(p_j, r_i - (d_j - p_j)) \leq x_i - x_j \leq (d_i - p_i) - r_j \right] \quad \forall i \in I, \forall j \in I, i < j \\ & x_i + p_i \leq MS \quad \forall i \in I \\ & Y_{ij} \in \{\text{False}, \text{True}\} \quad \forall i \in I, \forall j \in I \\ & x_i \in \mathbb{R}_+ \quad \forall i \in I \\ & MS \in \mathbb{R}_+ \end{aligned}$$

Benchmark Results for Single-Unit Sequencing Problem



Absolute Performance Profile of Single-Unit Scheduling Reformulations



Reaggregated Hull Reformulation ³⁻⁶

Suppose a GDP model has disjunctions with linear constraints in which all disjuncts share the same left-hand-side coefficient matrix, as follows

$$\bigvee_{j \in D_k} \left[\mathbf{A}_k \mathbf{x} \leq \mathbf{b}_{jk} \right] \vee Y_{jk}$$

The Hull reformulation of disjunction is given as

$$\begin{aligned} \mathbf{x} &= \sum_{j \in D_k} \hat{\mathbf{x}}_{jk} \\ \mathbf{A}_k \hat{\mathbf{x}}_{jk} - \mathbf{b}_{jk} y_{jk} &\leq 0, \quad j \in D_k \\ \mathbf{x}^L y_{jk} \leq \hat{\mathbf{x}}_{jk} \leq \mathbf{x}^U y_{jk} \\ \sum_{j \in D_k} y_{jk} &= 1 \quad \forall k \in K \\ y_{jk} &\geq 0 \end{aligned}$$

Summing the individual constraints under $\mathbf{x} = \sum_{j \in D_k} \hat{\mathbf{x}}_{jk}$ yields $\mathbf{A}_k \mathbf{x} \leq \sum_{j \in D_k} \mathbf{b}_{jk} y_{jk}$

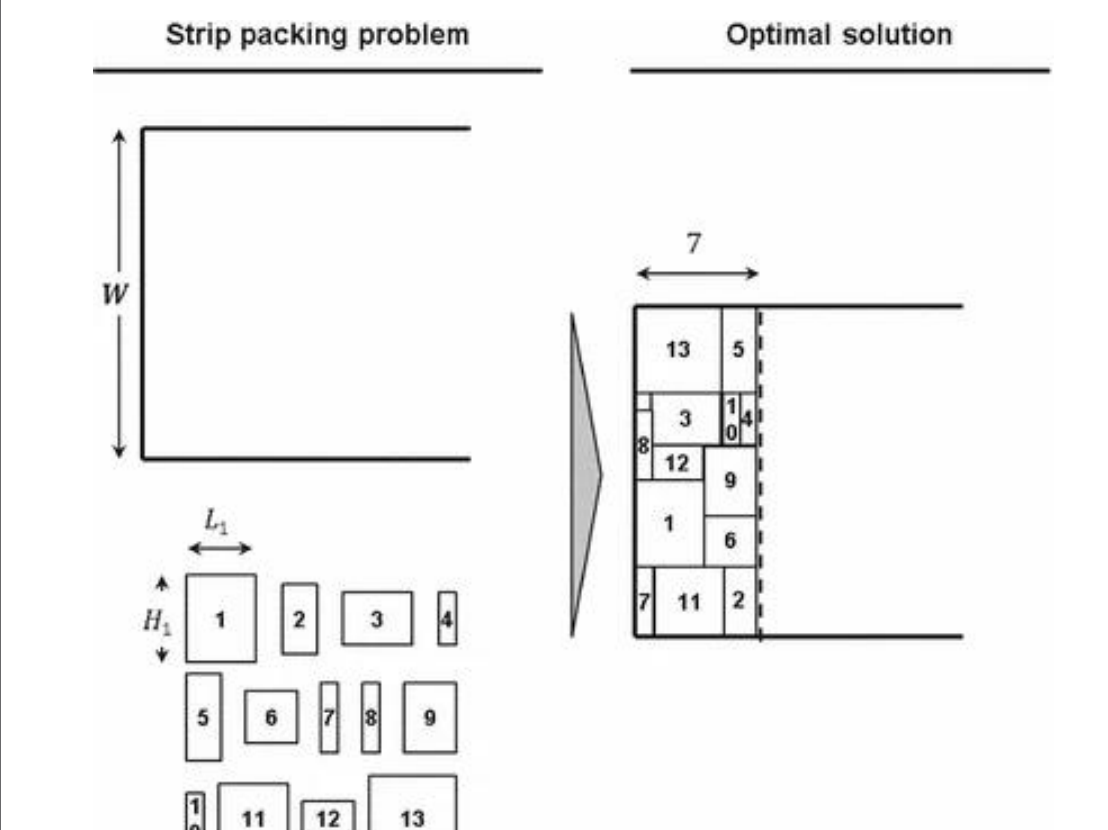
Thus, the reaggregated reformulation yields

$$\begin{aligned} \mathbf{A}_k \mathbf{x} &\leq \sum_{j \in D_k} \mathbf{b}_{jk} y_{jk} \\ \sum_{j \in D_k} y_{jk} &= 1. \end{aligned}$$

- Compact** - Removes all disaggregated continuous variables, keeping only the original problem variables for a smaller and cleaner formulation.
- Tight** - Maintains the exact convex-hull relaxation strength of the Hull reformulation.
- Faster** - Produces a smaller MILP with fewer constraints.

Strip-packing as GDP formulations ⁸

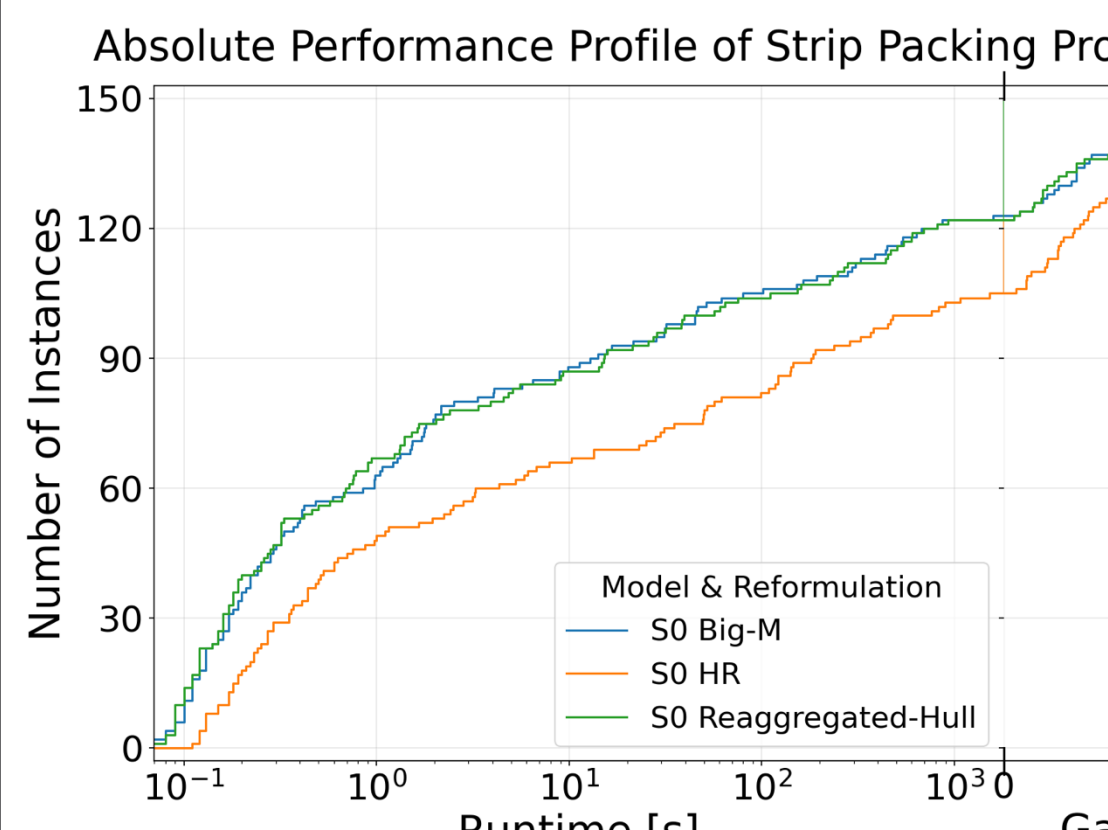
Strip Packing Problem



$$\begin{aligned} \min_{l_i, \mathbf{x}, \mathbf{y}, \mathbf{z}^1, \mathbf{z}^2} \quad & l_i \\ \text{s.t.} \quad & l_i \geq x_i + L_i, \quad \forall i \in N, \\ & \left[\begin{array}{l} Z_{ij}^1 \leq x_j \\ x_i + L_i \leq x_j \end{array} \right] \vee \left[\begin{array}{l} Z_{ij}^2 \leq x_j \\ x_j + L_j \leq x_i \end{array} \right] \quad \forall i, j \in N, i < j \\ & \left[\begin{array}{l} y_i - H_i \geq y_j \\ y_j - H_j \geq y_i \end{array} \right] \vee \left[\begin{array}{l} Z_{ij}^3 \leq y_i \\ y_i - H_i \geq y_j \end{array} \right] \quad \forall i, j \in N, i < j \\ & x_i \in [0, UB - L_i] \subseteq \mathbb{R}, \quad \forall i \in N, \\ & y_i \in [H_i, W] \subseteq \mathbb{R}, \quad \forall i \in N, \\ & Z_{ij}^1, Z_{ij}^2 \in \{\text{False}, \text{True}\}, \quad \forall i, j \in N, i \neq j. \end{aligned}$$

Arrange a set of non-rotatable rectangles into a fixed-width strip to minimize **total strip length**.

- Bounds created by algebraic constraints** applied to the disjunction enabled Reaggregated Hull for the strip-packing problem.



- Reaggregated-Hull vs. Hull:** The S0 reaggregated-Hull curve lies consistently above S0 HR, showing faster solve times and smaller optimality gaps across most instances
- Competitiveness with Big-M:** Reaggregated-Hull matches or slightly outperforms Big-M.
- Scalability advantage:** For medium and large instances, reaggregated-Hull closes gaps more quickly, whereas standard Hull lags behind in both runtime and solution quality.

Single-aUnit Scheduling as a GDP ⁷

General Precedence Concept (GP)

∨

$$Y_{ij} = \text{True} \quad Y_{ji} = \text{False} \quad \forall i \in I, \forall j \in I, i < j$$

Sequencing relationship between all pairs of jobs (i, j) which i < j, whether job i precedes job j or vice versa.

Immediate Precedence Concept (IP)

∨

∨

$$Y_{ij} = \text{True} \quad Y_{ji} = \text{True} \quad Y_i^{\text{last}} = \text{True} \quad \forall i \in I$$

∨

∨

$$Y_i^{\text{first}} = \text{True} \quad Y_i = \text{True} \quad Y_{j^i} = \text{True} \quad \forall i \in I$$

∨

$$Y_i^{\text{first}} = \text{True} \quad \forall i \in I$$

∨

$$Y_i^{\text{last}} = \text{True} \quad \forall i \in I$$

- Each disjunct specifies whether one job precedes the other jobs or is the first/last job.
- There is exactly one first job and one last job, and they must be different.

Time-Slot Concept (TS)

Time Slot

∨

∨

∨

∨

$$Y_{it} = \text{True} \quad Y_{jt} = \text{True} \quad \forall t \in T$$

∨

$$Y_{it} = \text{True} \quad Y_{jt} = \text{True} \quad \forall t \in T$$

- Each job is assigned exactly one time slot, where time slots equal jobs in number.

Conclusion

- Generalized Reaggregated Hull reformulation:** Developed a unified reformulation for GDP disjunctions with shared linear coefficients that retains convex-hull strength while removing disaggregated continuous variables, producing substantially smaller MILP models.
- Computational Performance Gains:** In single-unit scheduling and strip-packing benchmarks, the strengthened Hull consistently outperformed the traditional Hull and achieved solution times competitive with or better than Big-M formulations.
- Scalability and Robustness Demonstrated** consistent performance gains across problem sizes, confirming the method's practicality for large-scale GDPs.

References

- [1] Grossmann, I. E., & Lee, S. (2003). Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational optimization and applications*, 26(1), 83-100.
- [2] Trespalacios, F., & Grossmann, I. E. (2014). Review of mixed-integer nonlinear and generalized disjunctive programming methods. *Chemie Ingenieur Technik*, 86(7), 991-1012.
- [3] Balas, E. (1974). Disjunctive Programming: Properties of the Convex Hull of Feasible Points (No. RR348).
- [4] Balas, E. (1985). Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic Discrete Methods*, 6(3), 466-486.
- [5] Jeroslow, R. G. (1988). A simplification for some disjunctive formulations. *European journal of operational research*, 36(1), 116-121.
- [6] Blair, C. (1990). Representation for multiple right-hand sides. *Mathematical Programming*, 49(1), 1-5.
- [7] Castro, P. M., & Grossmann, I. E. (2012). Generalized disjunctive programming as a systematic modeling framework to derive scheduling formulations. *Industrial & Engineering Chemistry Research*, 51(16), 5781-5792.
- [8] Trespalacios, F., & Grossmann, I. E. (2017). Symmetry breaking for generalized disjunctive programming formulation of the strip packing problem. *Annals of Operations Research*, 258(2), 747-759.
- [9] Bynum, M. L., Hackbeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Sirlola, J. D., ... & Woodruff, D. L. (2021). *Pyomo-optimization modeling in python* (Vol. 67, No. s 32). Berlin/Heidelberg, Germany: Springer.
- [10] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2024. url: https://www.gurobi.com.

Acknowledgments

We acknowledge and express sincere gratitude to the Davidson School of Chemical Engineering at Purdue University for providing a supportive research environment and funding. We thank Gurobi Optimization for providing us with access to their software through their academic licensing program. We also thank Can Li for providing access to the hardware. All experiments were run on a single thread of an AMD EPYC 7643 48-Core Processor (96 logical cores) with 256 MB of L3 cache in a server running Red Hat Enterprise Linux 9.5 (Plow). The server is equipped with 1 TB of RAM and 15 GB of Swap Space.

Versions of software used: Pyomo - Version 6.9.0, Gurobi - Version 11.0

