Table of Contents

Evolution of Scoring Rules: A Summary

The scoring rules for both **column (variable)** and **row (constraint)** ordering have evolved to prioritize structural importance, numerical stability, and meaningful differentiation in optimization models. Below is a concise summary of the evolution and key features of these rules:

1. Initial Scoring Rules

Column (Variable) Ordering

$$\text{Score}_i = 100{,}000 \cdot P + 1 \cdot \sum (\log(1 + |\delta_j|)) + 100 \cdot \log(1 + |\text{Obj}_i|) + 1 \cdot \#\text{occurrences}$$

Row (Constraint) Ordering

$$\text{Score}_j = 100{,}000 \cdot P + 1 \cdot \sum (\log(1 + |\gamma_j|)) + 100 \cdot \log(1 + |\text{RHS}_j|) + 1 \cdot \log(1 + |\text{Range}_j|)$$

Key Features:

- **Type Priority (P):** Dominates with a multiplier of **100,000**, ensuring binary/integer variables and strict constraints ($>$, $=$) are prioritized.
- **Sum of Coefficients:** Measures the influence of variables/constraints across the model.
- **Objective/RHS Terms:** Logarithmic scaling ensures numerical stability for large values.
- **Occurrences/Range Terms:** Adds moderate influence based on participation in constraints or flexibility.

2. Enhanced Scoring Rules

Column (Variable) Ordering

$$\text{Score}_i = 1{,}000{,}000 \cdot P + 1 \cdot \sum \left(\log(1 + |\delta_j|)\right) + 100 \cdot \log(1 + |\text{Obj}_i|) + 1 \cdot \#\text{occurrences}$$

Row (Constraint) Ordering

$$\text{Score}_j = 1{,}000{,}000 \cdot P + 1 \cdot \sum \left(\log(1 + |\gamma_j|)\right) + 100 \cdot \log(1 + |\text{RHS}_j|) + 1 \cdot \log(1 + |\text{Range}_j|)$$

Key Enhancements:

- **Increased Multiplier for P:** The multiplier for type/sense priority was increased to **1,000,000** to enforce a stricter hierarchy.
- **Hierarchical Ordering:** Ensures binary variables and strict constraints ($>$, $=$) are always prioritized over others, regardless of intra-block scores.
- **Intra-Block Differentiation:** Sum of coefficients, objective/RHS, and occurrences/range terms refine ordering within each block.

3. Multi-Block Scoring Rules

Column (Variable) Ordering

- **Block Rules:** Variables are classified into blocks based on **type** (binary, integer, continuous) and **bounds** (finite, infinite, nonnegative).
- **Intra-Block Rules:** Sum of coefficients, objective contributions, and occurrences are combined into a single intra-block score.
- **Final Tuple:** Each variable is assigned a tuple $(BlockLabel1, BlockLabel2, \ldots, IntraBlockSum)$, sorted lexicographically.

Row (Constraint) Ordering

- **Block Rules:** Constraints are classified based on **sense** ($>$, $=$, $<$) and **composition** (integral, continuous, mixed).
- **Intra-Block Rules:** Sum of coefficients, RHS, and range terms are combined into a single intra-block score.
- **Final Tuple:** Each constraint is assigned a tuple $(SenseLabel, CompositionLabel, \ldots, IntraBlockSum)$, sorted lexicographically.

Key Innovations:

- **Multi-Dimensional Blocking:** Multiple block rules (e.g., type, bounds, sense, composition) create a richer hierarchy.
- **Lexicographic Sorting:** Tuples are sorted in descending order, ensuring block-level priorities dominate intra-block scores.
- **No Large Multipliers:** Eliminates the need for large multipliers by relying on tuple-based sorting.

4. Use Cases

The scoring rules are particularly effective in problems where:

- **Relative Differences Matter:** E.g., assignment, knapsack, and network design problems.
- **Structural Importance Varies:** E.g., binary/integer variables and strict constraints are prioritized.
- **Numerical Stability is Critical:** Logarithmic scaling handles wide ranges of coefficients and RHS values.

5. Key Takeaways

1. **Hierarchical Prioritization:** Type/sense priorities dominate, ensuring binary variables and strict constraints are always prioritized.
2. **Intra-Block Refinement:** Sum of coefficients, objective/RHS, and occurrences/range terms refine ordering within blocks.
3. **Multi-Block Flexibility:** Multiple block rules and lexicographic sorting enable richer, multi-dimensional ordering.
4. **Numerical Stability:** Logarithmic scaling ensures stability across wide ranges of values.

6. Tables of Typical Ranges

Column (Variable) Scoring

Term	Typical Min	Typical Max	Multiplier
Type Priority	1,000,000	3,000,000	1,000,000
Sum of Coefficients	0	~276,310	1
Objective Coefficient	0.0	~2,763	100
Occurrences	0	100+	1

Row (Constraint) Scoring

Term	Typical Min	Typical Max	Multiplier
Sense Priority	1,000,000	3,000,000	1,000,000
Sum of Coefficients	0	~276,310	1
RHS	0.0	~2,763	100
Range	0	~27.63	1

Conclusion

The evolution of scoring rules reflects a shift toward **hierarchical, multi-dimensional ordering** that prioritizes structural importance while maintaining numerical stability. By leveraging **block rules** and **lexicographic sorting**, the rules ensure meaningful differentiation without relying on large multipliers, making them suitable for a wide range of optimization problems.