INFORMS RAS 2026 Problem Competition

Railroad Blocking Problem for North American Class I Railroads

Dataset Description and Evaluation Criteria

This year's competition explicitly welcomes AI-based and hybrid submissions, including reinforcement learning, graph neural networks, and LLM-agent solvers, alongside traditional operations research methods.

Version: Draft 1.7e
Date: May 2026
Prepared by: RAS 2026 Problem Committee

1. Introduction

The **INFORMS Railway Applications Section (RAS) 2026 Problem Competition** addresses the **Railroad Blocking Problem** facing North American Class I freight railroads.

In North American freight rail operations, an operating plan involves three hierarchical decisions: (1) blocking, (2) train routing (train routes and block-to-train assignment), and (3) train scheduling. Due to the combinatorial complexity, these decisions are typically determined in a hierarchical manner. This competition focuses on the first and most strategic step: the blocking problem.

A block is a group of railcars consolidated at one classification yard and traveling intact (without splitting) to another classification yard. A blocking plan specifies which blocks to create (block design), how to route each commodity through a sequence of blocks (blocking sequence), and the physical network path for each block (block route). The railroad blocking problem is commonly formulated as a network design problem and has been studied extensively using a variety of exact and heuristic solution approaches (e.g., Newton et al., 1998; Barnhart et al., 2000; Ahuja et al., 2007). More broadly, freight railroad blocking belongs to the general class of transportation network design problems; interested readers are referred to Crainic and Hewitt (2021) for a comprehensive overview.

Following blocking, the train routing problem determines the train services used to transport each block through the network. Since a block represents a transportation request requiring pickup and delivery, the train routing problem (for manifest blocks) can be viewed as a variant of the pickup-and-delivery problem with transshipments (PDP-T). It simultaneously determines train routes and block-to-train assignments. A block may move through multiple trains via swaps (also referred to as transfers or transshipments), while each train may carry multiple blocks along its route.

Both the blocking plan and the train routing plan are static and do not explicitly consider time. A train scheduling plan complements the routing plan by incorporating train frequencies, operating days, and timetabling, typically over a weekly planning horizon. Together, blocking, train routing, and train scheduling define how each commodity traverses the designed service network as a complete itinerary. This integrated planning process is often referred to as trip planning. More details regarding operating plans and trip planning can be found in FreighRail101.

This competition considers a zero-based planning setting, meaning that the blocking plan is constructed from scratch without any pre-existing train or schedule information. The problem is static: all traffic demand is loaded instantaneously without time-dependent demand profiles. The input consists of a freight rail physical network and a set of yard-to-yard commodity demands. Participants are referred to Van Dyke and Meketon (2015) for a comprehensive review of the railroad blocking process.

All datasets are encoded using the General Modeling Network Specification (GMNS) format to facilitate interoperability and reproducibility. Since railroad blocking problems rely heavily on large-scale network routing and shortest path computations, participants may also find useful open-source routing and transportation network optimization resources in TAP101.

1.1 Scope and Commodity Types

This competition considers five freight commodity types, each with distinct operational rules:

Commodity Type	Block Category	Classification Yard Restriction	Blocking Rule
Merchandise	Manifest Block	Hump or Flat yard	Strictly separated; no commodity mixing within a block
Intermodal	Intermodal Block	Flat yard only (no hump)	Must be strictly separated
Automobile	Multilevel Block	Flat yard only (no hump)	Must be strictly separated
Coal	Bulk (Coal)	Hump or Flat yard	Unit train if large; strictly separated, no mixing
Grain	Bulk (Grain)	Hump or Flat yard	Unit train if large; strictly separated, no mixing

Merchandise represents the most common category of freight railcar traffic that requires classification and blocking through the yard network. Intermodal, automobile, coal, and grain commodities have their associated rules (e.g., hump yard prohibition for intermodal and auto traffic, unit train formation for coal and grain).

Specifically, intermodal commodity refers to container/trailer-based traffic handled through designated intermodal terminals rather than conventional hump classification yards. Commodity types restricted from hump classification (intermodal and automobile) may still physically pass through a yard on the main track or bypass route, but they cannot be processed (classified, re-blocked, or humped) at that yard. This distinction between pass-through traversal and classification processing is important for interpreting capacity constraints. Because intermodal and multilevel (automobile) blocks generally would not be reclassified at an intermediate yard, their blocking sequences are constrained to be direct — a single block from origin to destination. Multi-block sequences are permitted only for merchandise, coal, and grain.

2. Problem Description

The blocking problem involves three simultaneous decisions:

1. **Block Design:** Determine which yard-to-yard blocks to open.
2. **Blocking Sequence:** For each commodity, assign a path through the block service network, specifying a sequence of interconnected blocks from the origin yard to the destination yard. Each commodity must have exactly one blocking sequence (non-splittable volume).
3. **Block Route:** Determine the physical network path for each designed block. The default is the shortest path between the block origin and destination yards, subject to link capacity constraints. When capacity is binding, blocks may need to divert to alternative physical routes.

Note that these three decisions are solved simultaneously (unified), not sequentially. A block defines a service arc from one yard to another, and its physical route on the rail network is a separate decision that depends on link capacity interactions with other blocks.

2.1 Objective Function

The goal is to minimize the total cost comprising four components:

(a) **Fixed Cost:** A fixed cost (block_fixed_cost in setting.csv) is incurred for each block, representing the resources required to establish and maintain the block service. This discourages the creation of unnecessary blocks.

(b) **Transportation Cost:** The cost of moving railcars along block routes, computed as transport_cost_coefficient (from setting.csv) multiplied by commodity volume and block route distance measured in car-miles. This is computed as the commodity volume multiplied by the block route distance for each block used in the blocking sequence.

(c) **Classification Handling Cost:** A per-car handling cost is incurred at each intermediate yard where railcars are classified (reclassified) from one block into the next. Only inbound flow at intermediate yards is counted. The handling cost varies by yard.

(d) **Interchange cost:** A per-car interchange penalty is incurred for cross-railroad movement. It is computed from railroad-ownership changes along block routes, ignoring shared or connector elements with railroad_id = -1.

2.2 Constraints

The blocking plan must satisfy the following constraints:

C1. Flow Conservation: Every commodity must be routed from its origin yard to its destination yard through a connected sequence of blocks. Each commodity has exactly one blocking sequence (no splitting) without subtours. Additionally, each block's physical route must form a valid path on the released physical network, with consecutive node pairs connected by an existing link and with the route's start and end nodes matching the block's from-yard and to-yard.

C2. Classification Track Limit (Blocking Capacity): Each yard has a maximum number of outbound blocks it can build, limited by the number of classification tracks available. The total number of outbound blocks at a yard must not exceed this limit. For this constraint, only manifest and bulk blocks (i.e., classification blocks for merchandise, coal, and grain) consume the num_tracks budget. Intermodal and Multilevel (automobile) blocks bypass the classification bowl and use separate intermodal-ramp tracks or auto-facility loading slots, which are distinct physical infrastructure not represented in the num_tracks field.

C3. Throughput Capacity: Each yard has a maximum throughput capacity representing the total number of railcars it can classify within the operating cycle. The total volume of railcars handled (classified) at each yard must not exceed this capacity. It does not apply to blocks bypassing such a yard.

C4. Minimum Block Volume: To maintain the economies of scale, each block must carry a minimum number of railcars. The minimum volume is distance-dependent. These thresholds may be adjusted in specific problem instances.

Block Distance (miles)	Minimum Volume (railcars)
Less than 100 (exclusive)	350
100 to 500 (inclusive)	700
Greater than 500 (exclusive)	1050

C5. Link Throughput Capacity: The total volume of all blocks traversing a physical link must not exceed the link throughput capacity. This constraint governs the block route decision and may force block path diversion when shared links become congested.

C6. Maximum Circuitous Miles: For each block, its route on the physical network shall not exceed the corresponding shortest path from its origin to its destination by a specified margin. In this competition, this tolerance is quantified using a maximum circuitous ratio defined as follows.

$$\frac{\text{Distance of Block Route}}{\text{Shortest Path Distance}} \leq \text{max\_circuitous\_ratio}$$

C7. Single Path Uniqueness: For each commodity k (uniquely identified by the (commodity_id, commodity_type) pair), the submission reports exactly one blocking sequence and exactly one corresponding physical block route; the entire demand volume q_k flows along this single path. Splitting a commodity across multiple paths is not allowed. (This is the output-form counterpart to the no-splitting clause stated in C1.)

C8. Blocking Rule: A commodity can be only routed through a blocking sequence of its respective type. In other words, a block can only accommodate one type of commodity. This rule applies to all the five commodity types. Minimum block volume (C4) and all capacity/cost terms are evaluated per commodity type, not by aggregating mixed commodities.

2.3 Interchange Traffic

North America has six Class I railroads: Union Pacific (UP), BNSF, CSX, Norfolk Southern (NS), Canadian National (CN), and Canadian Pacific Kansas City (CPKC). When a block crosses between two railroads, an additional interchange cost (specified as interchange_cost in setting.csv) is incurred per car for such traffic. The interchange cost represents the combined handling, delay, and administrative penalty for transferring traffic between railroads.

In the initial sample problem (TWI Sample), all yards belong to a single railroad, so interchange costs do not apply. For the US National Model, the railroad_id field becomes essential, and interchange costs must be accounted for in the evaluation.3. Input Data Format

The competition dataset is provided as a set of CSV files following the GMNS convention. Each problem instance includes the following input files.

3.1 node.csv

It contains all nodes in the physical network. Nodes are categorized into three types: yard nodes (classification yards with blocking capabilities), station nodes (intermediate junctions on the physical network), and zone nodes (demand origin/destination zones, if applicable). The key fields are:

Field	Type	Description	Notes
`node_id`	int	Unique node identifier	Sequential integer

node_type	string	Node category	"yard", "station", or "zone"
x_coord	float	Longitude (WGS84)	Geographic coordinate
y_coord	float	Latitude (WGS84)	Geographic coordinate
name	string	Node display name	e.g., "JOLIET", "BIRMINGHAM"
yard_type	string	Classification yard type	"hump" or "flat" (yard nodes only)
yard_level	int	Yard size/importance level	1 = major hump yards; 2 = major flat yards; 3 = others
railroad_id	string	Owning railroad	UP, BNSF, CSX, NS, CN, or CPKC
num_tracks	int	Number of classification tracks (T_i)	Limits outbound blocks at this yard
handling_capacity	float	Throughput capacity in cars (W_i)	Max cars classified per operating cycle (default values provided by model developers)
handling_cost	float	Classification handling cost (h_i)	$/car at this yard
is_interchange	bool	Interchange point indicator	TRUE if yard serves as interchange

3.2 link.csv

It contains all physical track segments (edges) in the rail network. Links connect nodes and define the infrastructure over which blocks are physically routed.

Field	Type	Description	Notes
link_id	int	Unique link identifier	Sequential integer
from_node_id	int	Origin node of the link	References node_id in node.csv
to_node_id	int	Destination node of the link	References node_id in node.csv
length	float	Link length in miles	Used for car-mile cost calculation
capacity	float	Link throughput capacity (Cap_l)	Max total cars per operating cycle
railroad_id	string	Owning railroad	Used for interchange identification
free_speed	float	Maximum operating speed (mph)	For reference; not used in objective

3.3 demand.csv

It contains the commodity demand table. Each row represents a unique commodity (an OD demand record) specifying the origin yard, destination yard, volume in railcars, and commodity type. **All demands are specified as yard-to-yard pairs**, not zone-to-zone. If the original data source is zone-level, the mapping to yards has already been performed.

Field	Type	Description	Notes
commodity_id	int	Unique commodity identifier	Sequential integer
origin_yard_id	int	Origin yard node_id	Must reference a yard node

dest_yard_id	int	Destination yard node_id	Must reference a yard node
volume	int	Demand volume in railcars	Over the full operating cycle
commodity_type	string	Type of commodity	

3.4 setting.csv

It contains global parameters for the problem instance. These are provided as key-value pairs.

Parameter	Value	Unit	Description
min_block_vol_short(<100mi)	350	railcars	Min volume for blocks under 100 miles (70-day total)
min_block_vol_med(100-500mi)	700	railcars	Min volume for blocks 100-500 miles (70-day total)
min_block_vol_long(>500mi)	1050	railcars	Min volume for blocks over 500 miles (70-day total)
max_circuitous_ratio	1.3	ratio	Max detour factor vs. shortest path
operating_cycle	70	days	Planning horizon
block_fixed_cost	2500	$/block	Fixed cost per block (default provided by model developers)
transport_cost_coefficient	1	$/car-mile	Per car-mile transportation cost coefficient
interchange_cost	100	$/car	Per-car cost penalty for railroad interchange traffic

Note that the operating cycle is set to 70 days (10 weeks). All demand volumes represent the totals for the full operating cycle. All capacity values (handling_capacity, link capacity) should use the same time basis over the cycle. The typical operating plan period is a multiple of 7 days for North American freight railroads.

In this competition, transport_cost_coefficient is set to 1 to achieve a balanced trade-off between transportation and handling costs in the resulting blocking plan. Link throughput capacity (70-day) is assumed to be 140,000 cars. The actual per-yard handling_capacity values are provided in node.csv; the table below lists reference value ranges by yard level for orientation only — solvers must use the per-yard values from node.csv, not the table.

Yard Level	Handling Capacity (70-Day)
1	30,000 (typical) – 100,000 cars
2	1,500 – 20,000 (typical) cars
3	1,500 (typical) – 20,000 cars

Volume units: All demand volumes and block volumes are expressed in car-equivalent units (railcars). If the original data uses different native units for certain commodity types, the conversion to car-equivalents has already been performed.

Demand multiplier: setting.csv carries `demand_multiplier` (default 1.0; sub-cases 0.5× and 2.0× per § 5.5). All demand volumes are scaled by this factor before any check or cost calculation. Submissions must echo the chosen value into `inputs.settings.demand_multiplier` of solution_result.json; the validator assumes 1.0 if absent.

Numeric fields that are not applicable for a given row (for example `yard_level` on a non-yard node) are encoded as **-1** rather than as an empty cell, so that downstream parsers do not need to handle a separate “missing” sentinel and can keep the column dtype as integer or float. Any numeric column where the value is reported as -1 should be interpreted as “not applicable / not available” for that row. String fields that are not applicable remain empty (“”).

4. Solution Output Format

Participants must submit their solution as a single JSON file containing both the input data (for validation) and the solution output. The JSON structure has two top-level keys:

4.1 JSON Structure Overview

The output file `solution_result.json` must contain:

1. **inputs:** A copy of all input data (settings, nodes, demands, links) for reproducibility and validation.
2. **outputs:** The solution consisting of three components corresponding to the three decisions.

4.2 Output Component 1: Block Design

It shall outline a list of designed blocks along with the following fields.

Field	Type	Description
`block_id`	int	Unique block identifier
`from_yard_id`	int	Block origin yard
`to_yard_id`	int	Block destination yard
`commodity_type`	string	Commodity type of the block (merchandise, intermodal, automobile, coal, or grain)
`block_volume`	int	Volume in railcars (car-equivalent units)

4.3 Output Component 2: Blocking Sequence

For each commodity, the blocking sequence specifies the route through the designed blocking service network. Each entry contains:

Field	Type	Description
`commodity_id`	int	The commodity being routed
`commodity_type`	string	Type of the commodity

`origin_yard_id`	int	Origin yard of the commodity
`dest_yard_id`	int	Destination yard of the commodity
`volume`	int	Demand volume in railcars
`blocking_sequence`	string	Sequence of block_ids, e.g. "6 -> 12" meaning block 6 then block 12

4.4 Output Component 3: Block Route

Each block has a physical route on the rail network. Each entry contains:

Field	Type	Description
`block_id`	int	The block being routed on the physical network
`from_yard_id`	int	Block origin yard
`to_yard_id`	int	Block destination yard
`physical_path_nodes`	string	Sequence of node_ids on the physical path, e.g. "11151 -> 11143"
`physical_path_links`	string	Sequence of link_ids traversed, e.g. "8"

5. Problem Instances

The competition provides multiple problem instances of increasing scale. Participants are required to solve the mandatory instances and are encouraged to attempt the advanced national-scale instance.

Attribute	Sample (Practice)	*L1 — Super-hump*	*L2 — Major-flat*	L3 — Beyond major flat
Yards as Origins	8	21	132	1,041
OD Pairs	50	411	11,897	136,677
Physical Links	15	106,570	106,570	106,570
Railroads	1 (single)	6 (all Class I)	6 (all Class I)	6 (all Class I)
Interchange	No	Yes	Yes	Yes
Commodity Focus	Merchandise	All 5 types	All 5 types	All 5 types
Data Source	Constructed	Research-derived from BTS/FAF & OSM	Research-derived from BTS/FAF & OSM	Research-derived from BTS/FAF & OSM
Demand scenarios	1.0×	0.5× / 1.0× / 2.0×	0.5× / 1.0× / 2.0×	0.5× / 1.0× / 2.0×

5.1 TWI Sample Network (Practice)

The sample dataset provides a small network for development and testing. It consists of 8 classification yards (all flat type, single railroad BNSF), 5 intermediate stations, 15 physical links, and 50 commodity OD pairs with mixed commodity types. The network covers a region in the southeastern United States with yards including JOLIET, CURTIS, OTTAWA, MARTIN, MANCHESTER, BIRMINGHAM REG INTERMODAL, S&N, and SIBERT.

This dataset is provided in both CSV format (node.csv, link.csv, demand.csv, setting.csv) and as a complete JSON solution example (solution_result.json). A Python validator (validator.py) is included.

5.4 US National Model (Challenge)

The released benchmark is shipped at three network-resolution layers (L1, L2, L3) sharing the same physical network of 47,193 nodes and 106,570 links. The L3 layer is the largest layer, with 1,041 origin yards, 136,677 OD pairs, and all five commodity types (intermodal, merchandise, coal, grain, automobile). The instance covers all six Class I railroads (UP, BNSF, CSX, NS, CN, CPKC) and includes designated interchange points between them, so multi-railroad operations and interchange costs apply at L3. On the other hand, L1 and L2 are smaller subsets of the same national network and are intended for rapid prototyping and intermediate-scale validation, respectively. Section 5.5 details the L1/L2/L3 layers and the demand-scaling sub-cases. Participants are encouraged to solve the largest layer their methods can scale to.

5.5 Demand Scenario Layers (Network Resolution × Demand Scaling)

The released benchmark exposes a two-axis scenario matrix. The first axis is the network-resolution layer (L1, L2, L3); the second axis is a demand-volume scaling factor applied uniformly to all OD records. The intent is to let participants exercise their methods at three levels of geographic detail and at three demand intensities each, producing a 3 × 3 = 9-cell evaluation grid.

Network resolution layers (sharing the same physical network):

L1 — Super-hump skeleton. 21 origins / 21 destinations, 411 OD pairs. Demand is restricted to the 21 Class I super-hump complexes; suitable for rapid prototyping and for verifying that a method scales from a small instance.

L2 — Major-flat-yard tier. 132 origins / 132 destinations, 11,897 OD pairs. Adds the next tier of major flat yards onto L1; introduces hub-and-spoke routing pressure and stresses C2/C3 budgets.

L3 —Beyond major flat. 1,041 origins / 1,413 destinations, 136,677 OD pairs. The full curated national benchmark; designed to exercise scalability and the long-tail of low-volume ODs.

Demand scaling sub-cases (apply within each L1/L2/L3 layer):

0.5× — Light load. All OD volumes scaled by 0.5. Used as a lower-bound sanity case; few constraints bind and most reasonable methods should achieve a fully-served plan.

1.0× — Baseline. The released volumes used as-is. The primary scenario for cost-based ranking.

2.0× — Stress. All OD volumes doubled. Used as a stress scenario where C2 (track) and C3 (handling) budgets become the binding constraints; fully feasible plans may not exist, in which case the Loaded Demand Ratio and Stress Score (see § 6.4) become the primary evaluation metrics.

The 9-cell scenario matrix is computed from a single underlying benchmark by the `demand_multiplier` parameter in setting.csv (see § 3.4). For each L1/L2/L3 layer, the package ships a single demand.csv plus three setting variants (one per multiplier value: 0.5, 1.0, 2.0); the participant chooses a (layer, multiplier) cell, runs their solver, and records the chosen multiplier in `inputs.settings.demand_multiplier` of solution_result.json. The validator reads the multiplier from the submission and applies it before computing constraint checks and metrics, so demand.csv ships once per layer rather than as three separate pre-scaled CSV copies.

6. Evaluation Criteria

Solutions will be evaluated in two phases: feasibility validation and cost-based ranking.

6.1 Feasibility Validation

A provided validator program checks the constraints specified in Section 2.2. A solution that fails any constraint is deemed infeasible.

6.2 Cost-Based Ranking

Submissions are ranked by Stress Score (defined in § 6.4). On scenarios where a strictly feasible plan is achievable (typically the 0.5× and 1.0× sub-cases), the Loaded Demand Ratio equals 1 and the Stress Score reduces to Operating Cost. On stress sub-cases (2.0× and any layer where strict feasibility is unattainable), the unserved-demand penalty term is non-zero and the Stress Score becomes the operative ranking metric. Operating Cost is computed as:

Operating Cost = Block Fixed Cost + Transportation Cost + Classification Handling Cost+ Interchange Cost

Where:

- **Block Fixed Cost** = Sum of block_fixed_cost for all blocks,
- **Transportation Cost** = Sum over all commodities of (transport_cost_coefficient × volume × block route distance) for each block in the commodity's blocking sequence,
- **Classification Handling Cost** = Sum over all intermediate yards of (handling_cost × total inbound volume classified at that yard).
- **Interchange Cost = `interchange_cost` × total cars moved across railroad-ownership changes along block routes.**

6.3 Scoring Rubric

The final competition score considers multiple dimensions:

Criterion	Weight	Description
Solution Quality	50%	Total cost of the blocking plan (lower is better).
Feasibility	20%	Satisfaction of all constraints.
Scalability	15%	Ability to solve larger instances (L2 and L3 layers of the US National Model). Bonus credit for demonstrating scalable approaches.
Methodology & Report	15%	Technical merit of the solution approach, clarity of documentation, and reproducibility.

6.4 Loaded Demand Ratio and Stress Score (volume-weighted)

For instances in which a fully feasible solution may not exist (e.g., the 2.0× stress scaling within any L1/L2/L3 layer), submissions are evaluated using two volume-weighted metrics that complement Operating Cost. These metrics use the demand volume q_k (in railcars) of each commodity k, not the count of OD pairs, so that high-volume corridors dominate the score and a solver cannot improve its rank by simply dropping low-volume hard-to-serve OD pairs.

Loaded Demand Ratio. Volume-weighted fraction of total demand that the submitted plan actually flows (i.e., that appears in at least one valid blocking sequence with no constraint violation):

$$\text{Loaded Demand Ratio} = \Sigma_k \, q_k^{served} \, / \, \Sigma_k \, q_k^{total} \,;$$

$$\text{Unserved Demand Ratio} = 1 - \text{Loaded Demand Ratio}.$$

Stress Score. Composite cost that combines the achieved Operating Cost with a per-car-mile penalty for stranded traffic:

$$\text{Stress Score} = \text{Operating Cost} + M \times \Sigma_k \, q_k^{stranded} \times d_k^{NR}$$

where

- *Operating Cost* = Block fixed Cost + Transportation Cost + Classification Handling Cost + Interchange Cost (per § 6.2);
- $q_k^{stranded}$ is the stranded volume (in railcars) of commodity k;
- d_k^{NR} is the shortest-path physical distance (miles) between the origin and destination yards of commodity k, computed on the released physical network;
- *M* is a per-unit penalty coefficient ($/car-mile). The competition default is **M = 5** (i.e., five times the transport_cost_coefficient = 1 declared in setting.csv.

The Stress Score is monotone in unserved volume and in the geographic difficulty of unserved ODs (their shortest-path distance), so a solver cannot reduce it by selectively dropping high-distance hard-to-serve OD pairs — doing so increases the penalty proportionally to the imputed transport cost the dropped traffic would have required.

Reporting requirement. Submissions on stress scenarios (2.0× demand or any layer where full feasibility is not attainable) must report Operating Cost, Loaded Demand Ratio, and Stress Score together. Cost-based ranking on these scenarios uses Stress Score (not Operating Cost alone). On non-stress scenarios, Stress Score reduces to Operating Cost (Loaded Demand Ratio = 1).

7. Remarks on Problem Settings

In this competition, several simplifying assumptions are introduced that deviate from real-world railroad operating practices.

1. **Bulk Commodities and Unit Trains.** Coal and grain are two common examples of bulk commodities, although freight railroads transport a wide variety of bulk goods ranging from agricultural and energy products to construction materials, industrial products, chemicals, and minerals. Bulk traffic is typically characterized by high-volume demand between a single origin and destination pair. In practice, these shipments are commonly transported using unit trains consisting of a single car type moving directly between origin and destination industrial facilities, rather than through intermediate classification yards.

Modeling such operations would require substantially more detailed demand information and explicit representations of industrial facilities and terminal operations, which are beyond the scope of this competition.

2. **Intermodal Traffic and Terminals.** Similar to bulk traffic, intermodal traffic is generally managed separately from conventional manifest traffic and handled through designated intermodal facilities, commonly referred to by railroads as intermodal terminals. These operations involve distinct handling processes, terminal capacities, service requirements, and operational considerations that are not explicitly modeled in this competition.
3. **Commodity Types and Blocking Rules.** Commodity types in North American freight railroads are often separated according to handling requirements, equipment restrictions, service rules, and customer considerations. For simplicity, this competition considers only five commodity categories and enforces strict separation among them.

   In practice, blocking involves much more than selecting least-cost routes through a blocking plan. Each commodity must satisfy numerous compatibility constraints associated with commodity attributes and block characteristics, including factors such as plate size, hazardous material restrictions, consignor and consignee requirements, equipment compatibility, and service commitments. As a result, the blocking sequences used in practice are frequently not the minimum-cost alternatives from a purely network optimization perspective.
4. **Interchange**. Interchange between railroads is modeled at the macroscopic OD/block level in this competition. We do NOT require participants to identify a specific physical interchange node, yard pair, or intermodal transfer location.

In practice an interchange point is often an area-level concept (e.g., the Chicago region), where multiple yards owned by different railroads may exist and freight may need yard switching, drayage, or intermodal reload to transfer. Modeling such detailed yard-to-yard transfer links would require an additional train-layer or terminal-layer network representation, which is outside the scope of this competition. Interchange is therefore treated as an aggregate cost/penalty on cross-railroad movement, not a node-specific operational transfer constraint.8. Provided Files and Tools

The competition package is organized into two main components: input data and scoring tools.

• **Data files:** Each problem instance includes node.csv, link.csv, demand.csv, and setting.csv. These files define the physical network, yard-to-yard demand, and global parameters for the instance.

• **Scoring tools:** The package includes Python-based validators and scoring scripts. These tools check feasibility constraints, report violations, and compute the evaluation metrics used for scoring. Sample solution_result.json files are also provided to illustrate the required output format.

For detailed file schemas, output format requirements, and tool usage instructions, please refer to the corresponding README files included in the dataset and scoring folders.

9. References

1. Van Dyke C, Meketon M (2015) Railway Blocking Process. Patty BW, ed. Handbook of Operations Research Applications at Railroads (Springer US, Boston, MA), 119–162.
2. Newton HN, Barnhart C, Vance PH (1998) Constructing railroad blocking plans to minimize handling costs. Transportation Science 32(4):330–345.
3. Barnhart C, Jin H, Vance PH (2000) Railroad blocking: A network design application. Operations Research 48(4):603–614.
4. Ahuja RK, Jha KC, Liu J (2007) Solving real-life railroad blocking problems. Interfaces 37(5):404–419.
5. Crainic TG, Hewitt M (2021) Service network design. In Crainic TG, Gendreau M, Gendron B, eds. Network Design with Applications to Transportation and Logistics (Springer), 347–382.
6. INFORMS RAS 2019 Problem Solving Competition: Integrated Train Blocking and Shipment Path Optimization (TBSP). https://connect.informs.org/railway-applications/new-item3/problem-solving-competition682/problem-solving-competition2019
7. General Modeling Network Specification (GMNS). https://github.com/zephyr-data-specs/GMNS
8. FreightRail101. https://github.com/jdlph/FreightRail101
9. TAP101. https://github.com/jdlph/TAP101
10. U.S. Bureau of Transportation Statistics (BTS). Freight Analysis Framework (FAF) and national rail network datasets. Multiple versions, https://www.bts.gov/.

Data Provenance and Limitation Disclaimer

The demand side of the benchmark is partially derived from and calibrated against public freight-flow data, including BTS/FHWA Freight Analysis Framework (FAF) origin-destination commodity flow patterns. FAF provides freight-flow estimates by origin, destination, commodity type, and transportation mode, including state-level and FAF-zone-level OD structures. However, the released competition instances are not direct BTS or FAF datasets. The OD flows have been transformed into railcar-equivalent yard-to-yard commodity records through aggregation, mapping, filtering, scaling, and benchmark calibration. The physical network connectivity, yard locations and attributes, capacity assumptions, and cost parameters are research-derived and curated by the RAS Class I industry partners and the problem-development team for algorithmic evaluation. The physical network connectivity is also research-derived and integrated from open-source spatial references such as OpenStreetMap (OSM); the released benchmark should therefore be cited as research-derived data integrated from public freight-flow references (BTS/FAF-style OD patterns) and open-source network references (OSM-style connectivity), not as a direct BTS, FAF, or OSM extract.

End of Document