

Can you suggest how a user of the first evaluation matrix (covering 10 methods) could systematically optimise a choice of 2 or more methods such that they addressed all the question types with the highest possible method ratings, while minimising the number of methods used?

Systematic Method Optimization Guide

The Optimization Problem

Objective: Select the minimum number of methods that provide HIGH ratings across all five evaluation question types.

Constraint: Each question type must be covered by at least one HIGH-rated method.

This is a variant of the "set cover problem" from operations research, adapted for evaluation planning.

Systematic Procedure

Step 1: Identify Coverage Requirements

Create a coverage matrix showing which methods achieve HIGH ratings for each question type:

Question Type Methods with HIGH Ratings

Descriptive Horizon Scanning, Trend Analysis, Wild Cards/Weak Signals

Valuative Delphi Method, Backcasting

Explanatory Cross-Impact Analysis, Futures Wheel, Causal Layered Analysis

Predictive Scenario Planning, Delphi Method, Futures Wheel, Trend Analysis

Prescriptive Backcasting

Step 2: Identify Forced Selections

Some question types have limited HIGH-rated options:

- **Prescriptive** has only ONE HIGH option (Backcasting) → **Must select Backcasting**
- **Valuative** has only TWO HIGH options (Delphi, Backcasting) → At least one required

Since Backcasting is the only Prescriptive HIGH and also covers Valuative, it must be in any optimal solution.

Step 3: Determine Remaining Coverage Needs

After selecting Backcasting:

- ✓ Valuative: Covered (Backcasting HIGH)
- ✓ Prescriptive: Covered (Backcasting HIGH)
- ✗ Descriptive: Need HIGH coverage
- ✗ Explanatory: Need HIGH coverage
- ✗ Predictive: Need HIGH coverage

Step 4: Find Methods Covering Multiple Remaining Needs

Identify methods that cover multiple uncovered question types:

Multi-coverage methods:

- **Futures Wheel:** Explanatory HIGH + Predictive HIGH (2 needs)
- **Trend Analysis:** Descriptive HIGH + Predictive HIGH (2 needs)
- **Delphi Method:** Predictive HIGH (1 need, but Valuative already covered)

Step 5: Select Complementary Methods

Optimal 3-Method Solutions:

Solution A: Backcasting + Futures Wheel + Trend Analysis

- Descriptive: Trend Analysis (HIGH)
- Valuative: Backcasting (HIGH)
- Explanatory: Futures Wheel (HIGH)
- Predictive: Futures Wheel (HIGH) + Trend Analysis (HIGH) - redundant coverage
- Prescriptive: Backcasting (HIGH)

Solution B: Backcasting + Futures Wheel + Horizon Scanning

- Descriptive: Horizon Scanning (HIGH)
- Valuative: Backcasting (HIGH)
- Explanatory: Futures Wheel (HIGH)
- Predictive: Futures Wheel (HIGH)
- Prescriptive: Backcasting (HIGH)

Solution C: Backcasting + Cross-Impact Analysis + Trend Analysis

- Descriptive: Trend Analysis (HIGH)
- Valuative: Backcasting (HIGH)
- Explanatory: Cross-Impact Analysis (HIGH)
- Predictive: Trend Analysis (HIGH)
- Prescriptive: Backcasting (HIGH)

Step 6: Verify Minimum

Can we achieve full HIGH coverage with only 2 methods?

Testing 2-method combinations:

- Backcasting is required (covers Valuative + Prescriptive)
- Need one method covering: Descriptive + Explanatory + Predictive (all HIGH)
- **No single method provides all three remaining HIGH ratings**

Conclusion: Minimum is 3 methods

Decision Framework for Selecting Among Optimal Solutions

Since multiple 3-method combinations achieve full HIGH coverage, additional criteria help choose:

Criterion 1: Redundancy/Backup Coverage

Which solution provides backup HIGH ratings?

- **Solution A:** Has redundant Predictive coverage (both Futures Wheel and Trend Analysis) - provides resilience
- **Solutions B & C:** No redundancy - more efficient but less robust

Recommendation: Solution A preferred if resources allow full implementation; ensures one method can fail without losing HIGH coverage.

Criterion 2: Method Compatibility

Consider whether methods complement each other in practice:

Solution A (Backcasting + Futures Wheel + Trend Analysis):

- Trend Analysis provides data foundation
- Futures Wheel explores consequences
- Backcasting defines pathways to desired futures
- Strong sequential workflow potential

Solution B (Backcasting + Futures Wheel + Horizon Scanning):

- Horizon Scanning identifies emerging signals
- Futures Wheel explores implications
- Backcasting plans response pathways
- Strong emerging-issues workflow

Solution C (Backcasting + Cross-Impact Analysis + Trend Analysis):

- Trend Analysis identifies patterns
- Cross-Impact Analysis models interactions
- Backcasting designs interventions
- Strong quantitative-emphasis workflow

Criterion 3: Resource Requirements

Consider implementation effort:

Complexity ranking (lowest to highest):

1. Trend Analysis (straightforward data analysis)
2. Horizon Scanning (systematic but not technically complex)
3. Futures Wheel (structured brainstorming)
4. Backcasting (requires normative consensus)
5. Cross-Impact Analysis (requires interaction modeling)

Solution B has lowest average complexity. **Solution C** has highest average complexity.

Criterion 4: Stakeholder Engagement

Methods vary in participatory potential:

High participation methods: Backcasting, Futures Wheel, Horizon Scanning
Lower participation methods: Trend Analysis, Cross-Impact Analysis

Solution B maximizes stakeholder engagement opportunities.

Beyond Minimum: Strategic Expansion

When to Add a 4th Method

Consider adding methods when:

1. Critical gap in current coverage: Even with HIGH ratings in all categories, you might lack:

- **Delphi Method** for expert consensus when Backcasting's single HIGH for Valuative feels insufficient
- **Causal Layered Analysis** for deeper explanatory capacity when dealing with complex worldview issues

2. Practical constraints: If one optimal method proves infeasible (lack of expertise, time, resources), substitution may require additional methods:

- If Backcasting is too resource-intensive, might need Delphi (Valuative) + another Prescriptive method

3. Method strengths beyond ratings: Some methods offer valuable features not captured in ratings:

- **Delphi** provides expert validation across all outputs
- **Morphological Analysis** ensures systematic exploration of solution spaces

The 4-Method "Enhanced Coverage" Option

Backcasting + Futures Wheel + Trend Analysis + Delphi Method

Additional benefits:

- Delphi provides expert validation and consensus-building across the evaluation
- Redundant HIGH coverage for Valuative and Predictive (resilience)
- Balanced qualitative-quantitative approach
- Multiple stakeholder engagement opportunities

Alternative Optimization Objectives

The minimum-methods optimization assumes HIGH coverage is the only criterion.

Alternative objectives:

Objective 2: Maximize Average Rating Across All Questions

Instead of requiring HIGH in each category, maximize overall coverage quality.

Calculation: Average rating across all 25 cells (5 questions × 5 methods)

Why this matters: Generalist methods provide MEDIUM across many dimensions, which might serve evaluation needs better than focused HIGH ratings with LOW elsewhere.

Example: Adding Causal Layered Analysis (all MEDIUM or HIGH) provides consistent backup across all dimensions.

Objective 3: Minimize Maximum Gap

Ensure no question type is neglected in secondary coverage.

Example: While Solution B covers all types with HIGH, if Horizon Scanning is unavailable, Descriptive drops to MEDIUM (Futures Wheel). Solution A maintains HIGH even if one method unavailable.

Objective 4: Match Evaluation Budget/Timeline

Select methods where implementation effort matches available resources:

Rapid Assessment (2-3 months): Prioritize faster methods **Comprehensive**

Evaluation (12+ months): Can include resource-intensive methods

Practical Application Template

Decision Matrix Template

Users can complete this table to document their selection:

Criterion	Solution A	Solution B	Solution C	Selected Solution
Achieves minimum 3 methods	✓	✓	✓	
Provides redundancy	Yes (Predictive)	No	No	
Implementation complexity	Medium	Low	High	
Stakeholder engagement	Medium-High	High	Medium	
Sequential workflow	Strong	Strong	Strong	
Available expertise	[Assess]	[Assess]	[Assess]	
Available time	[Assess]	[Assess]	[Assess]	
Available budget	[Assess]	[Assess]	[Assess]	

Selection Justification Template

Selected Methods: [List]

Coverage Achievement:

- Descriptive: [Method] (HIGH)
- Valuative: [Method] (HIGH)
- Explanatory: [Method] (HIGH)

- Predictive: [Method] (HIGH)
- Prescriptive: [Method] (HIGH)

Rationale for Selection:

- [Why this combination over alternatives]
- [How methods complement each other]
- [How selection matches resources/constraints]

Contingency Plan:

- If [Method X] proves infeasible: [Alternative approach]

Software Implementation Concept

For larger method sets, this optimization could be automated:

INPUT:

- Method × Question matrix with ratings
- Required coverage level (HIGH, MEDIUM)
- Optional: method costs, compatibilities

ALGORITHM:

1. Identify forced selections (unique HIGH ratings)
2. Greedy set cover: repeatedly select method covering most uncovered questions
3. Verify minimum achieved
4. Generate alternative solutions with same minimum
5. Rank by secondary criteria

OUTPUT:

- Minimum number of methods required
- All optimal combinations
- Ranked recommendations based on criteria

Limitations and Caveats

What This Optimization Doesn't Consider:

- 1. Sequential dependencies:** Some methods work best after others (e.g., Trend Analysis before Futures Wheel)
- 2. Context specificity:** Optimal for "generic" evaluation; specific domains might shift preferences
- 3. Quality variation:** A method rated HIGH might be poorly implemented; MEDIUM well-implemented might be better
- 4. Stakeholder acceptability:** Some stakeholder groups may resist certain methods regardless of ratings
- 5. Learning effects:** Multiple similar methods might provide diminishing returns OR might provide validation through triangulation

When to Deviate from Optimum:

- **Organizational capacity:** Use familiar methods even if not optimal
- **Stakeholder buy-in:** Include methods stakeholders understand/trust
- **Time constraints:** Faster MEDIUM method might serve better than slow HIGH method
- **Triangulation:** Deliberately include multiple methods for same question type to validate findings

Summary Recommendation

For the 10-method matrix:

Minimum optimal solution: 3 methods

- Backcasting (required for Prescriptive)
- Futures Wheel (covers Explanatory + Predictive)
- Trend Analysis (covers Descriptive + provides Predictive redundancy)

Context-dependent selection:

- **Resource-constrained:** Solution B (Backcasting + Futures Wheel + Horizon Scanning) - simplest implementation
- **Technically capable:** Solution C (Backcasting + Cross-Impact Analysis + Trend Analysis) - quantitative emphasis
- **Balanced/uncertain context:** Solution A (Backcasting + Futures Wheel + Trend Analysis) - redundancy and workflow strength

Enhanced solution (if resources allow): 4 methods

- Add Delphi Method to any 3-method solution for expert validation and consensus-building across all evaluation dimensions

This systematic approach can be applied to any evaluation matrix, adjusted for different coverage requirements (e.g., "at least MEDIUM in all categories"), and extended with additional optimization criteria based on evaluation context.

Please tell me more about this: "This is a variant of the "set cover problem" from operations research, adapted for evaluation planning"

The Set Cover Problem and Method Selection Optimization

What is the Set Cover Problem?

The set cover problem is a classic optimization problem in computer science and operations research. Here's the fundamental structure:

The Classic Problem

Given:

- A **universe** of elements that need to be covered: $U = \{1, 2, 3, 4, 5\}$
- A **collection of sets**, each covering some elements:
 - Set A = {1, 2, 3}
 - Set B = {2, 4}
 - Set C = {3, 4, 5}
 - Set D = {1, 5}

Goal: Select the minimum number of sets that cover all elements in the universe.

Solution: Sets A and C together cover all elements {1, 2, 3, 4, 5} with just 2 sets.

Why It Matters

The set cover problem appears across many practical domains:

- **Facility location:** Where to place emergency services to cover all neighborhoods
- **Network design:** Which network nodes to monitor to observe all traffic
- **Sensor placement:** Where to place sensors to monitor an entire area
- **Resource allocation:** Which resources to deploy to address all needs

How Method Selection Maps to Set Cover

The Evaluation Planning Version

Universe to cover: The five evaluation question types

- $U = \{\text{Descriptive, Valuative, Explanatory, Predictive, Prescriptive}\}$

Collection of sets: The 10 foresight methods, each "covering" certain question types with HIGH ratings

Translation:

- Method = Set
- Question type covered with HIGH rating = Element in that set
- "Covered" = Has HIGH rating for that question type

Example Mapping

Backcasting covers: {Valuative, Prescriptive} **Futures Wheel covers:** {Explanatory, Predictive} **Trend Analysis covers:** {Descriptive, Predictive}

Question: What's the minimum number of methods needed to have HIGH coverage for all five question types?

Answer: 3 methods (e.g., Backcasting + Futures Wheel + Trend Analysis)

Important Differences from Classic Set Cover

The evaluation planning version has several distinguishing characteristics:

1. Multi-Level Coverage

Classic set cover: Binary (element either covered or not)

Evaluation version: Three levels (HIGH, MEDIUM, LOW)

This creates variations:

- **Strict version:** Require HIGH coverage for all question types (what we did)
- **Relaxed version:** Accept MEDIUM coverage for some question types
- **Weighted version:** Different question types have different importance

2. Small Problem Size

Classic applications: Often thousands or millions of elements and sets

- Example: Covering all addresses in a city with delivery zones

Evaluation version: 5 question types, 10-20 methods

- Small enough to solve by inspection or simple enumeration
- Doesn't require sophisticated algorithms

3. Multiple Optimal Solutions

In our case, several 3-method combinations achieve full HIGH coverage:

- Solution A: Backcasting + Futures Wheel + Trend Analysis
- Solution B: Backcasting + Futures Wheel + Horizon Scanning
- Solution C: Backcasting + Cross-Impact Analysis + Trend Analysis

This requires **secondary criteria** to choose among equally optimal solutions—something not typically part of classic set cover.

4. Additional Constraints and Preferences

Evaluation planning adds complexity:

- **Method compatibility:** Some methods work better together
- **Resource constraints:** Different methods have different costs
- **Sequential dependencies:** Some methods should precede others
- **Stakeholder acceptance:** Some methods may be culturally inappropriate
- **Redundancy preferences:** May want backup coverage
- **Expertise availability:** Can only use methods staff can implement

Classic set cover typically just minimizes the number of sets selected.

Computational Complexity

Why This Matters

The set cover problem is NP-complete. This means:

Technical definition: No known algorithm can solve all instances in polynomial time (time proportional to size^k for some constant k).

Practical meaning: For large problems, finding the guaranteed optimal solution becomes computationally infeasible as problem size grows.

What This Means for Method Selection

For evaluation planning:

- With 10 methods and 5 question types, the problem is trivially small
- Can enumerate all possible combinations if needed
- Maximum possible combinations: $2^{10} = 1,024$ subsets to check

- In practice, much fewer because we can eliminate obviously non-optimal choices

If scaling to larger method libraries:

- With 50 methods and 10 question types, exact optimization becomes harder
- Might need approximation algorithms (greedy approaches)
- Or use heuristics based on domain knowledge

The Greedy Algorithm

The classic approach to set cover uses a **greedy algorithm**:

1. Start with empty selection
2. Repeat until all elements covered:
 - Select the set that covers the most uncovered elements
 - Add to selection
 - Mark newly covered elements
3. Return selection

Applied to method selection:

Step 1: Uncovered = {Descriptive, Valuative, Explanatory, Predictive, Prescriptive}

Select method covering most types

Step 2: Find method with most HIGH ratings

- Scenario Planning: 1 HIGH (Predictive)
- Delphi: 2 HIGH (Valuative, Predictive)
- Futures Wheel: 2 HIGH (Explanatory, Predictive)
- Trend Analysis: 2 HIGH (Descriptive, Predictive)

Multiple ties at 2 HIGH. Pick Delphi.

Selected: {Delphi}

Covered: {Valuative, Predictive}

Uncovered: {Descriptive, Explanatory, Prescriptive}

Step 3: Find method covering most remaining types

- Backcasting: 1 HIGH (Prescriptive) [Valuative already covered]
- Futures Wheel: 1 HIGH (Explanatory) [Predictive already covered]
- Trend Analysis: 1 HIGH (Descriptive) [Predictive already covered]

Three-way tie. Forced to pick. Choose Backcasting for Prescriptive (since it's the only HIGH option for that type).

Selected: {Delphi, Backcasting}

Covered: {Valuative, Predictive, Prescriptive}

Uncovered: {Descriptive, Explanatory}

Step 4: Need to cover Descriptive and Explanatory

- Futures Wheel: 1 HIGH (Explanatory)
- Trend Analysis: 1 HIGH (Descriptive)

Need both since each covers only one remaining type.

Final Selection: {Delphi, Backcasting, Futures Wheel, Trend Analysis}

= 4 methods

Note: The greedy algorithm found a 4-method solution, not the optimal 3-method solution we found through systematic analysis. This demonstrates that **greedy doesn't guarantee the optimal minimum** for set cover.

Why Greedy Doesn't Always Work

The greedy algorithm provides an **approximation** but not always the optimal solution.

Example Where Greedy Fails

Consider this scenario:

Methods available:

- Method A covers: {Descriptive, Predictive}
- Method B covers: {Valuative, Explanatory}
- Method C covers: {Prescriptive}
- Method D covers: {Descriptive, Valuative, Explanatory, Predictive}

Greedy approach:

1. Selects Method D first (covers 4 types)
2. Still needs Prescriptive
3. Adds Method C
4. **Solution: 2 methods (D + C)**

But there's a problem if:

- Method D is actually infeasible (too expensive, no expertise)
- Would then need Methods A + B + C = 3 methods

Alternative optimal solution:

- Methods A + B + C = 3 methods
- More robust if any single method fails

This is why we need to consider **multiple optimal solutions** and secondary criteria.

Practical Application Framework

Given the characteristics of evaluation planning, here's a practical approach:

For Small Method Libraries (≤ 20 methods)**Use systematic enumeration:**

1. Identify forced selections (question types with only 1 HIGH option)
2. For each forced selection, identify what remains uncovered
3. Systematically test combinations of remaining methods
4. Identify all minimum-cardinality solutions
5. Rank by secondary criteria

This is what we did for the 10-method matrix.

For Medium Libraries (20-50 methods)

Use guided search:

1. Start with forced selections
2. Group methods by coverage patterns (how many types they cover)
3. Prioritize multi-coverage methods
4. Use greedy as starting point
5. Test variations around greedy solution
6. Verify minimum through targeted search

For Large Libraries (50+ methods)

Use approximation with validation:

1. Use greedy algorithm for initial solution
2. Apply local search (try swapping methods)
3. Use domain knowledge to identify promising combinations
4. Accept near-optimal solutions with better secondary criteria
5. Document that solution is "at most X% above optimal"

Extensions and Variations

The basic set cover framework can be extended for evaluation planning:

1. Weighted Set Cover

Scenario: Different question types have different importance.

Modification:

- Assign weights to question types: $w(\text{Descriptive}) = 1$, $w(\text{Predictive}) = 3$
- Objective: Minimize number of methods while ensuring high-weight types covered

Example: If Predictive is most important (weight = 5) and others (weight = 1), might accept:

- HIGH coverage for Predictive
- MEDIUM coverage for others
- Using fewer methods

2. Budgeted Set Cover

Scenario: Each method has a cost (time, money, expertise).

Modification:

- Assign costs: $c(\text{Delphi}) = \$10\text{K}$, $c(\text{Trend Analysis}) = \3K
- Constraint: Total cost \leq budget
- Objective: Maximum coverage within budget

Example: With \$15K budget:

- Can't afford all optimal 3-method solutions
- Must choose cheaper methods even if need more of them
- Might accept MEDIUM coverage to stay in budget

3. Maximum Coverage Problem

Reverse scenario: Given a fixed number of methods (say 3), maximize coverage.

Modification:

- Constraint: Select exactly 3 methods
- Objective: Maximize number of HIGH ratings achieved (not just question types covered)

Example: Must select 3 methods:

- Want as many HIGH ratings as possible across all 15 cells (3 methods \times 5 questions)
- Might select methods with multiple HIGH ratings even if some question types get no HIGH coverage

4. Multi-Objective Set Cover

Scenario: Optimize for multiple criteria simultaneously.

Objectives:

- Minimize number of methods (primary)
- Minimize total cost (secondary)
- Maximize stakeholder engagement potential (tertiary)

Approach: Pareto frontier analysis

- Identify solutions where improving one objective requires sacrificing another
- Present decision-maker with trade-off options

Connection to Other Optimization Problems

The evaluation method selection problem relates to several other classic problems:

1. Knapsack Problem

Scenario: Limited capacity (budget, time), items with values (coverage) and weights (costs).

Question: Which methods to include to maximize coverage value within capacity constraint?

2. Assignment Problem

Scenario: Assign methods to question types to maximize coverage quality.

Different from set cover: Each question type gets exactly one primary method assigned.

3. Scheduling Problem

Scenario: Methods must be sequenced (Trend Analysis before Futures Wheel).

Extension: Find minimum set that can be implemented in required sequence within time constraints.

Practical Implementation Tool Concept

A spreadsheet or simple software tool could automate this:

Input Sheet

Method	Descriptiv e	Valuativ e	Explanator y	Predictiv e	Prescriptiv e	Cos t	Time (weeks)
Delphi	LOW	HIGH	LOW	HIGH	MEDIUM	10K	8
Backcastin g	LOW	HIGH	MEDIUM	LOW	HIGH	15K	12
...

Algorithm Settings

- Required coverage level: [HIGH / MEDIUM]
- Budget constraint: [\$X]
- Time constraint: [Y weeks]
- Must include methods: [user-specified]
- Must exclude methods: [user-specified]

Output

Minimum Solution:

- Number of methods required: 3
- All optimal combinations: [A+B+C], [A+B+D], [A+C+D]

Ranked by Secondary Criteria:

1. Solution [A+B+C]: Cost \$25K, Time 16 weeks, Redundancy score: 2
2. Solution [A+B+D]: Cost \$22K, Time 18 weeks, Redundancy score: 1
3. ...

Sensitivity Analysis:

- If Budget reduced to \$20K → Need 4 methods
- If Time reduced to 12 weeks → Can't achieve full HIGH coverage

Why This Mathematical Framing Is Valuable

Understanding the set cover connection provides several benefits:

1. Conceptual Clarity

Recognizing this as a well-studied problem helps:

- Frame the decision systematically
- Know what solution characteristics to expect
- Understand when simple vs. sophisticated approaches needed

2. Knowing Computational Limits

Understanding NP-completeness means:

- Small problems: Exact solution feasible
- Large problems: Need approximations
- Can estimate effort required for different problem sizes

3. Leveraging Existing Knowledge

Decades of operations research provides:

- Proven algorithms
- Known approximation guarantees
- Software libraries

- Best practices

4. Communicating Trade-offs

The framework helps explain to stakeholders:

- Why multiple solutions exist
- What trade-offs are being made
- How constraints affect feasibility
- Why certain methods must be included

5. Extending the Approach

Understanding the underlying problem structure enables:

- Adding new constraints systematically
- Adapting to different optimization objectives
- Scaling to larger method libraries
- Integrating with other planning tools

Limitations of the Mathematical Model

The set cover framing is powerful but has boundaries:

What It Captures Well

- Minimum number of methods needed
- Coverage requirements
- Multiple optimal solutions
- Cost and resource constraints

What It Doesn't Capture

Qualitative factors:

- Cultural appropriateness of methods
- Stakeholder trust and buy-in
- Organizational learning from methods
- Political feasibility

Dynamic factors:

- Learning during implementation might reveal new needs

- Context changes might shift priorities
- Method effectiveness might vary with implementation quality

Interaction effects:

- Synergies between methods (1+1=3)
- Conflicts between methods (1+1=1)
- Sequential dependencies
- Triangulation benefits

When to Use Mathematical vs. Judgment-Based Approaches

Use mathematical optimization when:

- Large method libraries make exhaustive consideration difficult
- Objective constraints (budget, time) are clear and binding
- Need to justify decisions to external stakeholders
- Multiple evaluators need consistent selection framework

Use judgment-based approaches when:

- Small method set allows intuitive comparison
- Qualitative factors dominate
- Context is unique with limited precedent
- Flexibility and adaptation are priorities

Use hybrid approaches (recommended):

- Mathematical optimization identifies candidate solutions
- Expert judgment applies contextual factors
- Iterate between analysis and deliberation

Summary

The evaluation method selection problem is indeed a variant of the set cover problem, but with important adaptations:

Core similarity: Finding minimum collection of sets (methods) that cover all elements (question types with HIGH ratings)

Key differences:

- Multi-level coverage (HIGH/MEDIUM/LOW not just binary)

- Small problem size (tractable with simple approaches)
- Rich secondary criteria beyond just minimizing number
- Additional constraints from practical context

Practical value:

- Provides systematic framework for method selection
- Helps identify all optimal solutions
- Enables clear communication of trade-offs
- Scales to larger method libraries if needed

Limitations:

- Doesn't capture all qualitative factors
- Requires supplementation with expert judgment
- Assumes ratings are accurate and context-independent

For your anthology chapter, mentioning this connection demonstrates the intellectual rigor of the approach while also signaling that it's grounded in established optimization theory. It positions the framework as both theoretically sound and practically applicable.