1 Mathematical Formulation of the Two-Stage Stochastic Model

Source

This formulation is adapted from: Smeulders, B. M. L., Bartier, V., Crama, Y., & Spieksma, F. C. R. (2022). Recourse in Kidney Exchange Programs. *INFORMS Journal on Computing, 34*(2), 1191–1206. `https://doi.org/10.1287/ijoc.2021.1099`

1.1 Notation

- $G = (V, A)$: Directed compatibility graph.
- V: Set of patient-donor pairs.
- A: Set of possible donation arcs.
- B: Testing budget (maximum number of arcs to test).
- p_{ij}: Probability that arc (i, j) passes crossmatch test.
- S: Set of scenarios, each corresponding to a possible realization of arc outcomes.
- q_s: Probability of scenario s.
- C_s: Set of feasible cycles in scenario s.
- H_s: Set of feasible chains in scenario s.
- $a_{s,\ell}$: Number of transplants in cycle/chain ℓ in scenario s.
- K, L: Maximum cycle and chain lengths.
- β_{ij}: Binary decision variable for testing arc (i, j).
- $x_{s,\ell}$: Binary decision variable for selecting cycle/chain ℓ in scenario s.

1.2 Cycle and Chain Definitions

Let C_s denote the set of feasible cycles and H_s denote feasible chains in scenario s, respecting:

$$|c| \leq K, \quad |h| \leq L.$$

Here, K and L are pre-specified maximum cycle and chain lengths.

1.3 Assumptions

- Crossmatch test results are independent across arcs.
- Testing budget B is fixed and known in advance.
- Probabilities p_{ij} are deterministic and known.
- Cycle and chain lengths are bounded by K and L.

1.4 Objective Function

$$\boxed{\max_{\beta,x} \quad \sum_{s\in S} q_s \sum_{\ell\in C_s\cup H_s} a_{s,\ell}\, x_{s,\ell}} \tag{1}$$

1.5 Constraints

$$\sum_{(i,j)\in A} \beta_{ij} \le B, \tag{2}$$

$$x_s \in P(G_{s,\beta}), \quad \forall s \in S, \tag{3}$$

$$x_{s,\ell} \in \{0,1\}, \quad \forall s,\ \forall \ell, \tag{4}$$

$$\beta_{ij} \in \{0,1\}, \quad \forall (i,j) \in A, \tag{5}$$

$$\sum_{\ell\in Q(s,i,j)} x_{s,\ell} \le \beta_{ij}, \quad \forall s,\ \forall (i,j) \in A_s. \tag{6}$$

1.6 Scenario Generation

Scenarios are generated using Sample Average Approximation (SAA), where:

$$S = \{s_1, s_2, \ldots, s_N\}$$

Each scenario s is a realization of test outcomes generated according to arc probabilities p_{ij}. Scenario probability q_s is computed as:

$$q_s = \prod_{(i,j)\in A_s} p_{ij} \prod_{(i,j)\notin A_s} (1 - p_{ij}).$$

1.7 Description

Equation (1) maximizes the expected number of transplants over all scenarios, subject to:

- (2): Testing budget constraint.
- (3): Feasibility of cycles/chains given tested arcs.
- (4), (5): Binary nature of decision variables.
- (6): Only arcs that are tested and pass can be used in cycles/chains.

This model explicitly captures the two-stage nature of the problem:

- **First Stage:** Decide which arcs to test under budget constraints.
- **Second Stage:** Given realized test outcomes (scenarios), determine feasible cycles and chains to maximize transplants.

2 Computational Implementation

2.1 Overview

The problem is solved using a structure-based two-stage stochastic mixed-integer programming approach. The implementation consists of the following steps:

- Random instance generation for patient-donor compatibility graph.
- Scenario generation using Sample Average Approximation (SAA).
- Cycle and chain generation respecting length constraints.
- Structure-based optimization for arc testing selection and scenario-specific cycle/chain choices.
- Fallback to arc-matching if no feasible cycles/chains exist in any scenario.

2.2 Algorithm Outline

1. **Instance Generation:** Generate a random directed graph with given number of vertices and arc probability. Assign success probabilities p_{ij} to each arc.

2. **Scenario Generation:** Generate N scenarios by sampling arcs according to p_{ij}.

3. **Cycle and Chain Generation:** Generate all feasible cycles up to length K and chains up to length L starting from non-directed donors (NDDs), ensuring arc inclusion in A.

4. **Optimization:** Solve the two-stage stochastic problem using Mixed Integer Programming (MIP):

   - First stage: select arcs to test within budget B.
   - Second stage: choose feasible cycles/chains for each scenario maximizing expected transplants.

5. **Fallback:** If no feasible cycle/chain exists, revert to arc-matching optimization.

2.3 Solver Implementation

The solver is implemented in Python using the `mip` package with the following logic:

- Variables: β_{ij}, $x_{s,\ell}$.
- Constraints: testing budget, arc feasibility, vertex disjointness per scenario.
- Objective: maximize expected number of transplants plus a small tie-breaker term.
- Scenarios are equally weighted.

2.4 Experimental Setup

The computational experiments vary cycle and chain length $K = L$ from 2 to 5, and repeat runs for averaging:

- Number of vertices: 10.

- Arc probability: 0.3.
- Number of scenarios: 100.
- Budget: $\frac{|V|^2}{3}$.

Results are collected for:

- Objective values (expected transplants).
- Computational time.

3 Outputs

The `solve_selection_problem` function returns:

1. `selected_arcs`: arcs chosen for testing.
2. `chosen_structs_per_scenario`: selected structures (cycles/chains) for each scenario.
3. `status`: solver status (`OPTIMAL`, `FEASIBLE`, `INFEASIBLE`).
4. `obj`: objective value (expected number of transplants).

3.1 Sample Execution

Executed in __main__:

- Input: a directed graph with 10 nodes, budget of 1.5, maximum cycle length $K = 3$, maximum chain length $L = 3$, and NDD (0,1), 2 scenarios.
- Output: list of selected arcs and structures for several scenarios.

To assess model scalability and the impact of graph size on output, tests were conducted on graphs of sizes 10, 15, 20, 25, and 30 nodes. In all scenarios, the budget was set to 1.5, the maximum cycle length to 3, and the maximum chain length to 3. The objective (expected number of transplants) was computed for each graph and plotted.

Results show that as the number of nodes increases, the objective value generally grows. However, this growth is sub-linear due to budget constraints. Thus, in larger graphs, capacity for potential matches increases, but budget limitations impose a direct dependency on the number of arcs tested.

4 Sensitivity Analysis on K and L

4.1 Goal of the Analysis

The parameters K (maximum cycle length) and L (maximum chain length) in the Kidney Exchange Model determine the maximum size of feasible structures. Small values for these parameters limit the search space and simplify the model, but may result in fewer patients being included in cycles or chains, leading to a lower number of transplants. In contrast, larger values of K and L allow for longer cycles and chains, which can lead to an increase in the number of transplants; however, this significantly increases the computational complexity of the model and lengthens the solution time. This experiment was conducted to investigate the effect of changing the values of K and L on the **solve time** and the **objective value** (expected number of transplants).

4.2 Methodology

- $K = L$ values were varied from 2 to 5.
- For each value, the model was solved 5 times on a random graph to mitigate the effect of randomness-induced fluctuations.
- In each run:
  - The solution time from the start to the end was recorded.
  - The model's objective value was logged.
- Finally, the average times and average objective values were calculated for each $K = L$ value.
- The results were presented in a chart with two curves: one for the solve time and one for the objective value.

4.3 Results and Observations

1. **Objective Value**:
   - When $K = L = 2$, the model only considers short cycles and chains; consequently, exchange opportunities are limited, and the number of transplants is lower.
   - As $K = L$ increases to 3 and beyond, the model can form longer cycles and chains. This allows more patients to participate in the exchange, and the objective value increases.

- The growth of the objective value is typically **diminishing**; for example, there is a significant increase from 2 to 3, but the increase from 4 to 5 is smaller.

2. **Solve Time**:

   - At $K = L = 2$, the model is relatively small and solves quickly[cite: 131].
   - As K and L increase, the number of potential cycles and chains increases **exponentially**. This directly expands the size of the model (number of variables and constraints).
   - Consequently, the solve time increases rapidly. For some values, the solve time may multiply while the increase in the objective value is very limited.

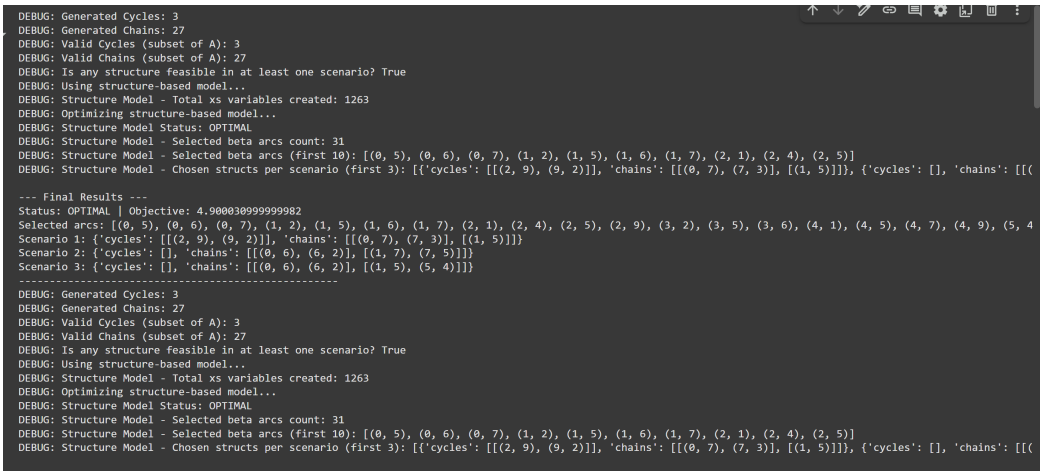

Figure 1: Solve Time and Objective Value vs. $K = L$

5 Sensitivity Analysis on Parameters K and L

5.1 Goal of this Analysis

The **K** and **L** parameters (**K** is the loop length, **L** is the maximum chain length) can determine the allowed structures in the entire exchange space. A small value for these parameters reduces the restricted exchange space, making the search and simplification easier. However, this may prevent the

consideration of longer loops and chains, which could lead to an increase in the number of exchange bonds. This significantly increases the model's complexity.

This experiment aims to study the effect of changing the **K** and **L** values on the **Time to Solution** and the **Objective Value** (Expected Number of Bonds).

5.2 Execution

- The **L** value is set from 2 to 5.
- For each **L** value, the model is run **5 times** on a random input to reduce the effect of stochastic fluctuations.
- In each run:
  - The **time to solution** is obtained from the start of the solution until the end.
  - The **objective value** is recorded.
- At the end, the mean of the solution times and the mean **objective value** for each **L** value is calculated.
- The results are presented in the form of two diagrams:
  - First for the **Time to Solution**.
  - Then for the **Objective Value**.

5.3 Results and Observations

5.3.1 Objective Value

- When $\mathbf{K = 2}$ and $\mathbf{L = 2}$, the model only considers short loops and chains. Therefore, the number of limited exchanges is low.
- By increasing **L** to **3** and greater, it becomes possible to form longer loops and chains. This leads to a greater number of allowed exchanges in the entire exchange system and an increase in the objective value.
- The growth of the **objective value** usually **decreases**, meaning that the increase from 4 to 5 is smaller than the increase from 2 to 3.

5.3.2 Solve Time

- When $\mathbf{L} = \mathbf{2}$, the model is very small and is solved quickly.
- With an increase in **K** and **L**, the number of loops and chains increases, which directly affects the model by increasing the number of variables and constraints.
- Therefore, the solution time increases with a sharp slope. In some values, the solution time might be several times greater, while the increase in the objective value is very small.

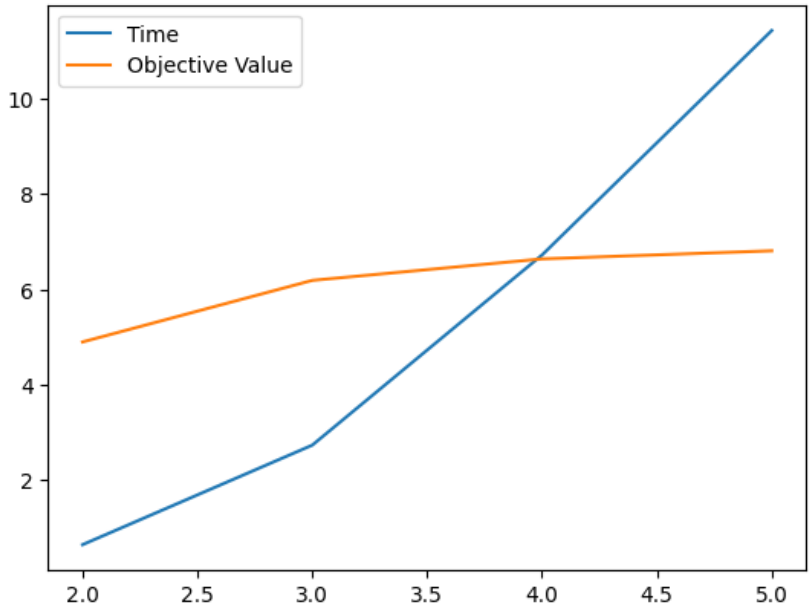

Figure 2: Plot of Solve Time and Objective Value versus Parameter L

6 Analysis and Summary

This experiment shows that there is a trade-off between the quality of the answer and the computational cost. High **K** and **L** values provide a higher quality answer but significantly increase the solution time. Considering the small increase in the number of bonds (Objective Value) when the **K** and **L** values are increased, and consequently the significant increase in the number of allowed exchanges, the computational complexity of the model increases,

and the time to solution grows sharply. Therefore, attention to both aspects of choosing appropriate **K** and **L** values is essential to control the speed and computational constraints.

7 Repository

The full Python implementation is available in the project repository.