Application of evolutionary algorithms to optimization of Boolean models in biomedicine

Bingyu Jiang *[1*], Pierre Klemmer[2], and Marek Ostaszewski[2]

[1]Department of Mathematics, University of Luxembourg
[2]Luxembourg Centre for Systems Biomedicine, University of Luxembourg

Abstract

Boolean networks have become essential tools for modeling gene regulatory systems and understanding cellular decision-making processes, but their optimization for biological relevance remains challenging. This study presents a comprehensive benchmark comparison of three prominent Boolean network optimization methodsCaspo, MEIGO, and CellNOpt (ILP and GA)evaluating their performance across structural robustness, method reliability, and biological relevance using Mean Squared Error (MSE) as the primary optimization criterion. Through systematic analysis of network reconstruction under varying perturbation levels (10-90

1 Introduction

Boolean models have emerged as a powerful framework for understanding complex biological systems, particularly in the context of gene regulatory networks. Originally introduced by Kauffman [7] in the 1960s, Boolean networks have been extensively applied to model genetic regulatory systems, providing insights into cellular decision-making processes and disease mechanisms. Boolean networks represent biological entities (genes, proteins, or other molecular species) as binary variables that can exist in either active or inactive states. The temporal evolution of these states is governed by Boolean functions that define how the state of each node depends on the states of its regulatory inputs.

As Boolean networks have become increasingly important in systems biology, numerous algorithms have been developed to optimize these models for better biological relevance and predictive accuracy. The goal of this project is to investigate whether structure-based optimization using experimental data leads to functionally equivalent networks in terms of dynamical behavior. We compare three prominent optimization methods–Caspo [5], MEIGO [2], and CellNOpt [4]–using Mean Squared Error (MSE) as our primary metric.

These optimization methods address a fundamental biological question: Given a proposed network structure, which parts of it are actually relevant under specific experimental conditions? Real biological networks are inherently context-dependentthe same cell might harbor hundreds of potential regulatory connections, but only a subset remain active under specific conditions

*Corresponding author: bingyu.jiang9185@gmail.com

such as drug treatment, disease states, or developmental stages. This context-dependency necessitates sophisticated optimization approaches that can identify the most relevant network components for given experimental scenarios.

Through systematic comparison of these methods, our investigation focuses on three critical aspects of optimization method performance:

First, Robustness Testing: How sensitive are these optimization methods to errors in the input network topology? This question is particularly relevant given that prior knowledge networks (PKNs) used as starting points often contain incomplete or inaccurate information about regulatory relationships.

Second, Method Reliability: Do different algorithms converge to similar solutions when starting from slightly different network topologies? Consistency across multiple runs and minor variations in input conditions is essential for generating reliable biological insights.

Third, Biological Relevance: Do the optimized networks capture known biological interactions and pathways? The ultimate validation of any optimization method lies in its ability to preserve biologically meaningful relationships while eliminating spurious connections.

The detailed algorithmic descriptions can be found in the supplementary document (table 2 shows a summary of these methods). This essay focuses primarily on the experimental comparison and validation of these optimization approaches using real biological data.

2 Benchmarking

Benchmarking provides essential quantitative validation of Boolean model optimization methods. This section, which follows the procedure in [6], presents a systematic framework for comparative analysis of different approaches.

The primary objective of Caspo, MEIGO, and CellNOpt is to address a fundamental question in systems biology: Given a proposed network structure, which components are actually relevant under specific experimental conditions? This challenge is particularly important because real biological networks are inherently context-dependentwhile a single cell may contain hundreds of potential regulatory connections, only a subset remains active under particular conditions such as drug treatment, disease states, or developmental stages.

Furthermore, experimental measurements and data collection processes introduce various biases, and prior knowledge networks (PKNs) are often incomplete, noisy, or even incorrect. Therefore, optimization methods must be capable of identifying the most relevant subnetwork that explains observed data while filtering out irrelevant or spurious connections.

The goal of this benchmark is to investigate whether structure-based optimization using experimental data leads to functionally equivalent networks in terms of dynamical behavior. A critical question we address is: How many errors in prior knowledge networks can be tolerated before the optimized network fails to reproduce observed dynamics?

2.1 Benchmark Datasets

There are three types of data:

1. **Toy datasets**: Derived from CellNOptR packages, providing controlled test cases with known ground truth

2. **Dream datasets**: Obtain from DREAM3 challenge [1], offering standardized benchmarking scenarios

[1] `https://gnw.sourceforge.net/dreamchallenge.html`

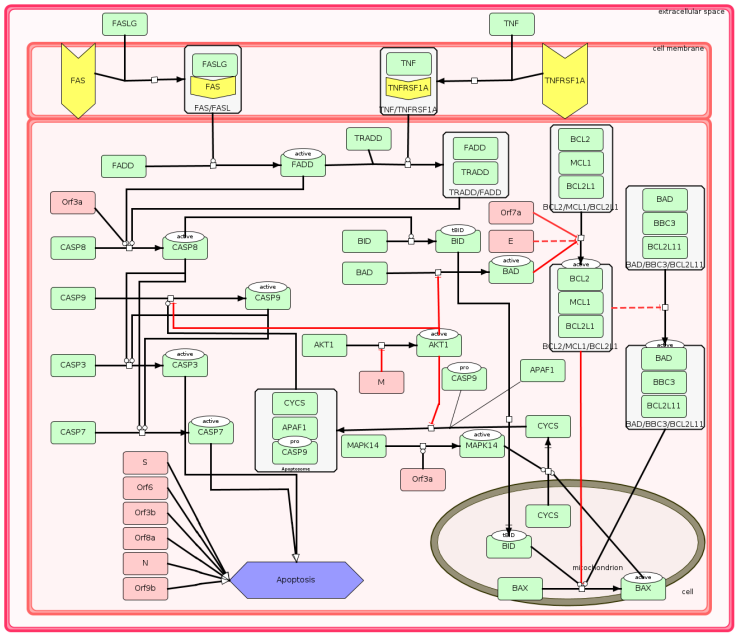

Figure 1: Example of a complex regulatory network: COVID-19 apoptosis submap. Taking from [3].

3. **Real datasets**: PKN obtained from the MINERVA Platform [3] in CellDesigner SBML format (transformed into SBML-qual format using the CaSQ tool [1]) or Cell Collective [2] in SBML-qual format. Note that experimental data are not available for these real networks.

2.1.1 MIDAS Data Structure

Standardized experimental data formats are crucial for Boolean network modeling and signal transduction studies. The MIDAS (Minimum Information for Data Analysis in Systems Biology) format provides a structured framework for organizing stimulus-response experimental data in tab-separated or comma-separated format.

The MIDAS structure captures complex stimulus-response relationships across multiple experimental conditions and time points through standardized column types.

- **Treatment columns (TR:)**: Input stimuli applied to the system, e.g., TNFa (binary: 1 = present, 0 = don't care)
- **Inhibitor columns (TR:) with suffix 'i'**: Pharmacological inhibitors, e.g., PI3K (binary: 1 = present, 0 = don't care)
- **Inhibitor columns (DA:)**: Time step
- **Readout columns (DV:)**: Measured protein activities or phosphorylation states

2.2 Evaluation Metrics

The comparison of optimization approaches relies on three primary categories of metrics.

[2] `https://research.cellcollective.org/`

Table 1: Dataset summaries

	Toy dataset	**DREAM dataset**	**T-Cell Data**
Network summary			
Nodes (species)	15	40	40
Reactions (edges)	16	58	52
Activating edges	16	58	52
Inhibitory edges	1	2	5
Experimental data summary			
Cues (inputs measured)	4	8	6
Stimuli applied	2	4	3
Inhibitors applied	2	4	3
Signals read out	7	7	8
Time-points per experiment	2	2	2
Total experiments (rows)	9	25	17
Attractor summary			
Number of Attractors	4	16	10

- **Attractors Based metrics**: The set of attractors obtained via BoolNet [8]. Compare two matrix similarities using Jaccard[3], Hamming distance, and F1 score.
- **Graph Based Metrics**: The Jaccard similarity in node and edge level.
- **Performance Metrics**: The running time and the MSE.

Mean Square Error The mean squared error (MSE) is the average of the squares of the errors between the estimated values and the true value:

$$MSE = \frac{1}{N} \sum_{i=1}^{N_{sig}} \sum_{j=1}^{N_{stim}} \sum_{k=1}^{N_{inhib}} (x_{i,j,k}^{pred} - x_{i,j,k}^{obs})^2$$

where N is the total number of data points, N_{sig} is the number of protein signals measured, N_{stim} is the number of stimuli, N_{inhib} is the number of inhibition conditions used between model predictions and experimental data.

Hamming distance For two strings of equal length (or binary vectors) $a, b \in \{0, 1, \text{NaN}\}^n$, the Hamming distance is the number of positions where they differ:

$$\text{Ham}(a, b) = \sum_{i=1}^{n} \mathbf{1}\{a_i \neq b_i\}$$

It is defined only when the lengths are equal and is computed in $O(n)$ time.

Note on Implementation: Since attractors calculated from original and candidate networks may have different dimensions, more sophisticated distance measures such as the longest common subsequence (LCS) or edit distance might be appropriate. However, our implementation introduces *NaN* values for missing entries, and the experimental results do not show any clear performance difference. For computational efficiency, we employ the simpler Hamming distance with *NaN* handling.

[3] `https://en.wikipedia.org/wiki/Jaccard_index`

Jaccard similarity For two sets A, B (or two binary attribute vectors), the Jaccard similarity index is

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

When $A, B \in \{0,1\}^n$, define the contingency counts.

$$M_{11} = \#\{i : A_i = 1,\ B_i = 1\}, \qquad M_{10} = \#\{i : A_i = 1,\ B_i = 0\},$$
$$M_{01} = \#\{i : A_i = 0,\ B_i = 1\}, \qquad M_{00} = \#\{i : A_i = 0,\ B_i = 0\},$$

so $J(A,B) = \dfrac{M_{11}}{M_{11} + M_{10} + M_{01}}$. The Jaccard distance is $1 - J(A,B)$. In set language: M_{00} are elements that belong to neither set. They do not affect either the intersection or the union, so they play no role in the Jaccard similarity. If we consider M_{00} to also be true, Jaccard degenerates in the Hamming similarity.

Precision, recall and F1 score Given true positives (TP), false positives (FP), and false negatives (FN), precision and recall are

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{M_{11}}{M_{11} + M_{01}}, \qquad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{M_{11}}{M_{11} + M_{10}}.$$

The F1 score is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\ \text{TP}}{2\ \text{TP} + \text{FP} + \text{FN}} = \frac{2M_{11}}{2M_{11} + M_{01} + M_{10}}.$$

These metrics provide complementary perspectives on the performance of the optimization method, enabling a comprehensive evaluation of both topological accuracy and dynamical fidelity.

2.3 Experimental Setup

Our experimental design follows a systematic five-step validation workflow to assess the robustness of Boolean network optimization methods. This approach enables a comprehensive evaluation of how well different algorithms perform when faced with imperfect prior knowledge networks.

Step 1: Synthetic Experimental Data Generation For data sets lacking experimental measurements, we developed a systematic approach to generate realistic synthetic data using the BoolNet R package. Our data generation pipeline employs two key functions:

1. **Experimental Condition Simulation**: We use *BoolNet::fixGenes* to simulate experimental perturbations by specifying which genes are overexpressed or destroyed, mimicking common laboratory interventions such as gene overexpression or RNA interference.

2. **Time Series Generation**: We apply *BoolNet::generateTimeSeries* with carefully calibrated parameters:

   - $numberSeries = 1000$: Generates sufficient statistical samples for robust analysis
   - $numberMeasurements = 50$ or $6 \times \#Nodes$: Scales the temporal resolution based on the network size to balance computational efficiency with temporal resolution

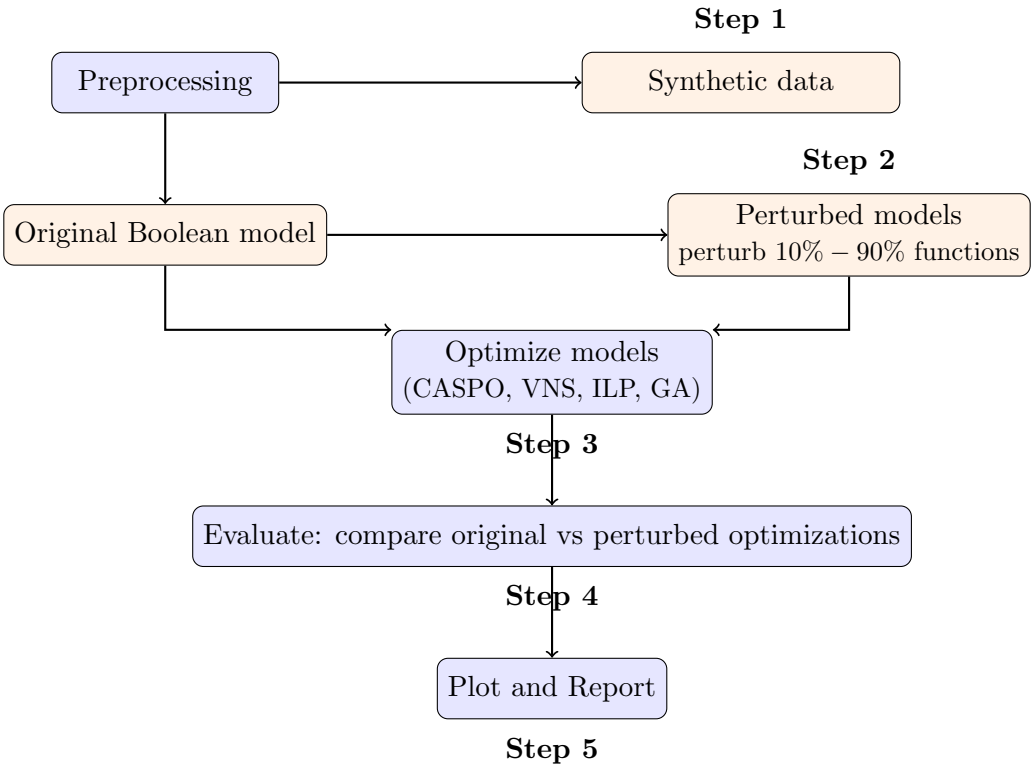

Figure 2: Optimized internal validation workflow for testing robustness of optimization methods

Data Processing Protocol: After simulating network dynamics for 50 time steps to reach steady-state behavior, we sampled the resulting distribution (dimensions: $1000 \times \#Genes$) to extract values of dependent variable (DV). This approach ensures that our synthetic data capture the essential dynamical properties of the underlying Boolean network while providing sufficient statistical power for optimization method comparison.

Methodological Consideration: Although synthetic data cannot fully capture the complexity of real biological measurements, they maintain crucial connections to the original PKN structure. This controlled environment is sufficient for our primary objective: testing the tolerance of optimization methods to network perturbations.

Step 2: Perturb Boolean network To evaluate method robustness, we systematically introduce controlled errors into the original prior knowledge networks using four distinct perturbation strategies:

1. **Change source/target**: Random reassignment of regulatory relationships by selecting new source and target nodes, simulating errors in interaction identification.

2. **Flip sign or direction**: for a given edge, flip its sign (positive -¿ negative regulation or vice versa), reverse its direction ($A->B$ becomes $B->A$), or do both.

3. **Rewire**: Random reconnection of regulatory paths while preserving overall network connectivity. For example, a linear cascade $A->B->C$ is transformed to $A->C->B$, maintaining the total number of connections but altering the patterns of the information flow.

4. **Split**: Division of single nodes into multiple components with appropriate edge redistribution to maintain network integrity and biological plausibility.

To ensure biological realism and network functionality, we enforce several constraints:

- **Input Preservation**: Source nodes representing external stimuli remain unmodified
- **Connectivity Maintenance**: Operations that would isolate nodes (zero incoming edges) are prohibited
- **Cycle Prevention**: Rewiring operations that create simple two-node cycles ($A->B->A$) are forbidden to avoid trivial oscillatory behavior.

Step 3: Optimization Recognizing that increased perturbation levels expand the optimization search space and computational complexity, we implemented an adaptive parameter scaling approach. The parameters of each optimization method are dynamically adjusted based on the perturbation percentage, balancing computational efficiency with solution quality.

Parallelization Strategy: While parallelization is essential for scaling to larger networks, several technical challenges emerge:

- **Method Heterogeneity**: Runtime varies significantly among optimization approaches
- **Sequential Dependencies**: Some methods, particularly Integer Linear Programming (ILP), do not support parallel execution
- **Cross-Platform Integration**: Methods implemented in different languages (R vs. Python) require careful interface management

Step 4: Evaluation Perturbed networks often exhibit dramatically different dynamical complexity compared to original models. Although original networks typically have fewer than 20 attractors, perturbed networks can generate more than 1000 attractors, creating significant comparison challenges.

Sampling-Based Comparison: For each of the 10 experimental replicates per dataset, we implement a fair comparison protocol:

- **Random Attractor Selection**: Sample equal numbers of attractors from both original and perturbed networks
- **Optimal Matching**: Apply the Hungarian algorithm to find optimal attractor pairings that minimize overall distance
- **Metric Computation**: Calculate evaluation metrics on matched attractor pairs

We acknowledge several limitations in our current evaluation approach. Although padding smaller attractor sets to match larger ones (creating 1000×1000 matrices) is possible, the $O(n^3)$ complexity of the Hungarian algorithm makes this computationally prohibitive. Investigation of attractor importance through basin size analysis revealed uniform distributions, which do not provide discriminatory power. Our results reveal inherent tensions between data-fit accuracy, graphical structure preservation, and attractor stability.

Computational Environment

- **Software Versions**: All methods were executed using the latest stable releases of their respective software packages, with version numbers documented for reproducibility.
- **Hardware Configuration**: All benchmarks were executed on AION HPC environment[4].
- **Random Seed Control**: Fixed random seeds across all stochastic methods to ensure consistent comparative results.

[4] `https://hpc-docs.uni.lu/systems/aion/`

2.4 Results

We conducted comprehensive experiments on three distinct datasets. Toy, DREAM, and T-Cell [5]. Each experiment was replicated 10 times to ensure statistical reliability, and the results were reported as averages across all runs. Fixed seeds across all stochastic methods ensured consistent comparative results.

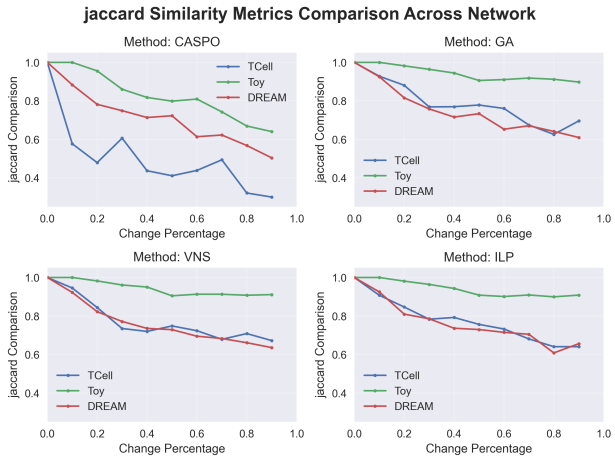

Figure 3: Jaccard similarity comparison between Toy, DREAM and T-Cell datasets.

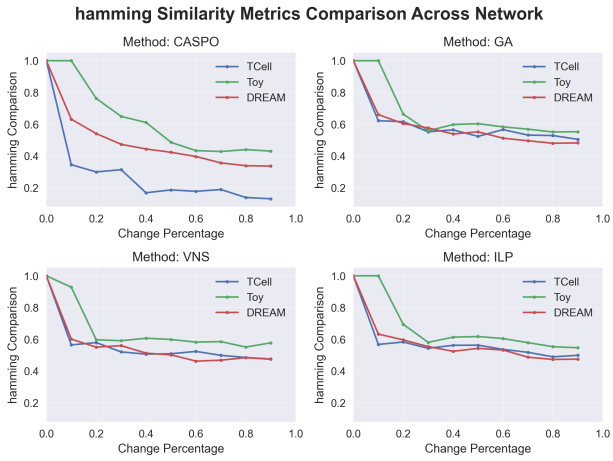

Figure 4: Hamming distance comparison between Toy, DREAM and T-Cell datasets.

The comparative analysis reveals a clear decreasing trend in performance as the perturbation percentage increases across all methods. Furthermore, as the size of the network increases, the performance deteriorates consistently. These trends are evident in the edge-level Jaccard similarity analysis 15, although they are not reflected in comparisons of the F1 score 16.

For Hamming similarity specifically, GA (Genetic Algorithm), VNS (Variable Neighborhood Search), and ILP (Integer Linear Programming) demonstrate comparable performance characteristics. This suggests that the Hamming distance may lack sufficient sensitivity to distinguish performance differences between data sets of varying sizes.

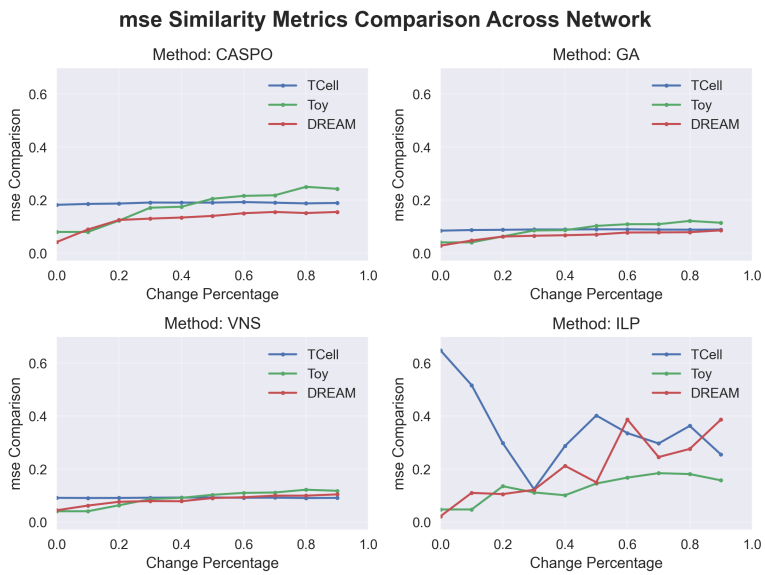

Figure 5: MSE comparison between Toy, DREAM and T-Cell datasets.

[5]Code is available at `https://gitlab.com/uniluxembourg/lcsb/BioCore/disease_maps/machine-learning-for-boolean-networks/genetic-algorithms-bingyu-jiang-internship/`

MSE analysis reveals several important patterns:

- CASPO consistently exhibits the highest error rates across all perturbation levels
- ILP demonstrates a clear increasing trend in MSE with perturbation intensity, aligning with theoretical expectations
- ILP shows exceptionally high MSE values at zero perturbation for the T-Cell dataset, with improvement as network perturbation increases

This counterintuitive behavior in the T-Cell dataset suggests that the synthetic experimental data may not accurately represent the true structure of the prior knowledge network.

The MSE values varied only slightly between CASPO, GA, and VNS. Although the absolute scale of MSE does not fundamentally change algorithmic behavior - all algorithms minimize the same loss function - it does affect sensitivity to performance differences. Small MSE variations become difficult to distinguish when all errors fall within narrow ranges (e.g., 0.15-0.25).

- **ILP (Exact Solver)**: As data becomes noisier or more perturbed, the true optimum shifts upward, and ILP accurately tracks this movement due to its exact solving capability.
- **GA (Genetic Algorithm)**: Being a heuristic method, GA does not guarantee global optimality but explores adaptively. As the complexity of the problem increases, it occasionally finds slightly suboptimal solutions, resulting in the observed upward MSE trend.
- **VNS (Variable Neighborhood Search)**: This heuristic explores neighborhoods locally and typically finds robust but approximate solutions. When data is perturbed, VNS tends to converge to similar solution types, showing less steep increases in MSE.
- **CASPO**: Solutions appear to be more constrained and do not adapt as tightly to random perturbations, resulting in increasing but less steep error trends.

Mathematically, MSE can be derived as the sum of variance and bias. Perturbing the PKN (removing/adding edges, flipping logic) changes the model class over which the learning procedure searches. Our perturbation introduced both variance and bias. Removing true edges or forcing wrong edges means that the model class no longer contains the true data-generating mechanism. Therefore, bias increases (systematic error). Adding many spurious possibilities (splitting connection) can increase variance (model can overfit), and may or may not reduce bias.

Minimizing an empirical loss or variance-like term during fitting often reduces variance but does not remove structural bias introduced by a wrong PKN. In such a case, treating PKN as soft prior, regularizing the search or ensemble may mitigate the problem.

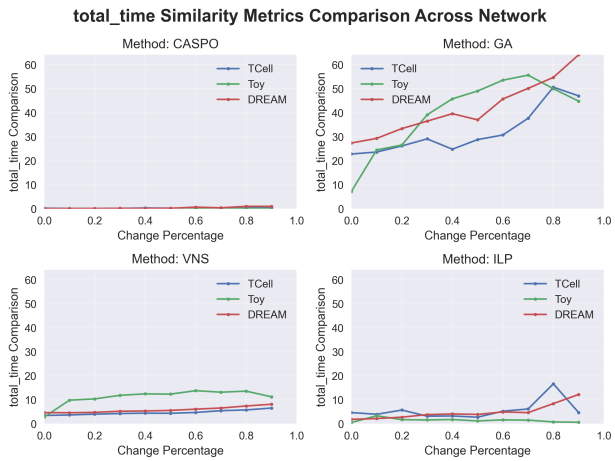

Figure 6: Total runtime comparison between Toy, DREAM and T-Cell datasets.

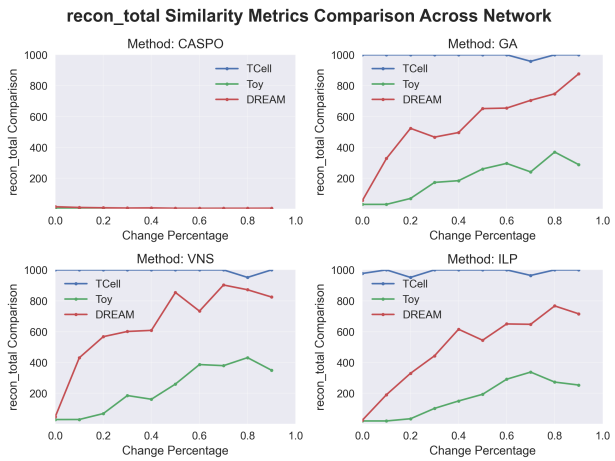

Figure 7: Number of reconstructed attractors: Toy vs DREAM.

Runtime analysis demonstrates significant differences in computational efficiency:

- **CASPO**: Maintains the fastest execution time across all datasets
- **GA**: Requires substantially more computation time on the DREAM dataset compared to the Toy dataset, while surprisingly requiring less time for the T-Cell dataset than DREAMpotentially due to insufficient synthetic data complexity
- **General Trend**: Increased perturbation levels generally correlate with longer optimization times.

The analysis of reconstructed attractor counts reveals that larger networks (DREAM and T-Cell) exhibit more rapid increases in attractor numbers with perturbation compared to the Toy dataset. This indicates an increase in dynamical fragmentation in larger networks, where structural perturbations more readily generate complex attractor landscapes.

Alternative Reconstruction Pipeline (Exploratory) We investigated the feasibility of reconstructing Boolean models directly from experimental data without prior knowledge networks. Our exploration included:

- **Path-centric reconstruction**: Identifying regulatory pathways from intervention data
- **Intervention-aware causal inference**: Leveraging experimental perturbations for causal discovery
- **Bayesian refinement methods**: Incorporating biological priors with data-driven approaches

Key Finding: Traditional causal inference methods designed for continuous systems fail in the Boolean context due to fundamental differences in discrete dynamics and experimental data requirements. Standard approaches like average treatment effect estimation are not directly applicable to binary state systems, leading to the failure of conventional Bayesian and causal discovery pipelines.

2.4.1 Summary of Findings

Our comprehensive benchmark analysis provides several key insights into the performance of the Boolean network optimization method.

Objective function: Using MSE solely as an objective function has some problems. First, minimizing MSE does not eliminate system bias. Second, MSE differences on small scales are difficult to distinguish, limiting comparative assessment.

Scalability Characteristics: Larger and more complex networks (DREAM and T-Cell) present significantly greater reconstruction challenges both structurally and functionally. All methods exhibit performance degradation with increased network size.

Method-Specific Performance Profiles: CASPO achieves optimal performance in topological similarity metrics while maintaining computational efficiency. ILP produces reasonable results in terms of MSE minimization, but shows a high variance. GA exhibits the longest computational requirements, but often achieves superior functional reconstruction stability. VNS shows tolerance to data perturbations.

Perturbation Sensitivity: All methods demonstrate critical performance thresholds around 10-30% perturbation levels, where both structural and functional reconstruction quality degrade substantially. This suggests fundamental limits to robustness in Boolean network optimization under structural uncertainty.

Evaluation Metric Limitations: Our analysis reveals that certain evaluation metrics (particularly the F1 score and Hamming distance) provide insufficient discriminatory power to distinguish between optimization methods, highlighting the need for more sensitive performance assessment criteria in this domain.

Parameter Optimization Trade-offs: The simulation step analysis reveals complex trade-offs between different optimization objectives. Extended simulations (240 steps) achieve lower MSE values, indicating better data fitting, but simultaneously reduce F1 performance and topological similarity metrics.

The experimental results collectively demonstrate that no single optimization method dominates in all evaluation criteria, suggesting that method selection should be guided by specific application requirements with respect to computational efficiency, reconstruction accuracy, and robustness to structural uncertainty.

3 Future Direction

Boolean networks demonstrate strong connections to several related mathematical frameworks, particularly probabilistic graphs and neural networks. Although Boolean networks differ from causal graphs, introducing probabilistic elements (beyond stochastic processes) would enable leveraging extensive probabilistic theoretical results.

Boolean functions can be conceptualized as specialized activation functions analogous to ReLU variants:

$$\text{ReLU-like function } : f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } 0 \leq x < 1 \\ 1, & \text{if } x \geq 1 \end{cases} \qquad \text{Step function } : f(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases}$$

This perspective suggests that simple continuous functions could approximate complex Boolean network behaviors, potentially enabling hybrid modeling approaches.

Boolean networks may benefit from Fourier transform analysis techniques. In graph signal processing, graph signals represent mappings over network structures. The application of Fourier transforms would enable the frequency domain analysis of the dynamics of the Boolean

network, similar to traditional signal processing methods. This approach could reveal periodic patterns, stability characteristics, and dynamic signatures in Boolean network behavior that are not apparent in time-domain analysis. Research Opportunities

Future research directions include

- Developing probabilistic Boolean network frameworks
- Exploring neural network approximations of Boolean functions
- Implementing graph signal processing techniques for Boolean network analysis
- Creating hybrid continuous-discrete modeling approaches

4 Acknowledge

I would like to express my sincere gratitude to Dr. Marek OSTASZEWSKI, who provided me with this invaluable research opportunity and offered exceptional guidance throughout the entire process. I am equally grateful to Dr. Pierre KLEMMER for his unwavering support, insightful feedback, and continuous encouragement. Furthermore, I extend my appreciation to Dr. Gleb SVININ for his valuable insights and engaging discussions that enriched my understanding of the subject matter.

References

[1] Sara Sadat Aghamiri, Vidisha Singh, Aurélien Naldi, Tomáš Helikar, Sylvain Soliman, and Anna Niarakis. Automated inference of Boolean models from molecular interaction maps using CaSQ. *Bioinformatics*, 36(16):4473–4482, May 2020.

[2] Jose A. Egea, David Henriques, Thomas Cokelaer, Alejandro Fernández Villaverde, Julio R. Banga, and Julio Sáez-Rodríguez. Meigo: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinformatics*, 15:136 – 136, 2013.

[3] Piotr Gawron, Marek Ostaszewski, Venkata Satagopam, Stephan Gebel, Alexander Mazein, Michal Kuzma, Simone Zorzan, Fiona McGee, Bruno Otjacques, Rudi Balling, and Reinhard Schneider. Minervaa platform for visualization and curation of molecular interaction networks. *npj Systems Biology and Applications*, 2:16020, 2016.

[4] Enio Gjerga, Panuwat Trairatphisan, Attila Gabor, Hermann Koch, Celine Chevalier, Franceco Ceccarelli, Aurelien Dugourd, Alexander Mitsos, and Julio Saez-Rodriguez. Converting networks to predictive logic models from perturbation signalling data with cellnopt. *Bioinformatics*, 36(16):4523–4524, 2020.

[5] Carito Guziolowski, Santiago Videla, Federica Eduati, Sven Thiele, Thomas Cokelaer, Anne Siegel, and Julio Saez-Rodriguez. Exhaustively characterizing feasible logic models of a signaling network using answer set programming. *Bioinformatics*, 29(18):2320–2326, 2013.

[6] Ahmed Abdelmonem Hemedan, Reinhard Schneider, and Marek Ostaszewski. Applications of boolean modeling to study the dynamics of a complex disease and therapeutics responses. *Frontiers in Bioinformatics*, Volume 3 - 2023, 2023.

[7] Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[8] Christian Müssel. BoolNet–an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10):1378–1380, 2010.

A Summary of Optimization algorithm

Table 2: Comparative Analysis of Boolean Model Optimization Methods

Challenge	**CellNOpt**	**optPBN**	**MEIGO**	**Caspo**
Parameter Estimation and Network Inference	—	—	BayesFit (MCMC Bayesian Inference)	—
Global Optimization Challenges	Integer Linear Programming (ILP)	particle swarm optimisation (PSO), evolutionary algorithm (EA)	Enhanced Scatter Search (eSS), Variable Neighbourhood Search (VNS)	Answer Set Programming (ASP), ASP Solver (Clingo)
Stochastic Dynamics and Control	—	PBN	—	—
Multi-objective Optimization Requirements	Weighted sum	—	arbitrary objective functions	lexicographic optimization
Model Selection and Validation	Cross-validation, Prior Knowledge Integration, Network Compression	Cross-validation on Control Policies, MoBoSS	—	Global Truth Table

B Result Analysis

B.1 Individual results

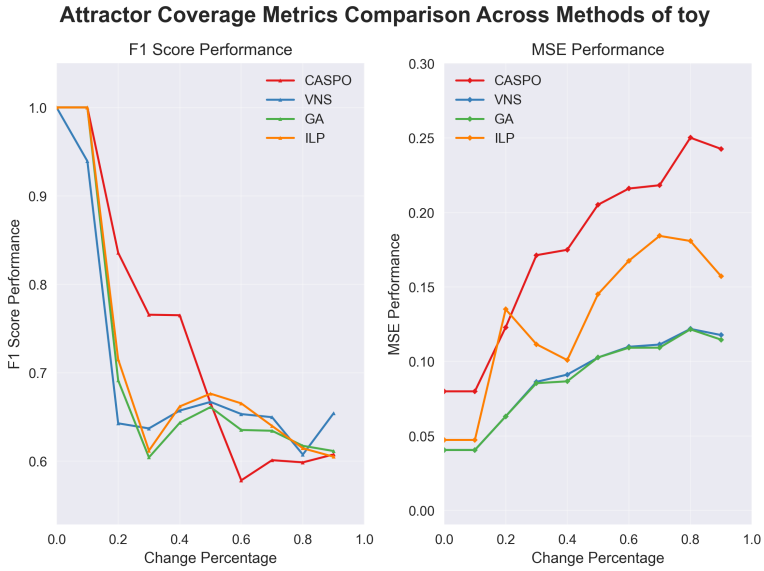

Figure 8: Attractor coverage metrics on the Toy dataset.

The attractor coverage analysis evaluates how frequently each optimization method (CASPO, ILP, VNS, GA) successfully recovers the reference model's attractor behavior as the prior knowledge network undergoes progressive structural perturbations. These curves serve as robustness indicators: flatter profiles indicate better preservation of dynamical outcomes under structural modifications. Our results demonstrate that CASPO maintains relatively high attractor coverage at low perturbation levels but exhibits steady degradation as the perturbation intensity increases. ILP shows considerable variance across experimental replicates, while VNS and GA demonstrate comparatively stable performance in the intermediate perturbation range.

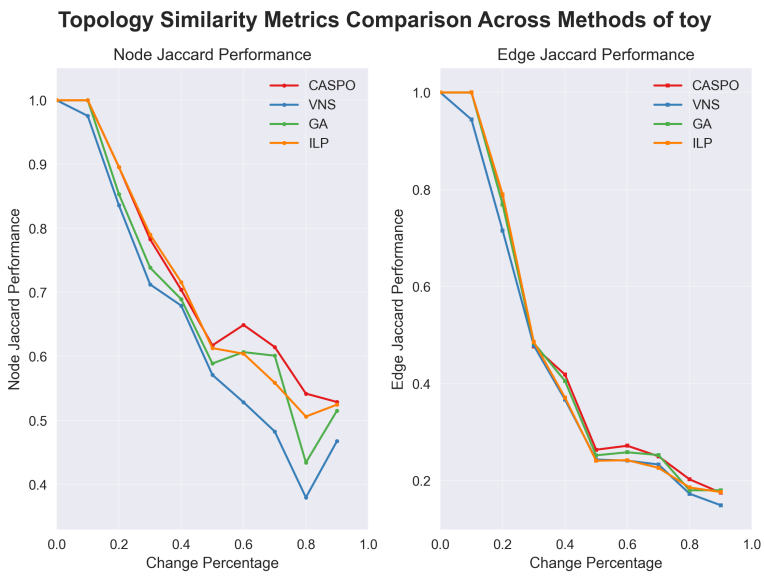

Figure 9: Topology-based similarity metrics on the Toy dataset.

Topological similarity metrics quantify the structural correspondence between reconstructed networks and ground truth through edge-level Jaccard indices and related graph distance mea-

sures. All methods exhibit performance degradation with increasing perturbation levels, with edge-level similarity typically declining more rapidly than coarse-grained node-level measures.

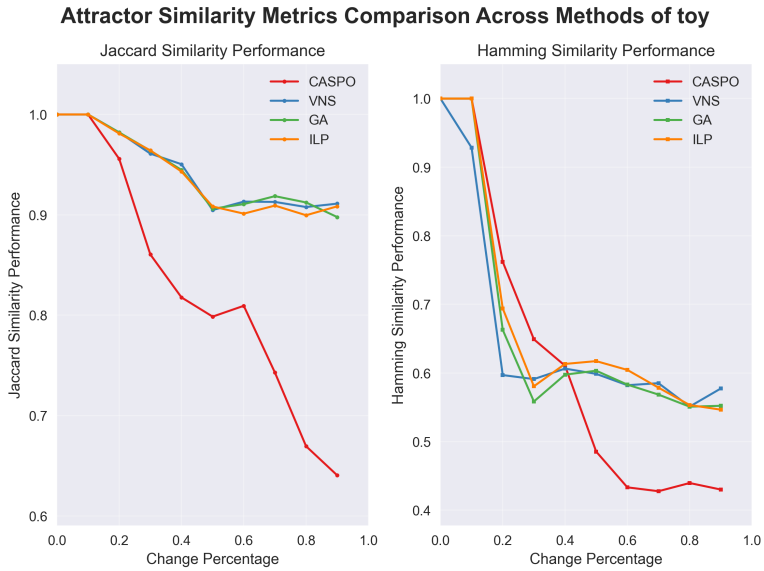

Figure 10: Attractor similarity metrics on the Toy dataset.

The assessment of similarity of the attractor employs multiple distance metrics (Hamming, Jaccard) to evaluate the functional correspondence of dynamical results rather than structural matching. All methods exhibit a notable performance threshold between the perturbation levels of 10 20%, indicating that moderate structural changes can substantially alter the dynamics of the system. CASPO demonstrates a continuous decline with perturbation intensity, whereas heuristic methods (VNS, GA) exhibit superior functional similarity preservation at intermediate perturbation levels.

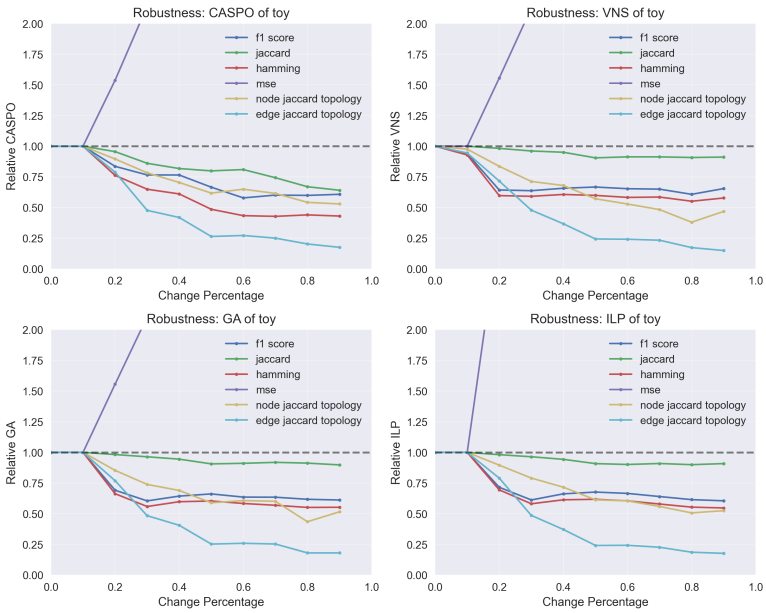

Figure 11: Robustness analysis on the Toy dataset.

Robustness analysis provides an integrated assessment by aggregating multiple evaluation metrics (structural and dynamical) to offer a comprehensive view of method resilience. For perturbation levels below approximately 10%, CASPO, GA, and ILP maintain comparable

performance characteristics. The VNS exhibits a modest performance decline in earlier perturbation stages. In particular, the edge-level Jaccard similarity decays more rapidly than others.

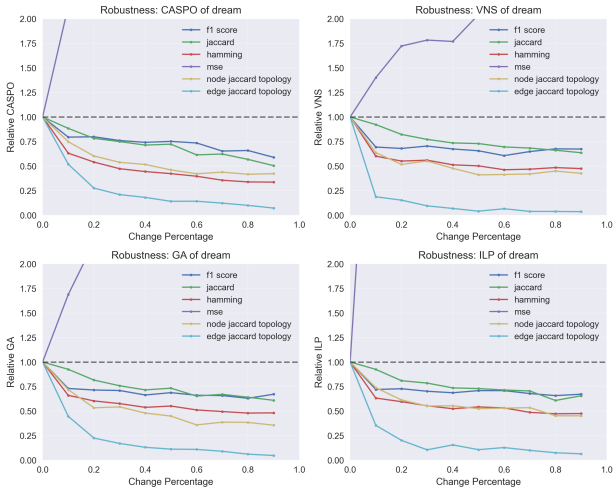

Figure 12: Robustness analysis on the DREAM dataset.

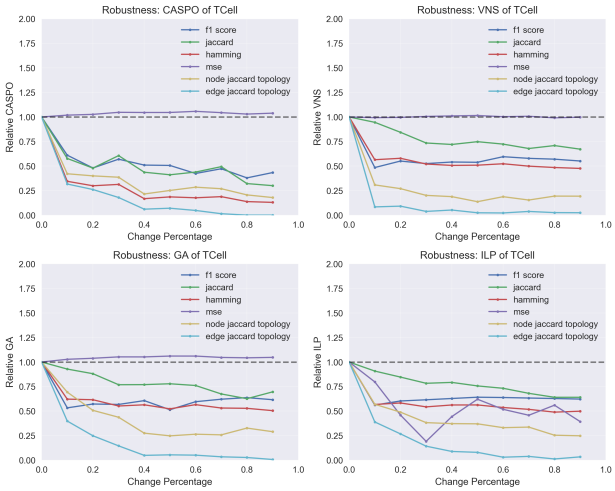

Figure 13: Jaccard similarity comparison between Toy, DREAM and T-Cell datasets.

The Mean Squared Error (MSE) profiles reveal distinct behavior patterns across datasets. Both the Toy and DREAM models exhibit rapid increases in MSE with perturbation level, indicating sensitivity to structural modifications. In contrast, the T-cell model displays relatively stable MSE fluctuations, which could be attributable to the synthetic nature of the experimental data used in this analysis.

Figure 14: Runtime and reconstructed attractor-size comparison on the Toy dataset.

Computational efficiency and reconstruction quality analysis reveal practical trade-offs inherent in each optimization approach. CASPO demonstrates the fastest execution times in these experiments while producing attractor-size distributions that closely match the reference model. GA requires the longest computational time, but often achieves more stable functional reconstructions. Other methods exhibit intermediate run-time characteristics with greater variability in reconstructed attractor counts. These results provide guidance in balancing computational cost with reconstruction quality requirements in practical applications.

B.2 Comparison between datasets

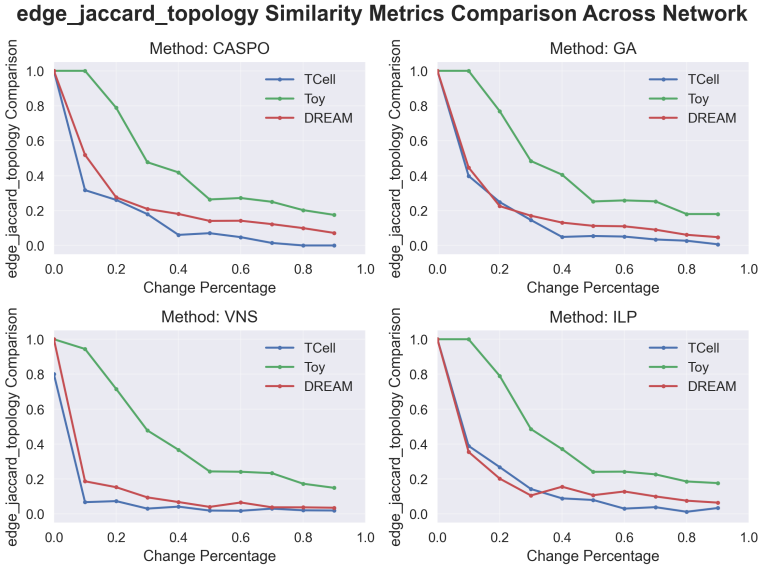

Figure 15: Edge Jaccard and topology similarity: Toy vs DREAM.

The relationship between network size and optimization performance is clearly demonstrated in this comparative analysis. Larger network models consistently exhibit degraded performance in all evaluation metrics. And we can say that CASPO performs best in the topology sense.

Figure 16: F1 score comparison between Toy, DREAM and T-Cell datasets.

The F1 score analysis fails to reveal clear relationships between network size, perturbation level, and method performance. This suggests that the F1 score may not provide sufficient discriminatory power for evaluating Boolean network optimization methods in this experimental context, as it does not effectively capture the relationship between performance characteristics and network scale or perturbation intensity.

B.3 Parameter Analysis

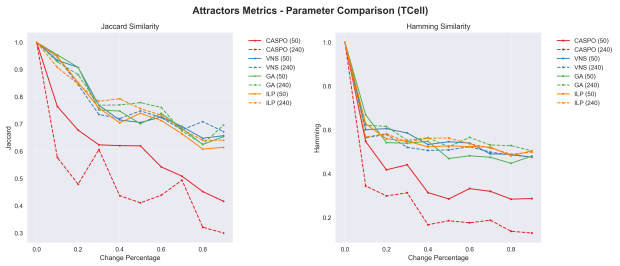

Figure 17: Effect of simulation steps on attractor reconstruction across optimization methods.

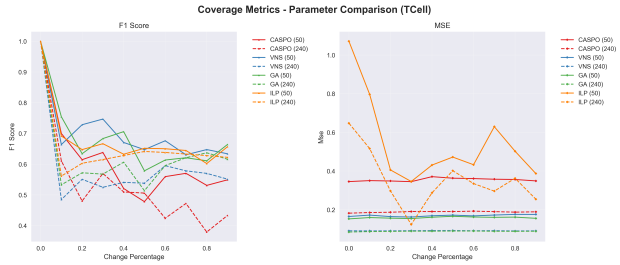

Figure 18: Simulation step effects on MSE and F1 performance metrics.

The analysis of simulation step parameters reveals distinct patterns across optimization methods. CASPO, ILP, and GA demonstrate relatively consistent attractor reconstruction regardless of simulation length. However, CASPO exhibits a counterintuitive trend in which increased simulation steps correlate with diminished performance, suggesting potential sensitivity to extended dynamics or computational overhead effects.

The right plot reveals a clear trade-off between different performance metrics. Extended 240-step simulations consistently produce lower MSE values in all methods, indicating better data fitting. In contrast, F1 performance scores decrease with longer simulations, suggesting that extended dynamics may compromise functional similarity measures. This divergence highlights the inherent tension between different optimization objectives. Moreover, 19 shows that the length of more simulations generates a more complex dynamical system.

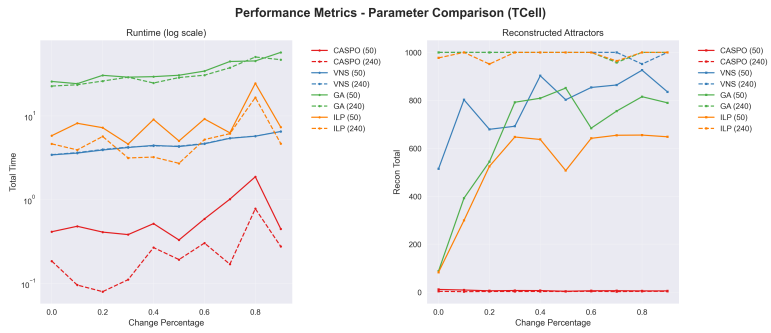

Figure 19: Computational complexity and dynamic system characteristics across simulation lengths.

Computational analysis reveals that 50-step simulations require more intensive optimization efforts and generate more complex dynamical systems across most methods. The sole exception is CASPO, which maintains consistent computational behavior regardless of the length of the simulation. This pattern suggests that shorter simulations may capture more diverse or challenging dynamical behaviors that require greater algorithmic effort to optimize effectively.

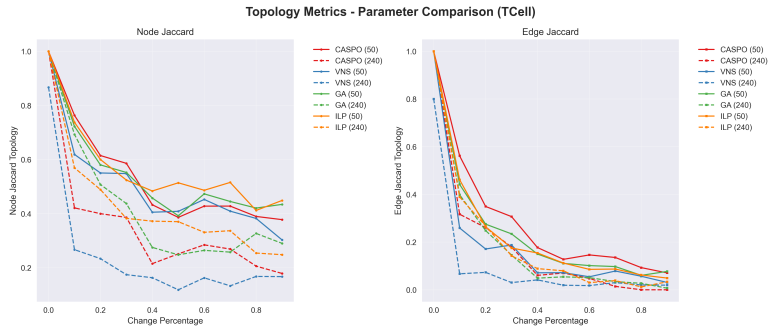

Figure 20: Topological similarity metrics: impact of simulation step length.

The quality of the topological reconstruction, measured through Jaccard indices at the node level and the edge level, consistently favors shorter 50-step simulations over extended 240-step protocols. This finding indicates that longer simulations may introduce dynamical complexity that obscures the underlying network structure, making topological reconstruction more challenging for optimization algorithms.