

Argumentation Framework Subsampling Generator

Lars Malmqvist
The Tech Collective
Denmark
lama@thetechcollective.eu

Abstract—This paper introduces benchmark abstract argumentation frameworks generated via systematic subsampling of existing instances. Four distinct methods are employed: random, degree-based, Breadth-First Search (BFS), and community-based subsampling. These techniques yield derived frameworks that preserve specific structural properties of the source frameworks while reducing their scale, thereby generating a spectrum of computational challenges for evaluating argumentation solvers.

I. INTRODUCTION

Reasoning tasks over Abstract Argumentation Frameworks (AFs) [1], such as determining extensions under various semantics, are computationally demanding. Evaluating and advancing solver performance necessitates diverse benchmark suites. This work contributes such a suite by generating derived AFs through systematic subsampling of source frameworks. This approach yields instances with controlled structural properties, reflecting characteristics of real-world argumentation structures while varying in scale and complexity. This is particularly useful in machine learning applications, where large amounts of diverse but related samples can help approximation accuracy [2].

II. SUBSAMPLING METHODOLOGY

We implemented four distinct subsampling methods. Given a source AF = (A, R) and a proportion parameter $p \in (0, 1]$, a subsampled AF' = (A', R') is generated where $|A'| \approx k = \lceil p \cdot |A| \rceil$ and $R' = \{(a, b) \in R \mid a, b \in A'\}$. The methods are:

A. Method 1: Random Subsampling

Random subsampling selects a subset $A' \subseteq A$ of size k uniformly at random. This baseline method is computationally efficient but may disrupt significant structural features of the original AF, such as attack chains or central conflicts. Formally:

$$A' = \text{Sample}(A, k) \quad (1)$$

where $\text{Sample}(S, n)$ returns n elements chosen uniformly at random from set S . The attack relation R' is induced by A' .

B. Method 2: Degree-based Subsampling

This method selects arguments based on their connectivity, hypothesizing that high-degree arguments are structurally significant. Arguments with high in-degree or out-degree often represent critical points within the argumentative structure. Optionally, extension membership frequency can be incorporated. A score is assigned:

$$\text{score}(a) = \text{in-degree}(a) + \text{out-degree}(a) + \alpha \cdot \text{extension-count}(a) \quad (2)$$

where $\text{extension-count}(a)$ is the frequency of a in preferred extensions (if available, weighted by α). Arguments are selected greedily based on score:

$$A' = \{a_1, \dots, a_k\} \text{ s.t. } \text{score}(a_i) \geq \text{score}(a_{i+1}) \forall i \quad (3)$$

This method tends to preserve dense regions, potentially increasing instance difficulty.

C. Method 3: BFS (Breadth-First Search) Subsampling

BFS-based subsampling, akin to snowball sampling, preserves local topological structure by exploring the neighborhood around a seed argument. It aims to maintain connected components and attack/defense chains, which are crucial in argumentation dynamics. Unlike random or degree-based sampling, BFS ensures the preservation of local connectivity. The process starts from a randomly selected seed $a_0 \in A$.

Algorithm 1 BFS Subsampling

```

1: Select random  $a_0 \in A$ 
2: Initialize  $A' = \{a_0\}$ ,  $Q = [a_0]$  (Queue)
3: while  $|A'| < k$  and  $Q \neq \emptyset$  do
4:   Dequeue  $a$  from  $Q$ 
5:   for each  $b$  s.t.  $(a, b) \in R$  or  $(b, a) \in R$  do
6:     if  $b \notin A'$  then
7:        $A' \leftarrow A' \cup \{b\}$ ; Enqueue  $b$  in  $Q$ 
8:   if  $|A'| = k$  then break
9: return  $A'$ 

```

The algorithm explores both incoming and outgoing attacks to capture the argument's local context.

D. Method 4: Community-based Subsampling

Real-world AFs often exhibit community structure (dense intra-group connections, sparse inter-group connections), representing distinct sub-topics or perspectives. This method preserves this modular organization. Community detection (e.g., using the Louvain method on the underlying undirected graph derived from R) identifies communities C_1, \dots, C_m . Arguments are then sampled proportionally from each community.

This method contrasts with random (structure-agnostic) and BFS (local focus) sampling by maintaining the global organization of the AF.

Algorithm 2 Community-based Subsampling

```
1: Detect communities  $C_1, \dots, C_m$  in AF (e.g., via modularity
   optimization)
2: Initialize  $A' = \emptyset$ 
3: for each community  $C_i$  do
4:   Compute quota  $q_i = \lceil \frac{|C_i|}{|A|} \cdot k \rceil$ 
5:    $S_i = \text{Sample}(C_i, \min(q_i, |C_i|))$ 
6:    $A' \leftarrow A' \cup S_i$ 
7: return  $A'$ 
```

III. IMPLEMENTATION

A Python implementation is provided, processing argumentation frameworks in the standard .af format. It comprises a library (`subsampling_lib.py`) implementing the core methods using NetworkX for graph operations, and a command-line tool (`af_subsample.py`) for batch generation. The tool processes source directories, applies selected subsampling methods with configurable parameters (e.g., proportion p , number of samples per method), and produces structured output directories containing the derived AFs in .af format. This modular design supports both specific generation tasks and large-scale benchmark creation, with reusable library components.

IV. BENCHMARK CHARACTERISTICS

The generated benchmarks offer controlled variation:

A. Size and Structural Properties

- 1) **Scalability:** Variable sizes ($|A'| \approx p \cdot |A|$) allow testing solver scalability.
- 2) **Structural Diversity:** Each method preserves distinct structural aspects: random (baseline), degree-based (hubs/centrality), BFS (local topology/chains), community-based (global modularity).
- 3) **Difficulty Gradient:** The combination of methods and proportions p yields instances spanning a range of computational complexities.

B. Empirical Observations

Preliminary empirical evaluation using standard argumentation solvers indicates relative difficulty patterns:

- Degree-based sampling (especially for $p > 0.7$) tends to produce the most computationally challenging instances, likely due to the preservation of dense, interconnected subgraphs.
- Random sampling generally results in easier instances, attributed to the disruption of complex attack structures.
- BFS sampling yields instances of intermediate difficulty, potentially varying with the seed choice.
- Community-based sampling tends to scale the difficulty relative to the original framework's structure and the sampling proportion p .

These observations align with the theoretical understanding that structural properties, particularly cycles and dense connectivity preserved by methods like degree-based sampling,

significantly impact the complexity of argumentation reasoning.

REFERENCES

- [1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence*, vol. 77, no. 2, pp. 321–358, 1995.
- [2] Malmqvist, L., Yuan, T., and Nightingale, P. (2024). Approximating Problems in Abstract Argumentation with Graph Convolutional Networks. *Artificial Intelligence*, 336, 104209.