Abstract Zobrist Hashing: An Efficient Work Distribution Method for Parallel Best-First Search

Yuu Jinnai and Alex Fukunaga (The University of Tokyo)

1. Hash Distributed A* (HDA*)

Hash Distributed A* (HDA*) is a parallel best-first graph search (A*) which distributes nodes according to a hash function which assigns each state to a unique process. As HDA* relies on the hash function for load balancing, the choice of hash function is crucial to its performance!

2. Previous Work Distribution Methods

2.1. Zobrist hashing (ZHDA*)

(Zobrist 1970; Kishimoto et al. 2013)

**Goal:** Distribute nodes uniformly among process

**Method:** xor the hash value of each feature

ZH(s) = R[t1] xor R[t2] xor ... xor R[tn]

**Strength:** good load balance

**Limitation:** high communication overhead

2.2. Abstraction (AHDA*)

(Zhou and Hansen 2007; Burns et al. 2010)

**Goal:** Assign neighbor nodes to the same process

**Method:** Project nodes in the state space into abstract states, and abstract states are assigned to processors using a modulus operator.

AH(s) = R[A(s)]

**Strength:** low communication overhead

**Limitation:** worse load balance

3. Abstract Zobrist hashing (AZHDA*)

**Goal:** Distributes nodes uniformly while assigning neighbor nodes to the same process

**Method:** Project features into abstract features and xor the hash value of each abstract feature

AZH(s) = R[A(t1)] xor R[A(t2)] xor ... xor R[A(tn)]

**Strength:** simultaneously address good load balance and communication overhead!

4. Results on Sliding-tile and Multiple Sequence Alignment

AZHDA* outperformed previous methods on 15-puzzle, 24-puzzle, and MSA, successfully mitigating communication overhead and achieving good load balance, while ZHDA* and AHDA* only succeeded in achieving either low CO or good load balancing.

**Communication overhead (CO)** := (#nodes sent to other processes) / (#nodes generated)

**Search overhead (SO)** := (#nodes expanded in parallel) / (#nodes expanded in sequential A*)

Search overhead is caused by poor load balance.

5. Results on Domain-Independent Planning

AZHDA* achieved modest improvements over previous methods with automatically generated abstract features.