A Cache-based Approach to Dynamic Switching between Different Dataflows in Crowdsourcing

Yusuke Suzuki†, Masaki Matsubara†, Keishi Tajima‡, Toshiyuki Amagasa†, Atsuyuki Morishima†

Background

We may want to change dataflows halfway.

Dataflow Example: Identifying the missing and survivors when a disaster occurs

- Task 1: Please enter the characteristics of the person in the photo.
  - Results: 50s, Woman

- Task 2: Are the people in the two photos the same person?
  - Results: Not same

Example: Change to an efficient method because the number of survivors was less than expected.

Change

- Task 3: Is there someone in the photo on the right that shows the person on the left?
  - Results: One missing person
  - Multiple answers can be obtained with one task

Problem & Purpose

Existing method cannot cope with the changing dataflows[1].

To reduce the total monetary cost in the dataflow change process

Proposed Method

Propose a rerunning method in changing dataflows.

Combining Caches

- Process by cache

Additional Task

- Issue additional tasks to use dangling results. However, there is a tradeoff between cost reduction by cache and the cost increase by additional tasks.

Cost Estimation

- Compare the costs of the two plans
  - Monetary cost for the old dataflow
  - Monetary cost for the new dataflow

Simulation

- It is possible to identify the best point to minimize the total cost and there is no obvious solution.

Simulation settings

- d: Data nodes, c: Cost
  - Task cost distribution
    - Total cost of the task in the dataflow before change: 0.7
    - The task cost in the dataflow after change: 0.7

Simulation Results

- The minimum cost of each distribution appears in red.