**What is DART?**

Efficient engine for discovery of opportunities to aggregate multiple similar purchases by independent buyers fractured across organization.

DART is based on a probabilistic bi-clique clustering algorithm that runs in polynomial time.

**Benefits of Demand Aggregation?**

- Lower prices in bulk purchasing contracts
- Larger vendor tendering
- Lower shipping and handling fees
- Reduced legal and admin overhead

**How does Demand Aggregation work in an organization?**

- **Understand the process**
  - Collate and combine requirements of multiple buys
  - Contract suppliers based on combined demand
  - Standardise and establish best-buy strategy

- **Know the problem**
  - Fractured purchasing process done individually by departments and units results in similar buys for different prices from multiple suppliers
  - Big amount of data makes it difficult to understand best options for cost savings
  - Complex network of suppliers, items they provide, prices and demand over time makes it difficult to solve this multi-variate problem with standard approaches

**SOLUTION DESCRIPTION**

- **Demand Aggregation to Bi-clique clustering**
  - Construct vendor - item bipartite graph

- **Problem re-definition**
  - Find max bi-cliques is equivalent to biggest potential Demand Aggregation (DA) patterns

- **Procurement specific customisations**
  - Requirements for parameters such as min/max purchasing value, volume, trends over time etc.

- **Work in realistic production scenario**
  - Bi-clique clustering is NP-complete but DART is polynomial complexity!

**EVALUATION AND DEPLOYMENT**

- **Evaluated on 3 year procurement database**
  - 271,219 items x 7,319 vendors [1,032,275 POs]

- **3 rounds of evaluation**
  - Modifications to bi-clique clustering algorithm and adding procurement post-processing filters

- **Deployed in A*STAR Procurement Office**
  - Decision support system for annual reports

**TECHNOLOGY USAGE**

- **Language** | JAV A8
- **3rd Party Libraries** | Apache 2.0 + LGPL
- **Commercialisation and Use** | Semantist startup