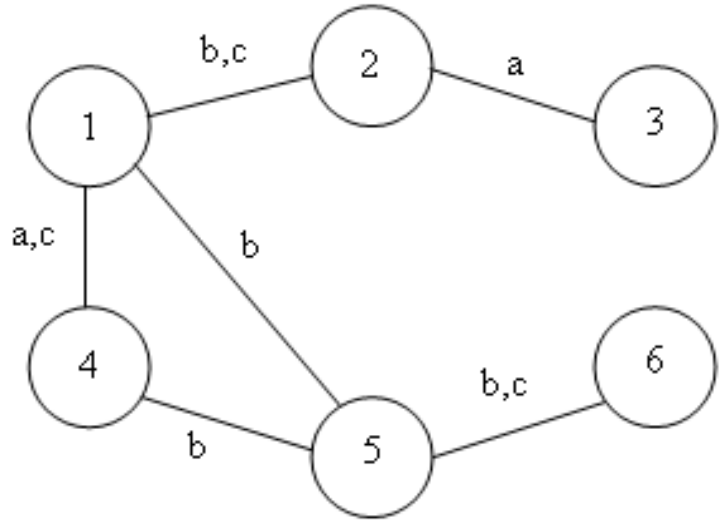
Comparison of Heuristics for Solving the GMLST Problem

Yiwei Chen, Namrata Cornick, Andrew O. Hall, Ritvik Sahajpal, John Silberholz, Inbal Yahav, Bruce Golden

GMLST Problem Definition

- Each edge labeled with one or more label
- To select edge, use exactly one label
- Goal: construct a spanning tree using the smallest number of distinct labels

GMLST Problem Example



- 2 to 3 with *a*, 1 to 2 with *c*, 1 to 4 with *c*, 4 to 5 with *b*, 5 to 6 with *c*
  - *a, b,* and *c* used, so total cost of **3**
- 2 to 3 with *a*, 1 to 2 with *b*, 1 to 5 with *b*, 4 to 5 with *b*, 5 to 6 with *b*
  - *a* and *b* used, so total cost of **2**

MIP Formulation

- Created MIP formulation for GMLST problem
- Could only solve problems up to size 50 nodes with MIP model using CPLEX
- Even with faster computer or better MIP solver or use of cuts to speed solution of MIP, would not run in reasonable speeds for large problems

Need for Heuristics

- Demonstrates needs for heuristics
  - Modified MVCA
  - Rarest Insertion
  - Iterative Perturbation and Correction
  - Modified Genetic Algorithm
  - Increasing Diverse Population Genetic Algorithm

Modified MVCA

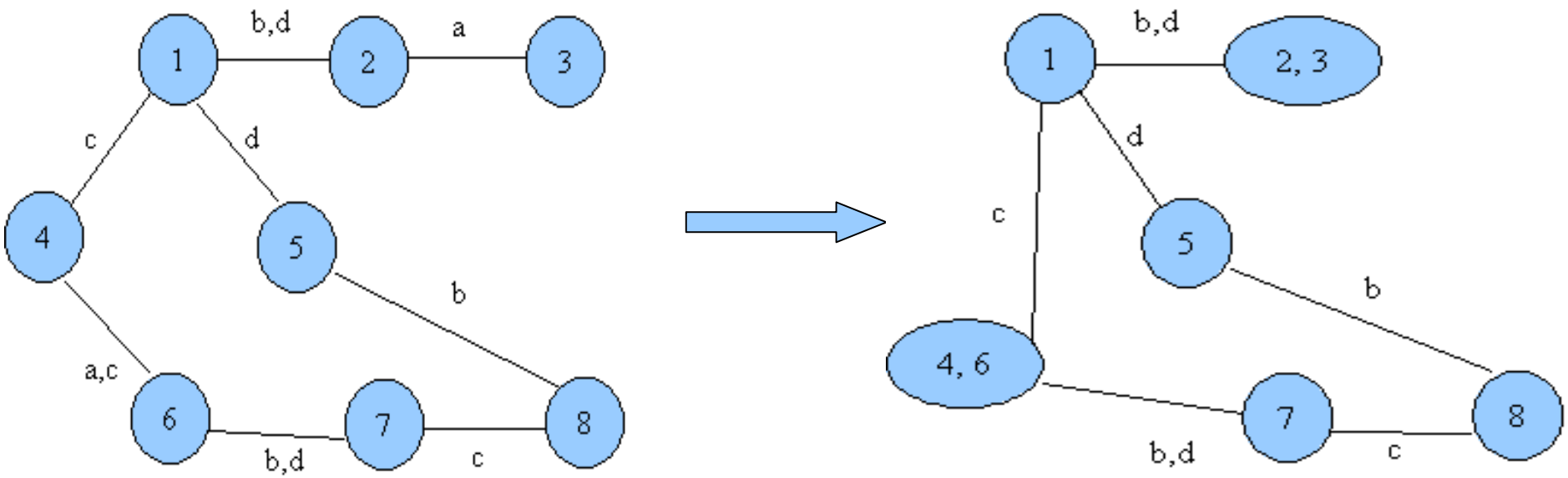
- Proposed in Chang and Leu, 1997
- Store: *C*, a list of labels added
- Iteratively:
  - Calculate the number of components in the subgraph induced by *C* on graph *G*
  - Calculate list *P* of labels p_i that minimize the number of components in the subgraph induced by *C* U $\{p_i\}$ on graph *G*
  - Randomly select label *p* from *P* and add it to *C*
  - Continue until the subgraph induced by *C* on graph *G* is connected

Rarest Insertion

- Similar to MVCA: store list of labels added *C*
- Each iteration, select component of subgraph induced by *C* on *G* that is connected to other components by the fewest labels (call the list of these labels *L*)
- Perform MVCA considering only labels in L
- As in MVCA, continue until spanning tree has been constructed

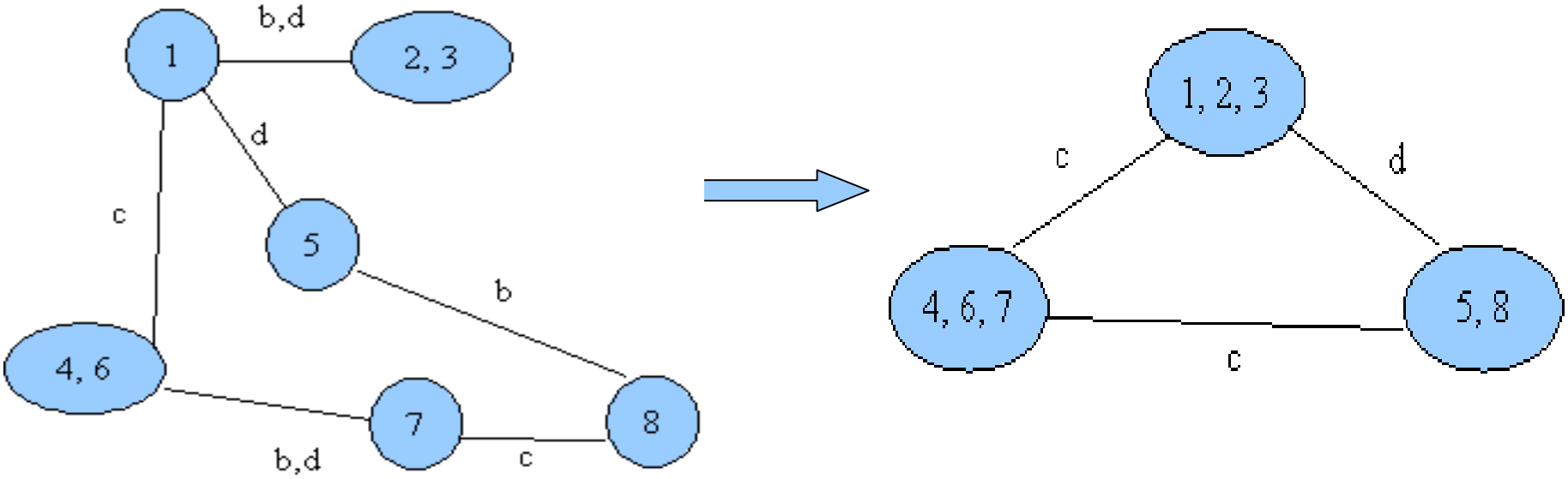
Rarest Insertion Example

- Component {3} is rarest component
- Label *a* is added



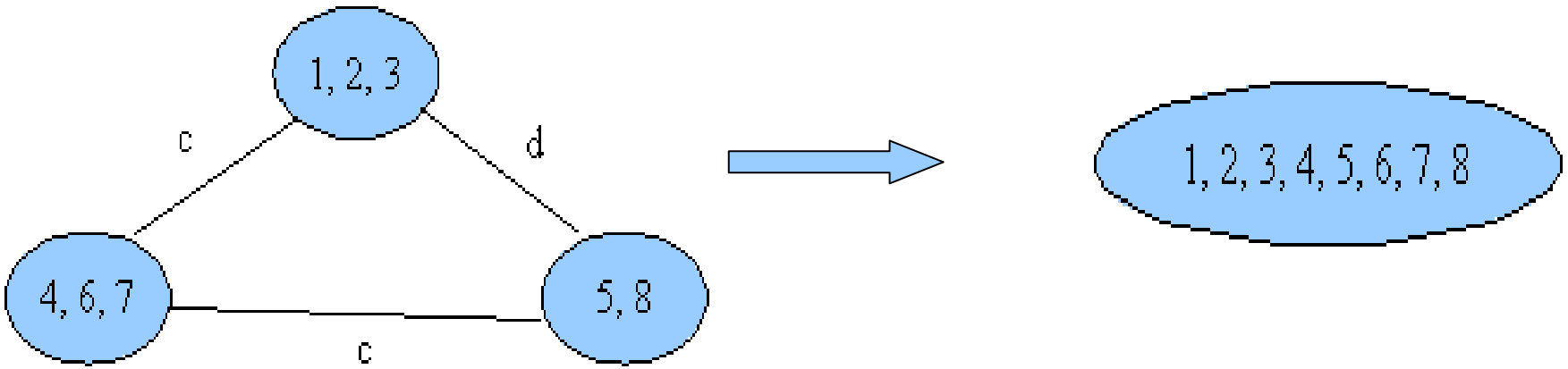
Rarest Insertion Example

- Components {2,3}, {5}, and {8} tie for rarest
- Select {8} randomly
- Labels *b* or *c* can be added
- By MVCA rule, *b* is added



Rarest Insertion Example

- Component {4, 6, 7} is rarest
- Label *c* is added
- We are done, with final label set {a, b, c}



Iterative Perturbation and Correction

- Uses weighted selection with exponents
- Weight for a label is $e^{-\lambda(\frac{F-f_l}{F})}$
- F is maximum frequency of any label in graph, f_l is frequency of label l
- Parameter λ guides how greedy selection is

Iterative Perturbation and Correction

- To create initial solution, use exponential selection repeatedly
- Iteratively:
  - Remove labels until none more can be removed
  - Add a label selected with exponential selection
- Repeat until no improvement within a set number of iterations

Modified Genetic Algorithm

- Proposed in Xiong et al., 2005
- Genetic algorithm solution for the MLST problem
- No modification needed to use this approach for the GMLST problem

Increasing Diverse Population GA

- Initial population generation
  - Randomly select 10 labels and add the best one
  - Repeat until chromosome is valid
- Crossover
  - Maintain all labels in both parents
  - Randomly select unused labels from both parents until offspring is valid
- Mutation
  - Add random label

Increasing Diverse Population GA

- Local Search
  - Non-intensive – Remove one label, if possible
  - Intensive – Remove as many labels as possible
- Generation Structure
  - Select chromosomes for replacement using roulette wheel selection
  - Replace with offspring of crossover, where parents are selected with roulette wheel selection
  - Perform mutation and local search

Increasing Diverse Population GA

- Population Structure
  - Disallow chromosomes with concurrent label sets
  - Maintain 6 isolated populations
    - Non-intensive local search
  - Combine best solutions after a set number of generations
  - Maintain final population for a set number of generations
    - Intensive local search
    - Add 1 MVCA solution each generation

Computational Testing

- Small-world datasets
  - Based on technique from Watts and Strogatz, 1998
  - Smaller number of labels per instance
- TSPLib-based datasets
  - Label datasets deterministically
  - Can be reproduced for comparison of results
  - Larger number of labels per instance

Computational Testing

- Small Datasets

Heuristic	IDP	RMVCA	RRI	MGA	IPC	Sum
IDP	0	23	18	16	4	61
RMVCA	0	0	5	1	0	6
RRI	0	14	0	2	0	16
MGA	1	22	16	0	1	40
IPC	5	23	17	15	0	60
Sum	6	82	56	34	5	183

- Medium Datasets

Heuristic	IDP	RMVCA	RRI	MGA	IPC	Sum
IDP	0	28	29	13	14	84
RMVCA	3	0	10	3	8	24
RRI	4	21	0	1	9	35
MGA	18	30	27	0	20	95
IPC	15	28	24	11	0	78
Sum	40	107	90	28	51	316

Computational Testing

- Large Datasets

Heuristic	IDP	RMVCA	RRI	MGA	IPC	Sum
IDP	0	17	19	13	17	66
RMVCA	3	0	7	7	12	29
RRI	2	11	0	8	10	31
MGA	10	17	17	0	20	64
IPC	8	14	15	5	0	42
Sum	23	59	58	33	59	232

- Modified GA and Increasing Diverse Population GA most effective
- Repeated Rarest Insertion improvement over Repeated MVCA
- Iterative Perturbation and Correction does well on small datasets and those with low densities

Conclusions

- Heuristics performed well
- Several heuristics presented were competitive with the best published to date
- Future Directions
  - Real-world datasets
  - Mathematical analysis of rarest insertion algorithm runtime
  - Test GMLST problem solutions on MLST problem