Comparison of Heuristics for the Colorful Traveling Salesman Problem

John Silberholz

R.H. Smith School of Business

University of Maryland

Joint Work With:

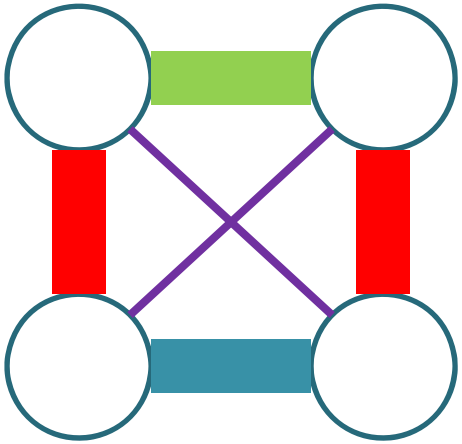
Andrea Raiconi, Raffaele Cerulli, Monica Gentili, Bruce Golden, Si Chen

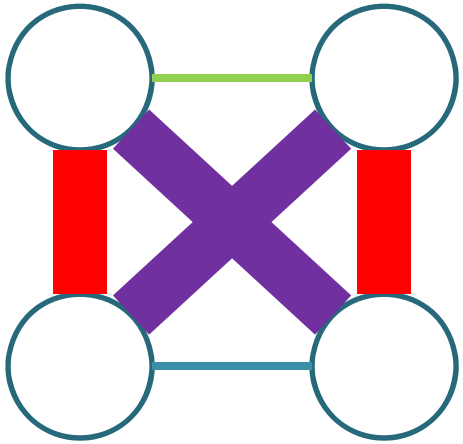
Presentation Outline

- The CTSP and its applications
- Existing work
- Two new approaches
  - Iterated local search heuristic
  - Genetic algorithm
- Computational results
- Conclusions

The Colorful Traveling Salesman Problem (CTSP)

- Label (color) associated with each edge
- Goal: Minimize number of labels used in a Hamiltonian tour.





- Cost: 3

Cost: 2

Applications

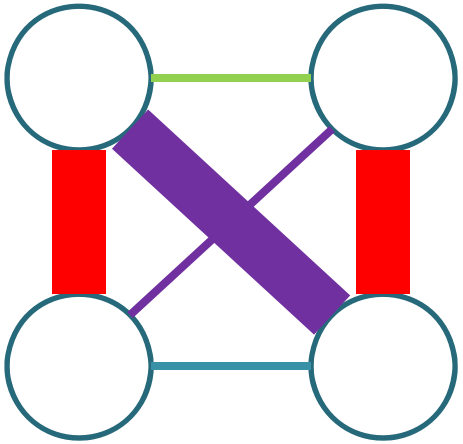
- TSP with fixed-rate transportation providers
- Minimizing types of threats to a convoy as it visits locations
- Minimizing specializations needed in a machine scheduling problem

Existing work

- Cerulli et al. (2006)
  - Greedy algorithm and tabu search
  - Proof that CTSP is NP-complete
- Xiong et al. (2007)
  - Greedy algorithm: MPEA
- Couëtoux et al. (2008)
  - Greedy algorithm
- Jozefowiez et al. (2011)
  - Branch-and-cut algorithm

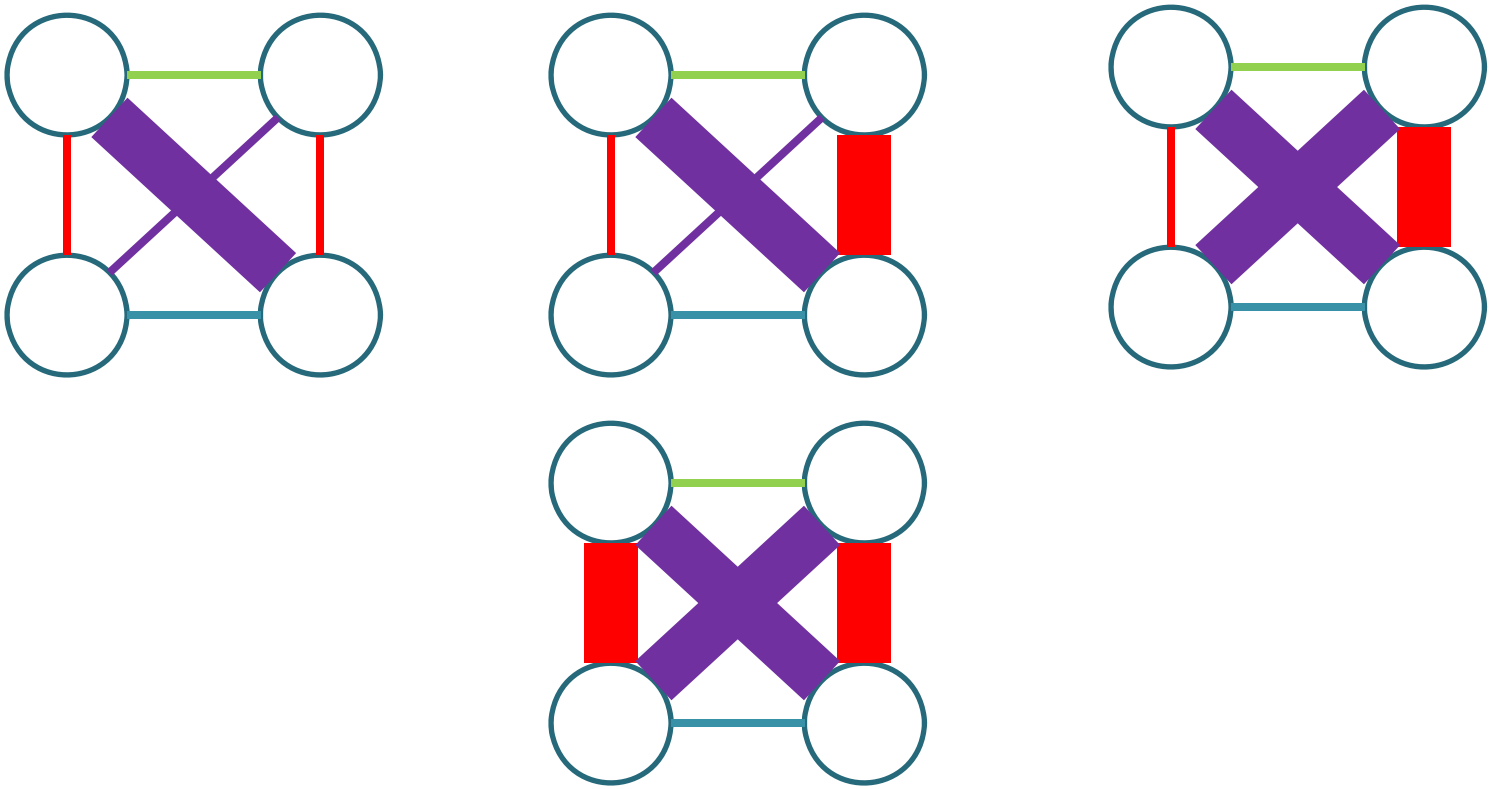
Maximum Path Extension Algorithm (MPEA)

- Proposed in Xiong et al. (2007)
- First, record the labels from a heuristic solution to the minimum label spanning tree problem



MPEA

- Extend from endpoints with same labels
  - If impossible, add with highest frequency label
- Finally, connect the endpoints

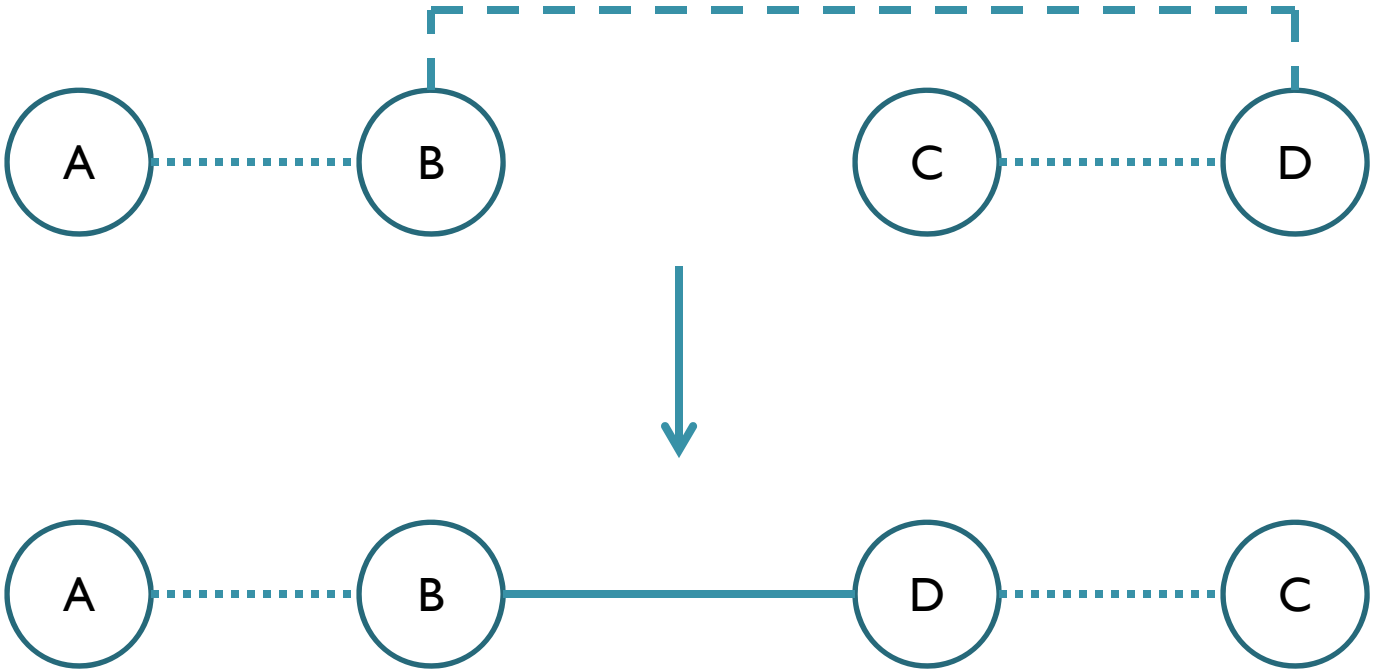


Iterative Deconstruction and Repair heuristic (IDR)

- Start with an initial Hamiltonian tour
  - We'll see later how to get this
- Randomly select a label in the tour, proportional to its inverse frequency
  - Remove every edge with this label from the tour
  - Repair to create a new Hamiltonian tour
    - Connect disjoint paths
    - Don't add new labels when repairing
- Repeat until no labels can be removed

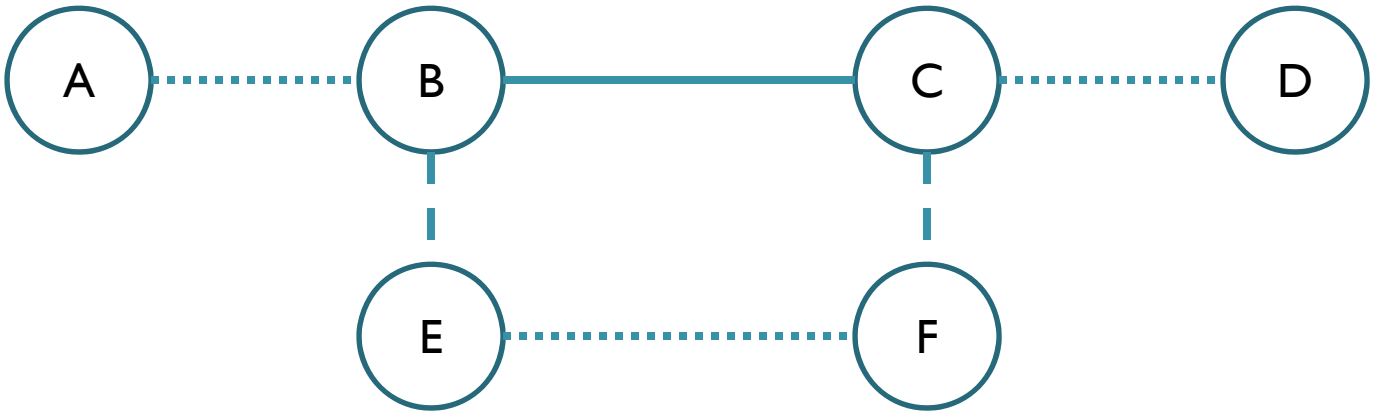
Direct path connection

- Connect paths if an edge with a label in the acceptable label set C exists between their endpoints



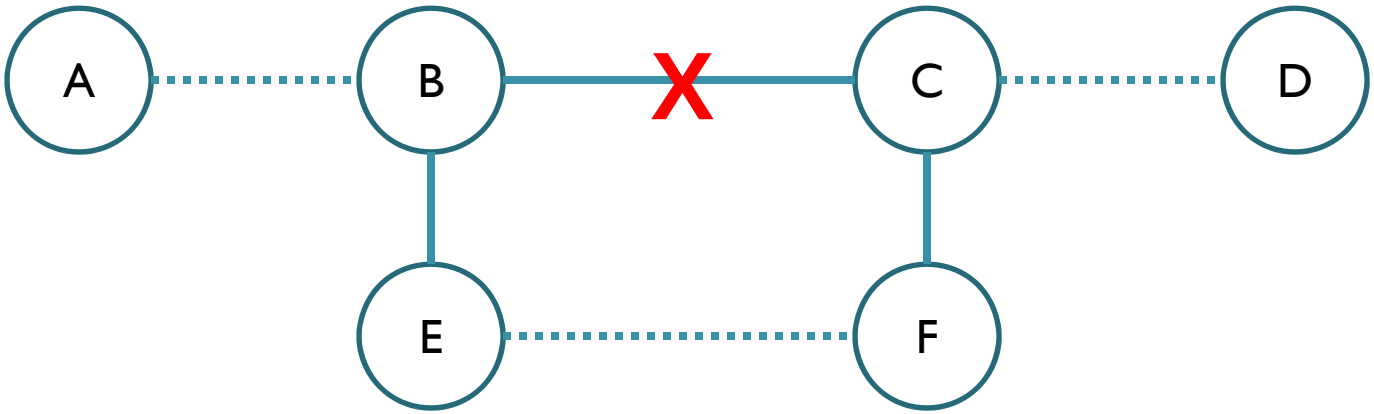
Path Insertion

- Insert one path inside another using only labels in the acceptable label set C



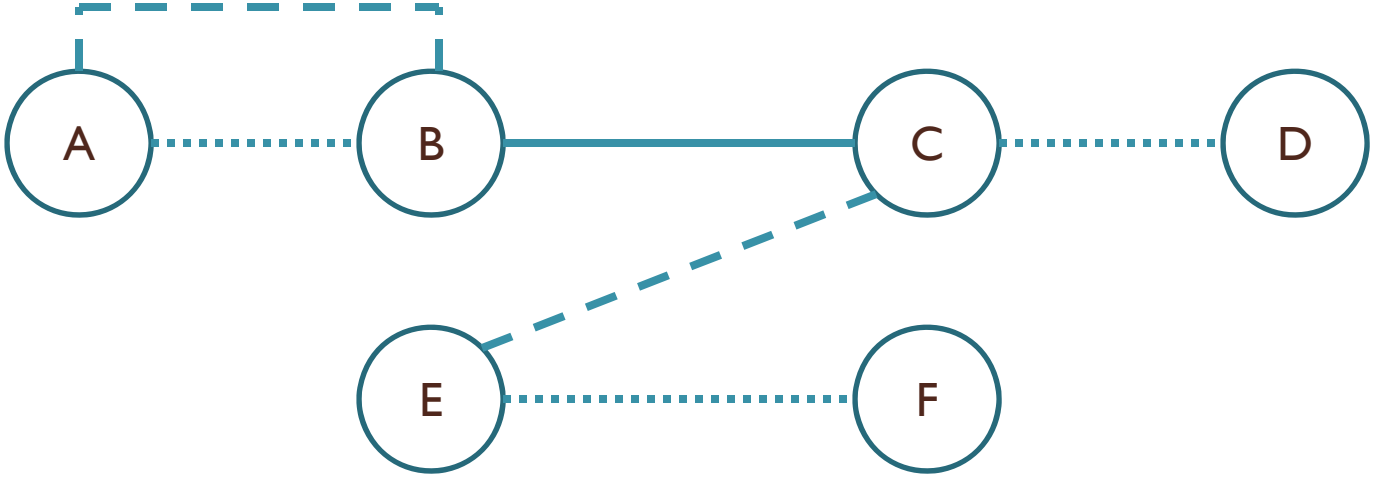
Path Insertion

- Insert one path inside another using only labels in the acceptable label set C



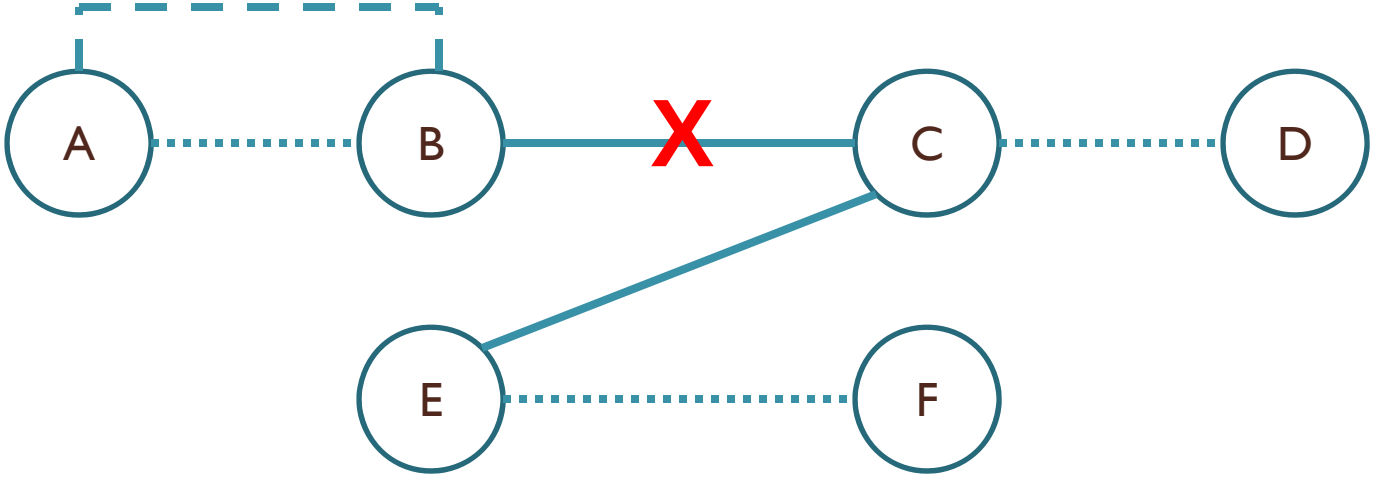
Circular splicing

- Path insertion is more powerful when the endpoints are linked with a valid edge



Circular splicing

- Path insertion is more powerful when the endpoints are linked with a valid edge

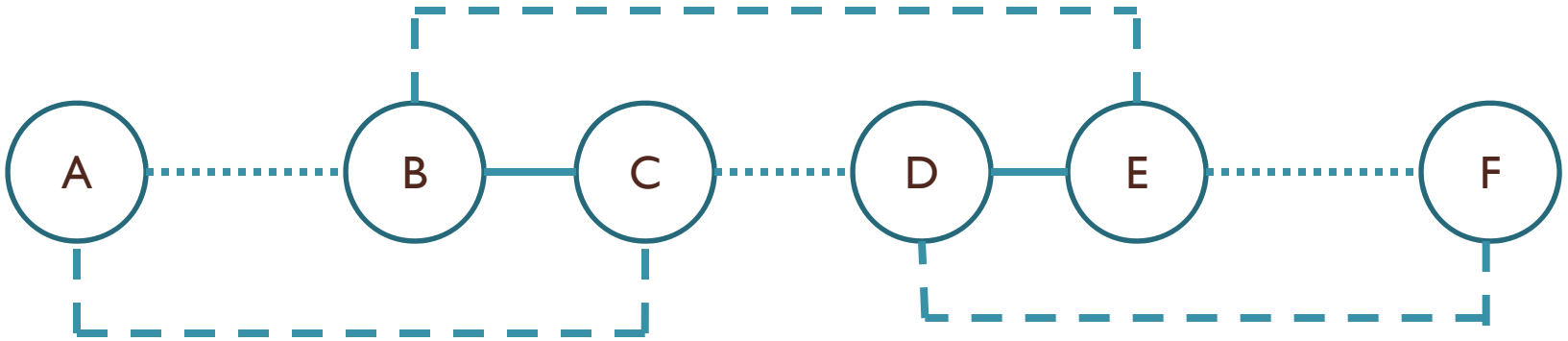


Combining into a Hamiltonian path

- While we have multiple paths:
  - Use direct path connection if possible
  - Use path insertion if possible
  - Use circular splicing if possible
    - Subject to a parameterized limit
  - Use non-circular splicing if possible
    - Subject to a parameterized limit
- Fail if we reached our limit

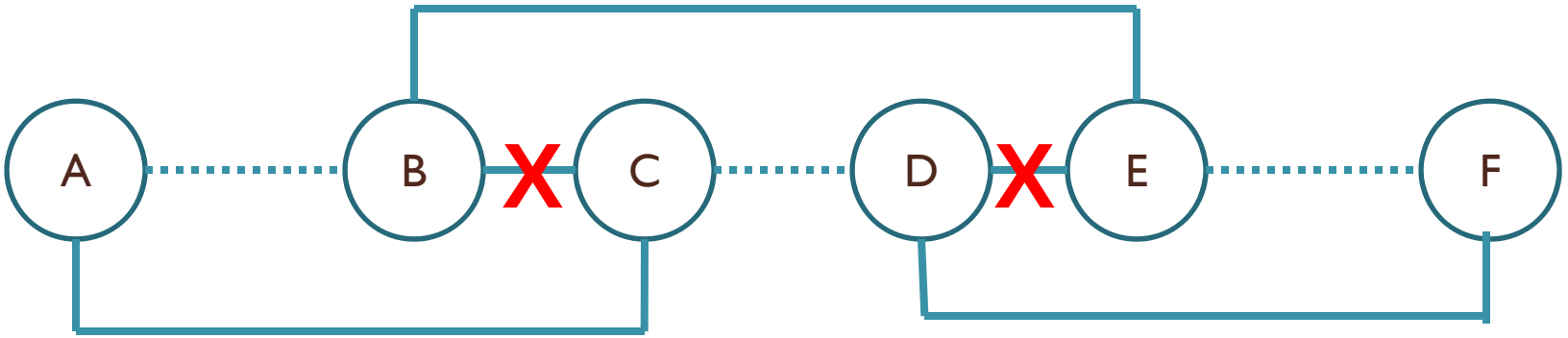
Converting to a Hamiltonian cycle

- Connect endpoints linked with valid edge
- Otherwise, try direct endpoint matching:



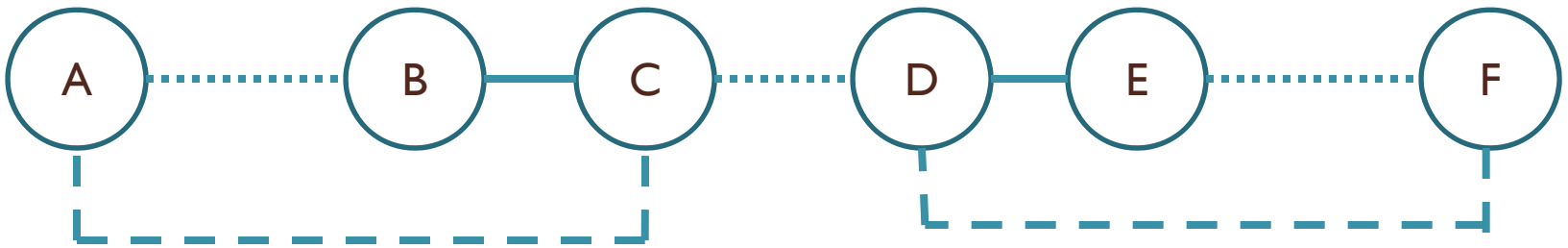
Converting to a Hamiltonian cycle

- Connect endpoints linked with valid edge
- Otherwise, try direct endpoint matching:



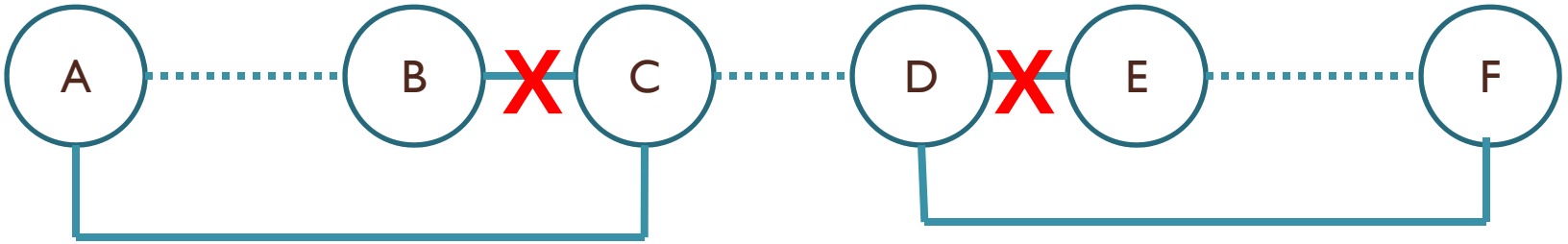
Changing the endpoints

- If we can't do direct endpoint matching, we need to change the endpoints and try again.



Changing the endpoints

- If we can't do direct endpoint matching, we need to change the endpoints and try again.



Final technique for converting paths to cycles

- Use direct endpoint matching if possible
- Change the endpoints a single time and then use direct endpoint matching if possible
- If unsuccessful, randomly change the endpoints and try again
  - Repetition subject to a parameterized limit

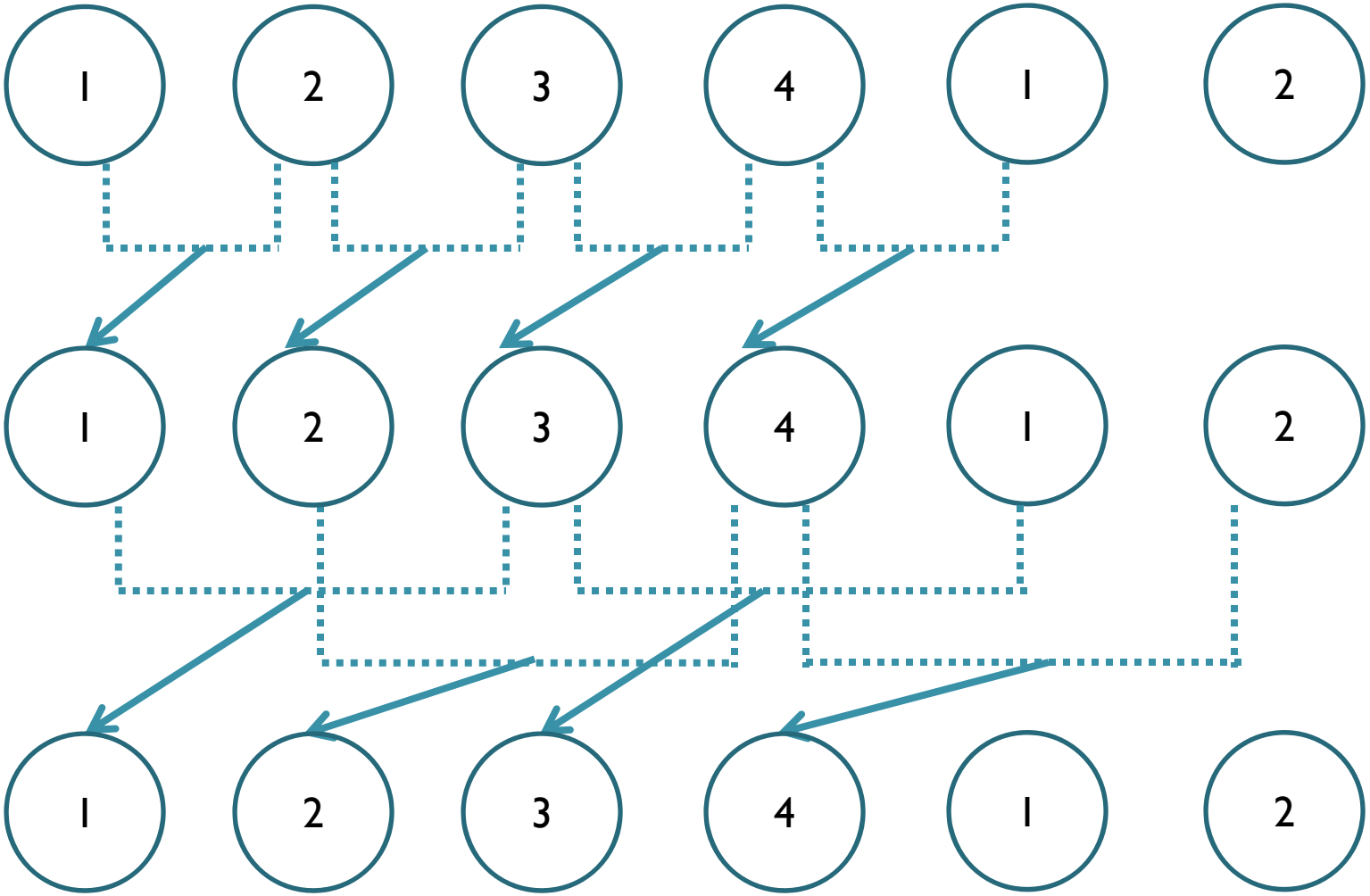
Generating an initial solution

- Allow all colors
- Start with each path as a single node
- Reconnect using the path repair approach just described

Deconstruction and Repair Genetic Algorithm (DRGA)

- Uses one-parameter genetic algorithm due to Xiong et al. (2005)
- p-1 generations, each with p chromosomes
- The chromosomes crossover and mutate in a structured manner

DRGA population (p=4)



DRGA Crossover

- Each label in the first parent as ½ chance of remaining
  - Deconstruction results in disjoint paths
- Randomly select labels from either parent to allow, and run the repair heuristic
- Repeat until a new Hamiltonian tour is generated

DRGA replacement policy

- After crossover, a chromosome is mutated:
  - Allow a random label not in the chromosome
  - Randomly remove labels until none can be removed
- If the child is better than the first parent, replace in the new generation
  - Break ties randomly

Performance on random instances

	50-100 Nodes		150-200 Nodes	
	Gap (%)	Sec.	Gap (%)	Sec.
iMPEA	45.6	0.2	71.3	0.9
IDR	18.7	0.1	19.7	0.2
DRGA	2.3	1.9	3.6	4.7

- Each category has 8 instances
- Each heuristic run 5 times per instance
- Optimal solutions from Jozefowiez et al. (2011)

Performance on TSPLib instances

	101-280 Nodes, 102-784 Labels		532-1,000 Nodes, 2,830-10,000 Labels	
	Gap (%)	Sec.	Gap (%)	Sec.
iMPEA	59.1	9	50.2	1714
IDR	23.6	2	29.6	185
DRGA	3.1	62	2.1	2765

- Labels from Chen et al. (2008)
- Each category has 32 instances
- Gaps are from best heuristic result
  - On nearly all instances, this is DRGA
  - DRGA gaps are variability to seed over 5 runs

Conclusions

- New "deconstruction and repair" approach to CTSP and related problems
- Best heuristic solution qualities to date
  - First to produce near-optimal solutions
- Reasonable runtime growth