A Fast Heuristic for Genetic Algorithms in Link Weight Optimization

Christoph Reichert

Thomas Magedanz

Fraunhofer FOKUS, Berlin, Germany

Outline

- Traffic Engineering in IP Networks
- Genetic Algorithms
- The Local Heuristics
- Results
- Conclusion

TE in IP Networks

- OSPF/IS-IS paths are determined by SPF and link weights
- Choose link weights so that
  - congestion is avoided
  - delay bounds are met
  - etc
- Fortz, Thorup, “Traffic Engineering by Optimizing OSPF Weights”

Objective Function

- Fixed:
  - network topology
  - link capacities
  - traffic matrix
- Given:
  - link weights
- Return:
  - maximum link load (MLL) in the network
- Goal:
  - minimize MLL

Genetic Algorithms

- Population of chromosomes
- Start with random population
  - evaluate each chromosome
  - select parents according to their fitness (fitness = 1/MLL)
  - crossover: generate children from parents
  - random mutation
- Until #generations reached
- return best chromosome

Local Heuristic as Inner Loop

- Increment weight of most loaded link
- Inner Loop:
  - increase MLL weight…
  - …as long as max. load does not increase
- Inner loop finds nearby local minimum.

Directed Mutation

- Increment MLL weight by one.
- Explicit operation within GA.

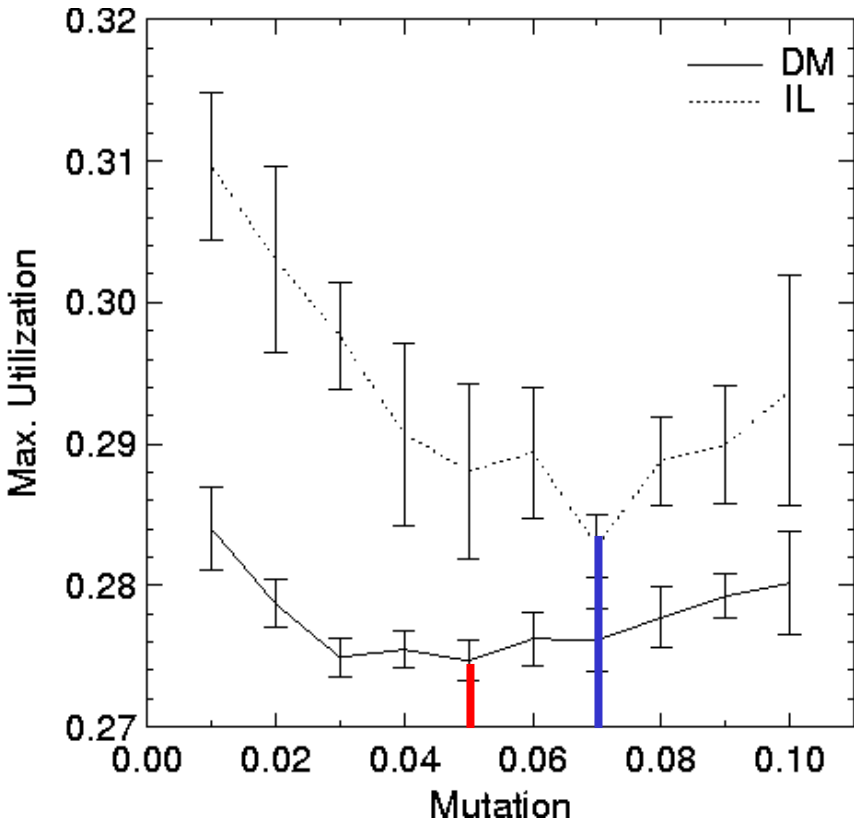
Caching

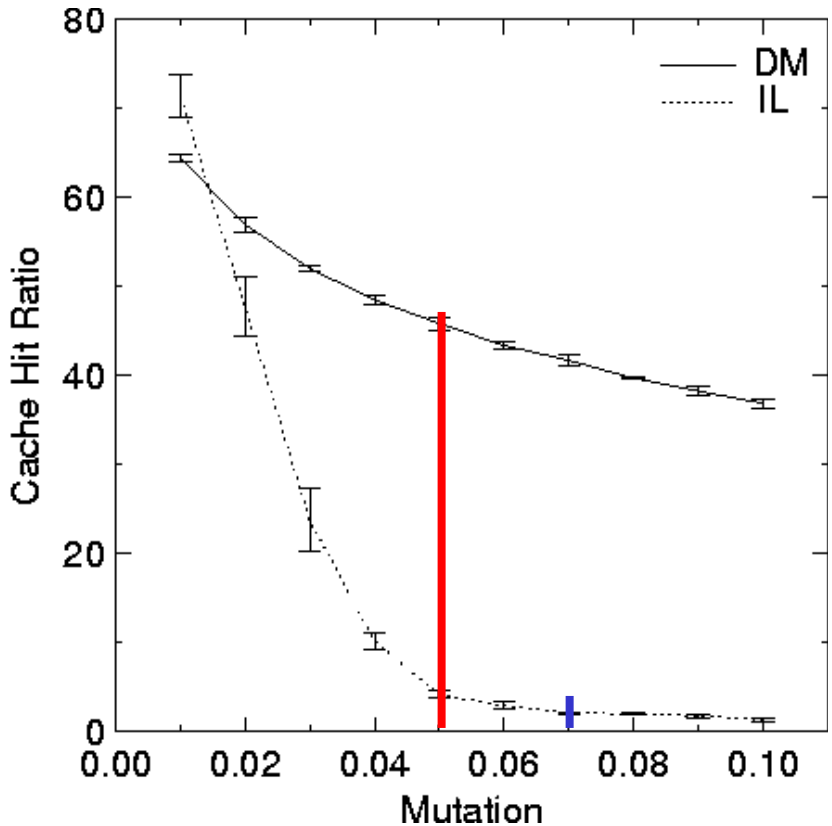
- Store chromosome with max. load in a cache.
- Cache lookup before calling the real objective function.
- Count cache hits => cache hit ratio

Evaluation Framework

- Compare IL and DM
- Each strategy gets the same amount of evaluations (including cache hits)
- Which one returns better results?
  - final MLL
  - cache hit ratio

Comparison: IL versus DM





Conclusion

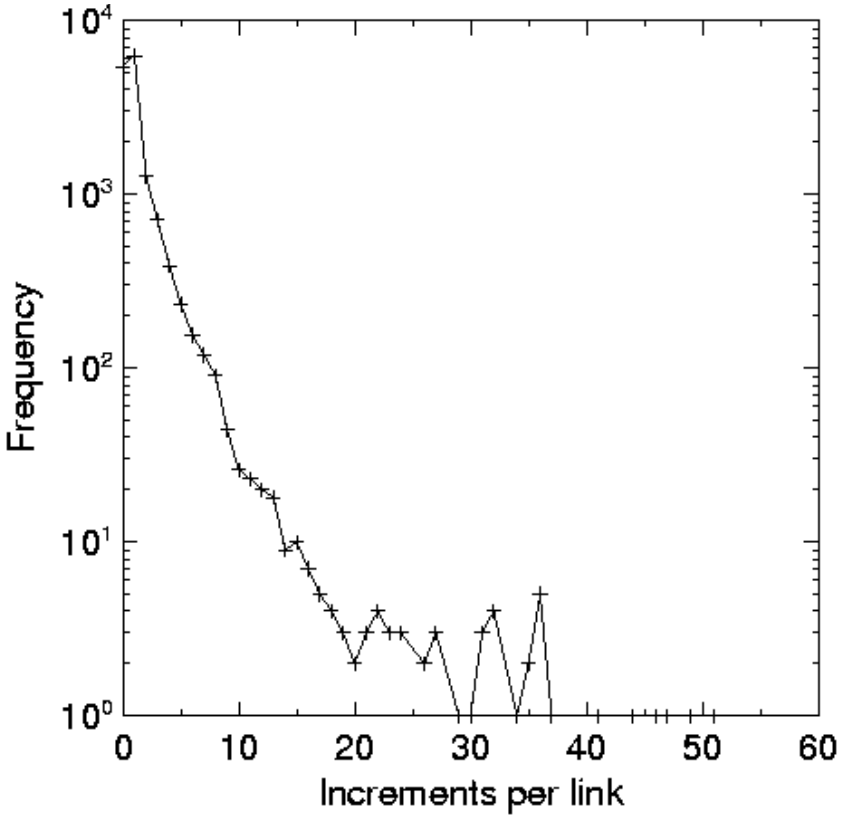
- Directed Mutation outperforms Inner Loop in terms of
  - quality (lower MLL)
  - reliability (lower variance)
  - CPU performance (much more cache hits)

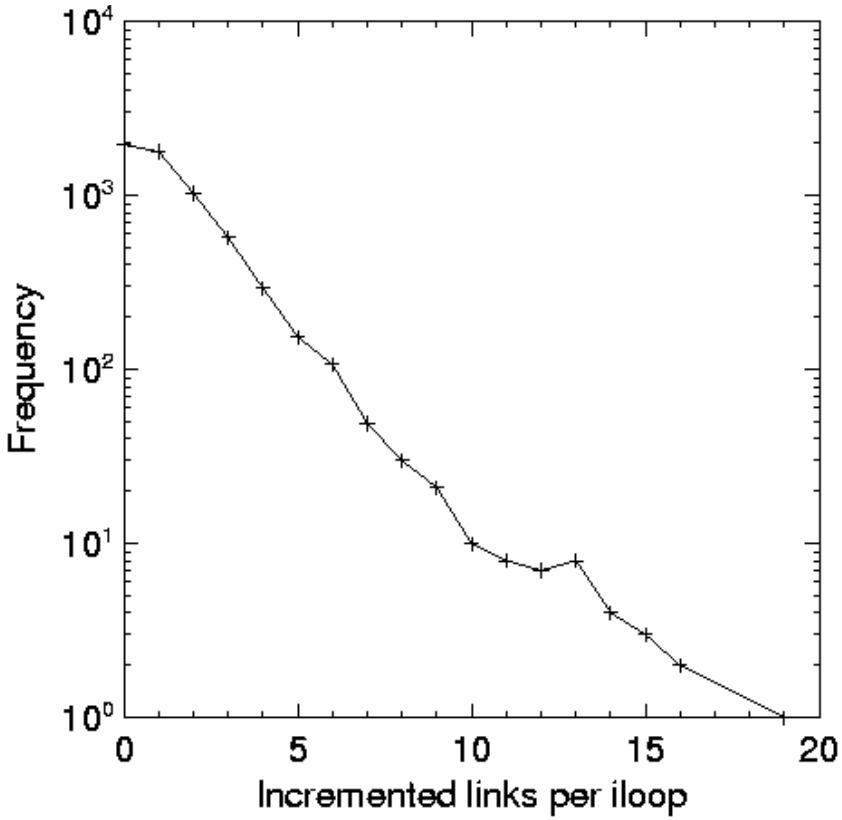
Conclusion (contd)

- Why should a strategy exploring each point in the search space twice (on avg.) be optimal?

THANK YOU!

Inner Loop Investigations





Genetic Algorithms (contd)

- Rank selection:
  - sort chromosomes according to fitness
  - selection probability depends on position in sorted sequence
- N-Point-Crossover:
  - randomly choose N splitpoints for a pair of parents
  - exchange N+1 subsequences alternately
- Random Mutation:
  - randomly select a link weight
  - assign new random value