Multihop Routing Optimization in Communication Networks Using Genetic Algorithms

Shilpa Sirikonda

Department of Electrical Engineering & Computer Science
KU

Master's Project Defense

December 4, 2007

Committee

Dr. James P. G. Sterbenz (Chair)

Dr. Alexander M. Wyglinski (Co-Chair)

Dr. Victor Frost

Outline

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Introduction

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Introduction

- Challenges faced by multihop networks
  - Finding the best path between end nodes
  - Achieving all the desired metrics simultaneously
- For example, it is difficult to find a path
  - Minimizing both the number of hops and BER
- Earlier GAs were used for single metric optimization
- Proposed approach
  - Multi-objective GA optimization is proposed
  - Simultaneously optimizes five conflicting metrics

Genetic Algorithms (GA) Overview

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Genetic Algorithms (GA) Overview

- GA is a random search technique
  - Searches for the best fit based on a 'fitness function'
- Search space
  - Population of binary coded configurations
  - Configurations are also called 'chromosomes' or 'strings'
- Fitness function
  - Evaluated at each individual point in the search space
  - Repeated over several generations
  - A configuration is found that meets the desired objective

Genetic Algorithms (GA) Overview

- Configurations of next generation
  - Selected through a genetic transformation process
  - Transformation done using genetic operators
- Genetic Operators
  - Reproduction
  - Crossover
  - Mutation

Genetic Operators

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Genetic Operators

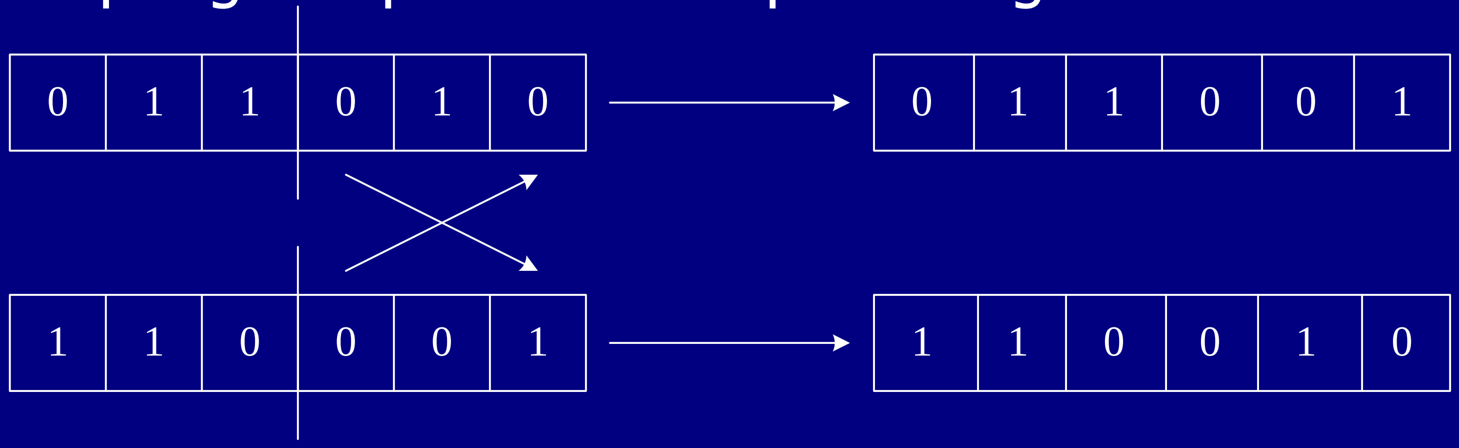
Reproduction

- Individual configurations
  - Copied directly to the next generation
  - Based on their fitness function values
- Configurations with a higher value of fitness function
  - Have higher probability of contributing
  - Usually one or more off-spring copied to next generation
  - Based on biased roulette wheel selection

Genetic Operators

Crossover

- Recombination operator
- Combines subparts of two parent chromosomes
- Offspring has parts of both parents' genetic material



PARENT CHROMOSOMES NEW OFFSPRING

Genetic Operators

Mutation

- Mutation introduces variations into the chromosome
- Randomly alters the value of a string position
- In the string shown below second bit is mutated

0	**1**	1	0	1	1

PARENT CHROMOSOME

→

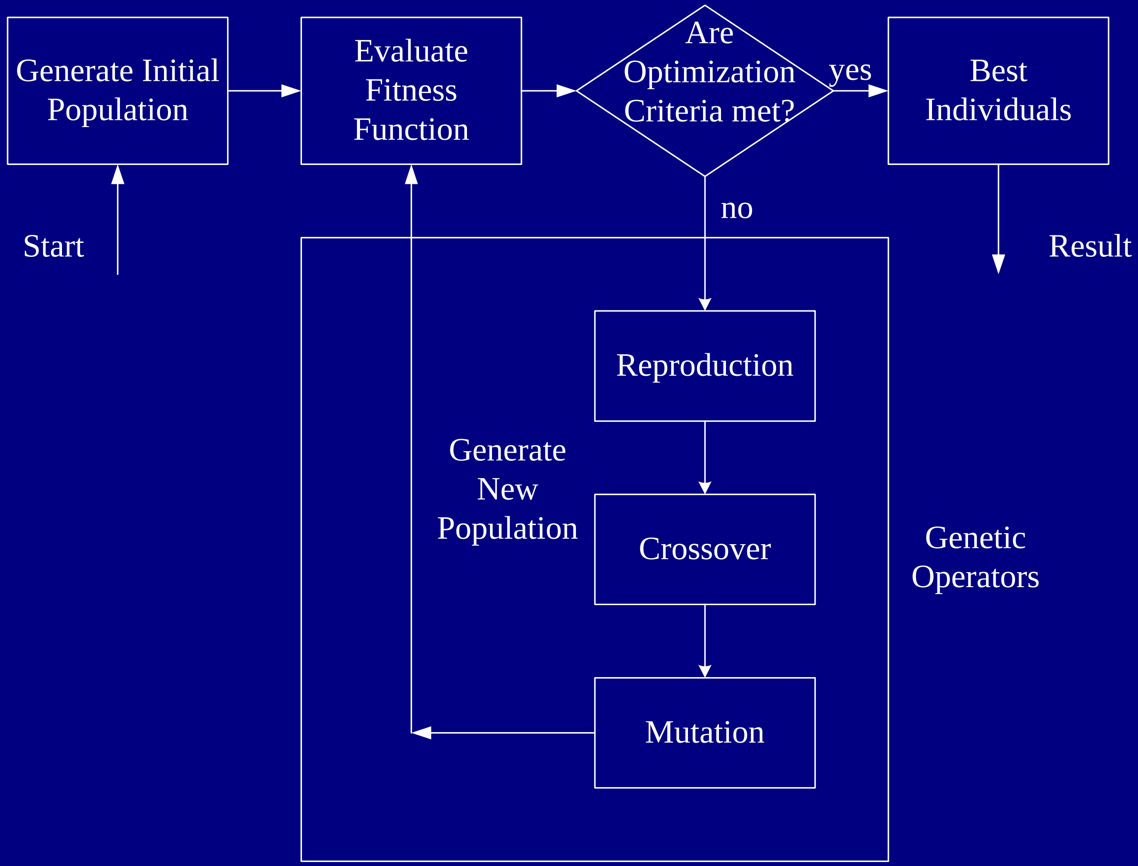
0	**0**	1	0	1	1

MUTATED CHROMOSOME

GA Procedure

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

GA Procedure



Proposed Approach

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Proposed Approach

- Objective of the project
  - Devise an optimization algorithm based on GAs
  - Search for best possible path between end nodes
- The metrics used in determining the best path
  - minimum end-to-end distance
  - minimum latency
  - minimum bit error rate (BER)
  - minimum number of hops
  - maximum bandwidth

Derivation of Fitness Function

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Derivation of Fitness Function

- Each node is a given binary representation
- Chromosome
  - The path with group of binary represented nodes
  - Ex: 001 | 100 | 101 – Chromosome
- Derivation of fitness function
  - Calculate each metric over a particular path
  - Evaluate overall fitness score
- Final fitness score
  - Weighted sum of the individual metrics
  - Path with maximum fitness score is the best path

Derivation of Fitness Function
Example

- Binary representation for 5 node distribution
  - 000, 001, 010, 011, 100 – used for representing 5 nodes
  - 101, 110, 111 – don't care nodes (do not exist in distribution)
  - Don't care nodes keep the chromosome length constant
- Fitness calculation for GA generated example path

  Ex: 000 | 001 | 100 | 101 | 010

  Source |Hops in between | Destination
- Chromosome is intermediate path without end nodes

  001 | 100 | 101

  ø

  Ex: Here hop count = 3

Derivation of Fitness Function
End-to-End Distance (meter)

D is represented as the end-to-end distance for a path

$$D = \sum_{i=0}^{N-1} d_i$$

d_i - Distance between i^{th} and $(i+1)^{th}$ node

N - Number of nodes in the distribution

$$\mathbb{D} = \frac{D}{P}$$

$\mathbb{D}$ - Normalized distance

P - Perimeter of the service area

Derivation of Fitness Function
End-to-End Latency

L is represented as end-to-end latency for a path

$$L = \sum_{i=0}^{N-1} \ell_i$$

ℓ_i - Latency of i^{th} node in a path

N - Number of nodes in the distribution

$$\mathbb{L} = \frac{L}{N \ell_{\max}}$$

$\mathbb{L}$ - Normalized latency

ℓ_{max} - Maximum latency of node distribution

Derivation of Fitness Function
Bit Error Rate

B is represented as aggregate BER over a path

$$B = \sum_{i=0}^{N-1} b_i$$

b_i - BER of the link between i^{th} and $(i+1)^{th}$ node

N - Number of nodes in the distribution

$$\mathbb{B} = \frac{B}{\max(\mathrm{B})}$$

$\mathbb{B}$ - Normalized BER

$\max(\mathrm{B})$- Maximum BER of the node distribution

Derivation of Fitness Function

Bit Error Rate

b_i is BER of the link between i^{th} and $(i+1)^{th}$ node

$$b_i = Q(\sqrt{2\gamma})$$

$\gamma = \frac{C_{pl}.P_t}{N_v}$ is signal-to-noise ratio

C_{pl} - Constant of path loss which is proportional to $\frac{1}{distance^2}$

P_t - Power transmitted

N_v - Noise variance

Derivation of Fitness Function
Number of Hops

H is one less than the number of nodes in a path

$$\mathbb{H} = \frac{H}{N-1}$$

N – Total number of nodes in the distribution
$\mathbb{H}$ – Normalized hop count

Derivation of Fitness Function
Bandwidth (Rate)

R is minimum link bandwidth over all links in a path

$$R = \min(r_i)$$

r_i – Link bandwidth in a particular path

$$\mathbb{R} = \frac{R}{r_{\max}}$$

$\mathbb{R}$ – Normalized bandwidth

$r_{\max}$ – Maximum bandwidth of the node distribution

Derivation of Fitness Function

$$S = W_D(1-\mathbb{D}) + W_L(1-\mathbb{L}) + W_B(1-\mathbb{B}) + W_H(1-\mathbb{H}) + W_R\mathbb{R}$$

- S - Fitness score of a particular path
- D - Normalized end-to-end distance
- L - Normalized latency
- B - Normalized bit error rate
- H - Normalized number of hops
- R - Normalized bandwidth.

W_D, W_L, W_B, W_H, W_R are the weights assigned to each metric

Simulation Results

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- **Simulation Results**
- Research Contribution
- Conclusion
- Future Work

Simulation Results

- Generated random (x, y) locations for nodes
- Exhaustive search
  - Generated all possible paths between end nodes
  - Calculated fitness score over all possible paths
  - Path which yields high fitness score is chosen best path
- GA search
  - Calculated fitness score over paths chosen in generation I
  - New paths (chromosomes) generated using GA operators
  - Fitness score is calculated over new paths
  - Repeated over 150 generations to find path with high score
  - Crossover rate = 0.6 Mutation rate = 0.001 Population = 50

Simulation Results
Test Cases

- Testing with various GA weight vectors $(W_D, W_L, W_B, W_H, W_R)$

  (0.2, 0.2, 0.2, 0.2, 0.2) - Equally weighing

  (1, 0, 0, 0, 0) - Minimizing distance

  (0, 1, 0, 0, 0) - Minimizing latency

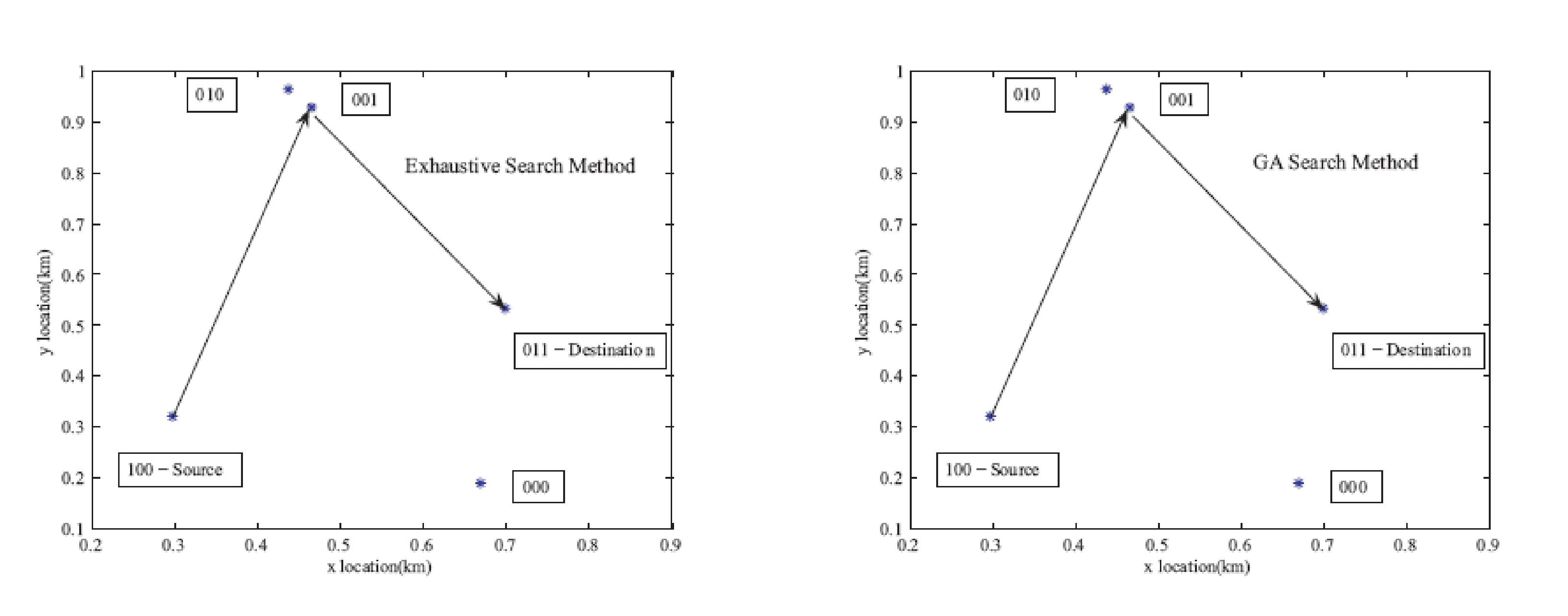
  (0, 0, 1, 0, 0) - Minimizing BER

  (0, 0, 0, 1, 0) - Minimizing hop count

  (0, 0, 0, 0, 1) - Maximizing bandwidth
- Variation of fitness score over generations
- GA over a typical network

Simulation Results

Exhaustive Search vs. GA Search

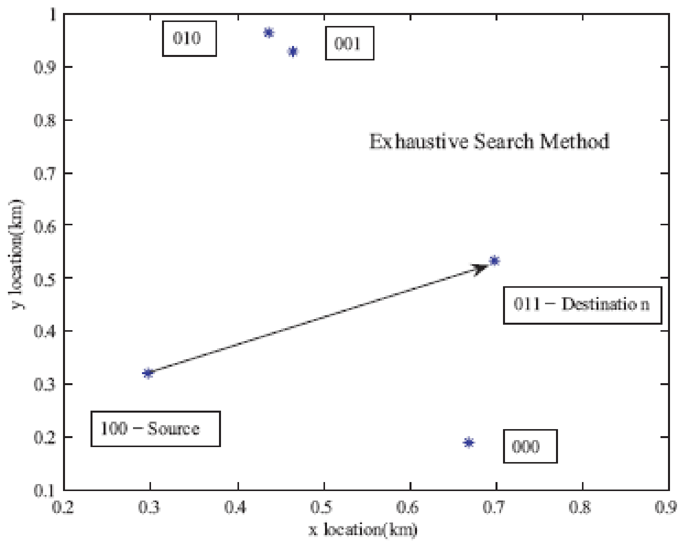


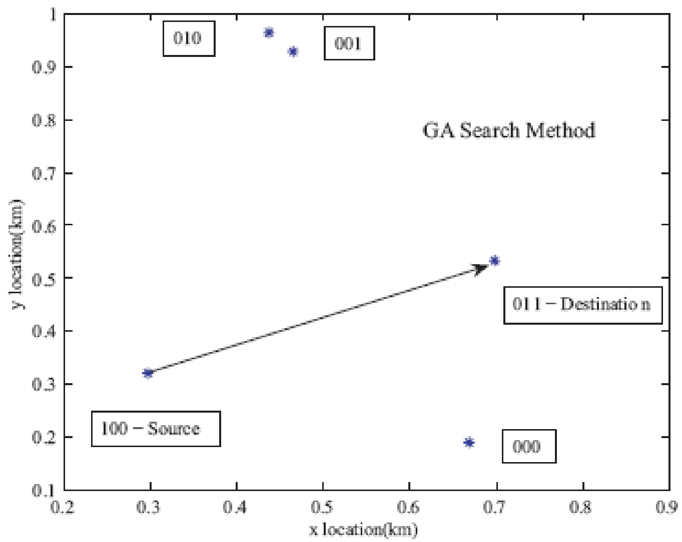
Best Path Equally Weighing Metrics

$(W_D, W_L, W_B, W_H, W_R)$ = (0.2, 0.2, 0.2, 0.2, 0.2)

Simulation Results

Exhaustive Search vs. GA Search



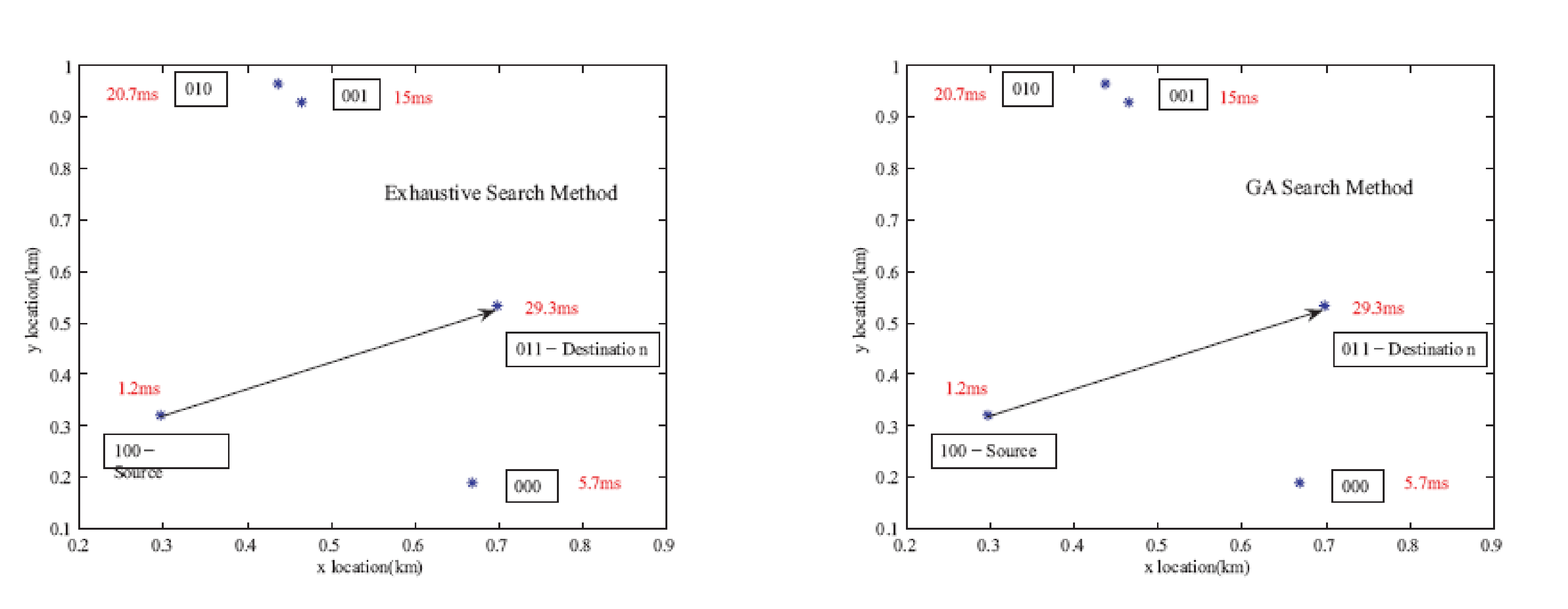


Best Path Minimizing the Distance

$(W_D, W_L, W_B, W_H, W_R)$ = (1, 0, 0, 0, 0)

Simulation Results

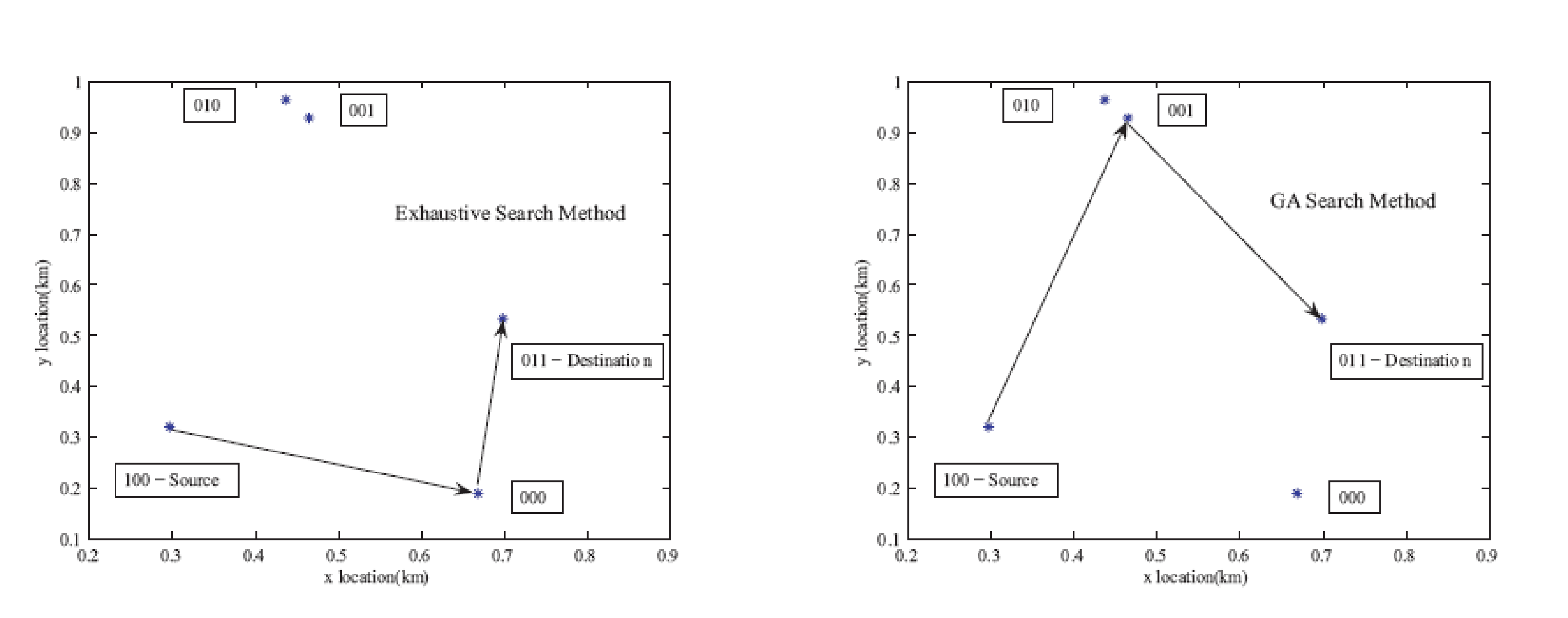
Exhaustive Search vs. GA Search



Best Path Minimizing Latency

$(W_D, W_L, W_B, W_H, W_R) = (0, 1, 0, 0, 0)$

Simulation Results
Exhaustive Search vs. GA Search

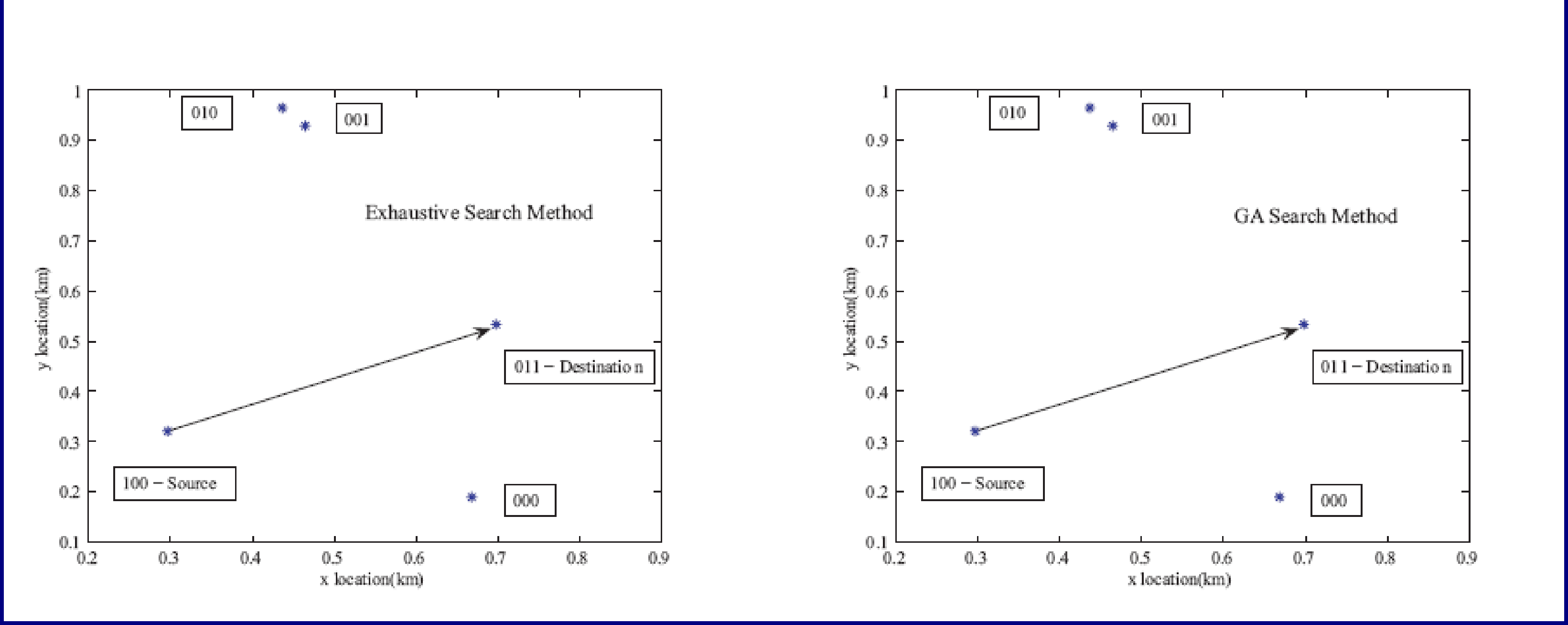


Best Path Minimizing BER

$(W_D, W_L, W_B, W_H, W_R)$ = (0, 0, 1, 0, 0)

Simulation Results

Exhaustive Search vs. GA Search

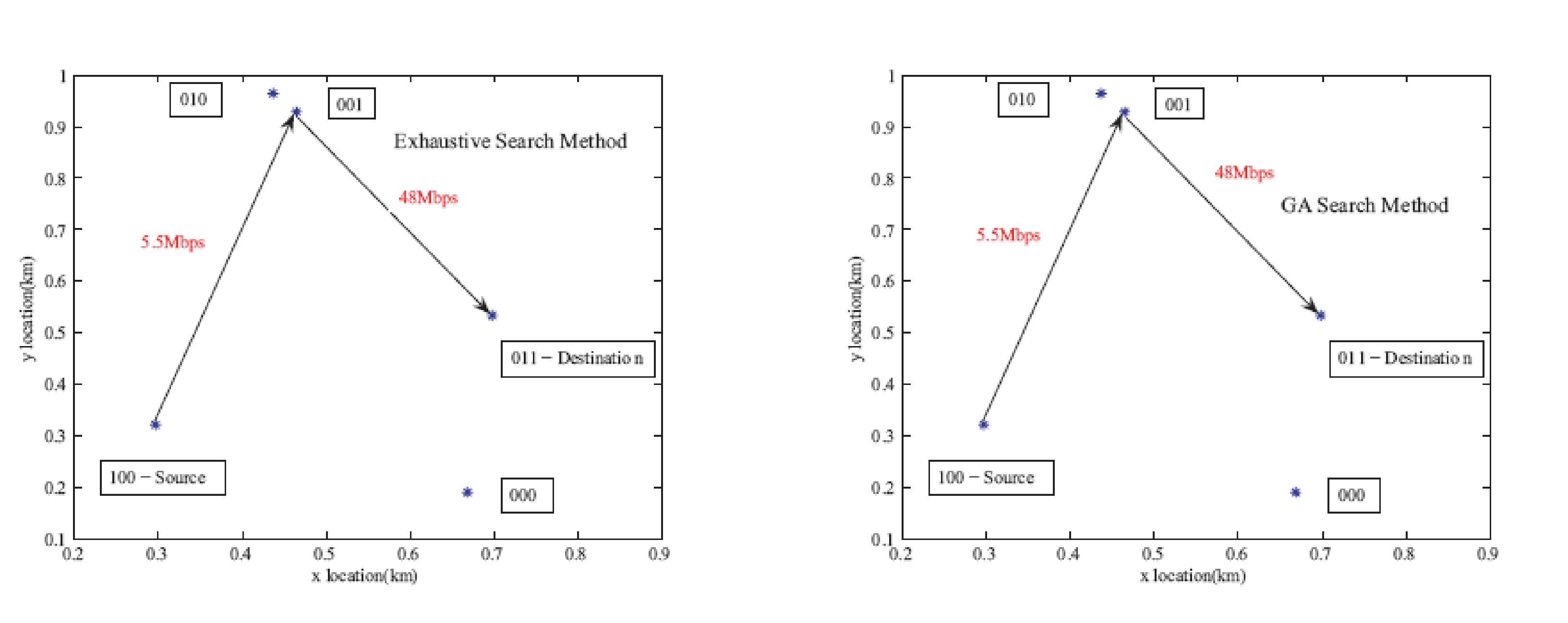


Best Path Minimizing Number of Hops

$(W_D, W_L, W_B, W_H, W_R)$ = (0, 0, 0, 1, 0)

Simulation Results

Exhaustive Search vs. GA Search

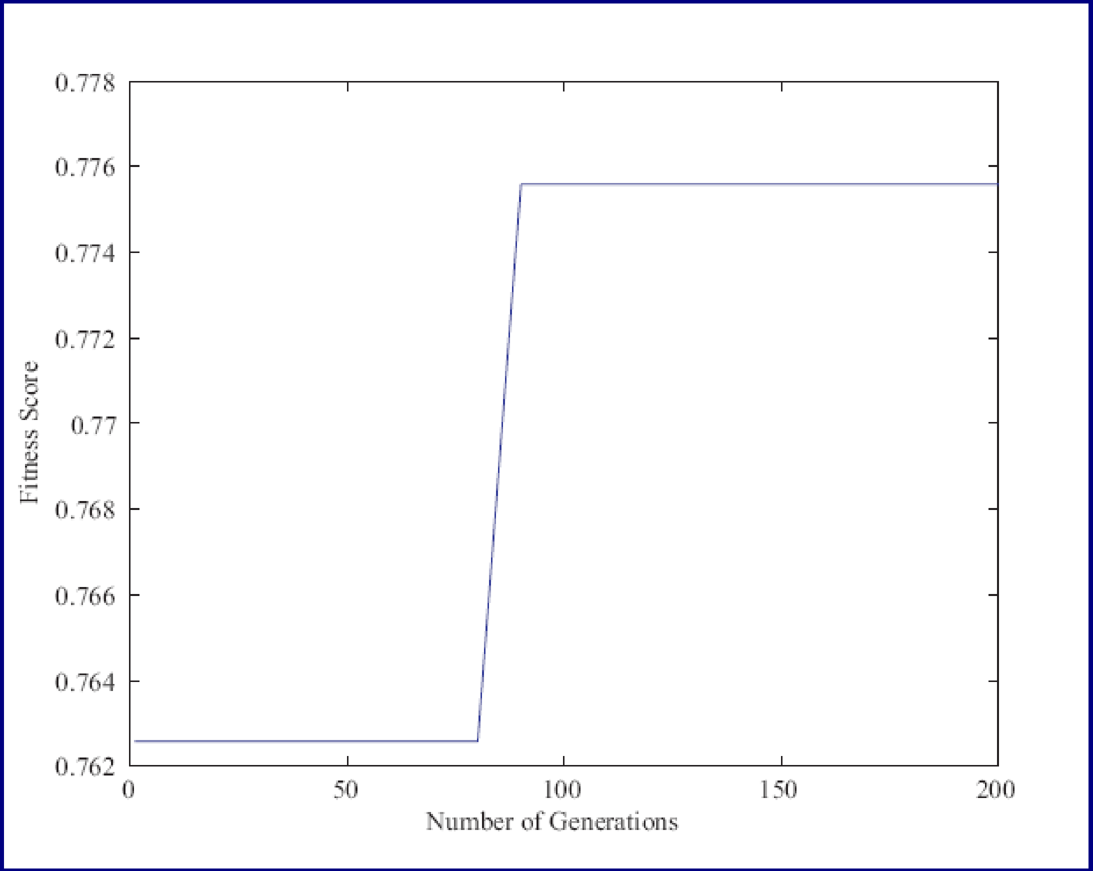


Best Path Maximizing Bandwidth

$(W_D, W_L, W_B, W_H, W_R)$ = (0, 0, 0, 0, 1)

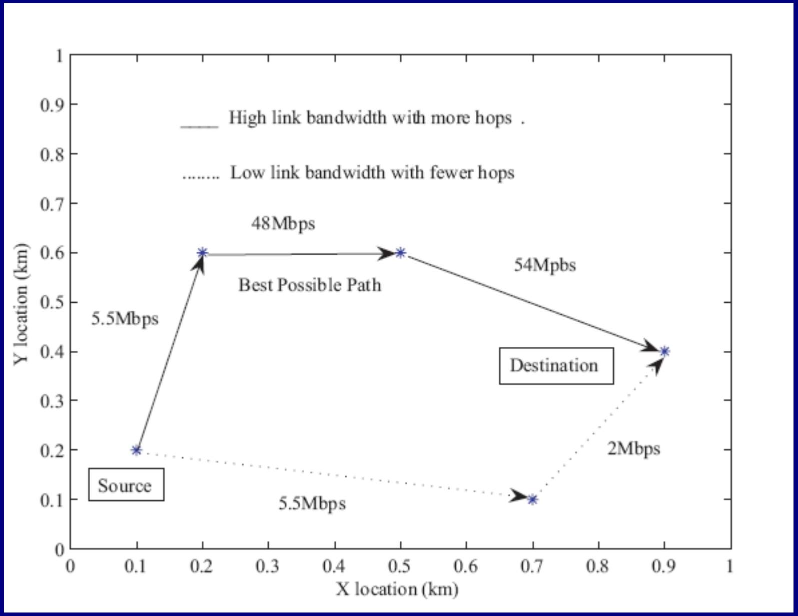
Simulation Results

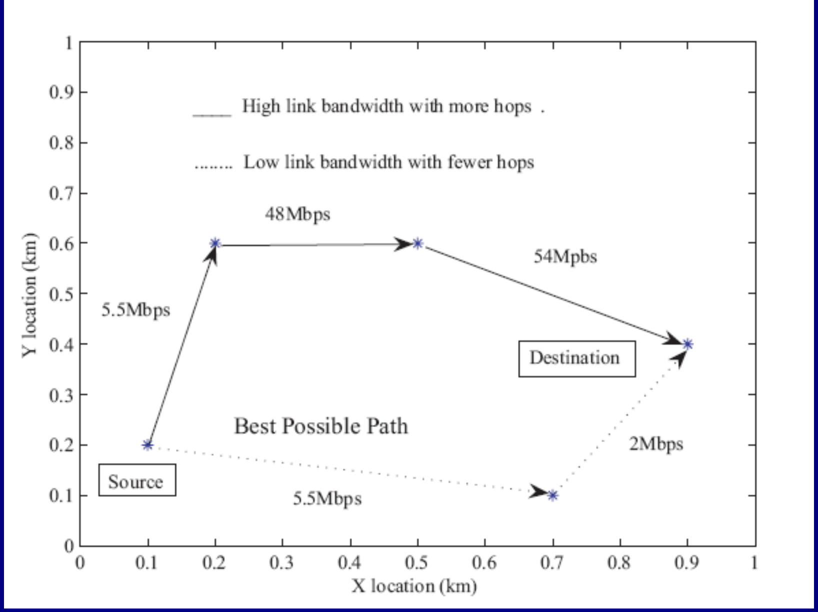
Variation of Fitness Score Over Generations



Simulation Results

GA Performance Over a Typical Network





(0, 0, 0, 0.2, 0.8)

$$(W_D, W_L, W_B, W_H, W_R)$$

(0, 0, 0, 0.5, 0.5)

$$(W_D, W_L, W_B, W_H, W_R)$$

Research Contribution

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Research Contribution

- Node distribution
  - Generated using ‘C’ numerical recipes
- Exhaustive search
  - Code written in C
- GA framework has been implemented
  - Sga-c source code available at IlliGAL Institute http://www.illigal.uiuc.edu/web/
  - Modified to work for the proposed approach
- Fitness function for multi-objective optimization

Conclusion

- Introduction
- Genetic Algorithms (GA) Overview
- GA Operators
- GA Procedure
- Proposed Approach
- Derivation of Fitness Function
- Simulation Results
- Research Contribution
- Conclusion
- Future Work

Conclusion

- The proposed framework
  - Useful for multiple metric optimization in routing
  - Weight factors can be adjusted to match user's requirement
- Best path
  - GA results compare favorably with exhaustive search
- Exhaustive search vs. GA search
  - GA takes lesser time compared to exhaustive search
  - GA searches for best path using fewer configurations
  - Exhaustive search evaluates fitness over all configurations

Future Work

Future Work

- Introduce time-variant node metrics
- Unreachable nodes
- More network topologies
- Larger networks
- Multiple source and destination nodes

Thank You

Questions???