Relative Lokalisierung in Sensornetzen

Raphael Schmid

Seminar „Lokalisierung“ bei Prof. F. Mattern, SS06

Betreuung: Christian Frank

Inhalt des Vortrages

- Einführung und Motivation
- Übersicht über relative Lokalisierung
- Algorithmus von Priyantha, Balakrishnan, Demaine, Teller (MIT Laboratory for Computer Science, 2003)

Einführung und Motivation

- Viele Anwendungen wo Lokalisierung nötig ist, aber GPS nicht möglich ist
  - Geometrisches Routing
  - Chemie-Labor
  - Relative Messungen
  - Informationsaustausch in naher Umgebung
  - „Closest-Server-Suche“

Übersicht über relative Lokalisierung

- Die drei bekanntesten Probleme
  - Nachbarn sind bekannt
  - Distanzen sind bekannt
  - Winkel sind bekannt

- Unit Disk Graph Einbettung: NP-hart! [2]

Übersicht über relative Lokalisierung

- Verschiedene Arten von Algorithmen
  - Inkrementelle Algorithmen
  - Parallele/lokale Algorithmen
  - Anchor-basierte Algorithmen
  - Anchor-freie Algorithmen

Übersicht über relative Lokalisierung - Inkrementelle Algorithmen

- Nachteile:
  - Fehler werden aufaddiert!
  - Globale Optimierungen enden sehr oft in lokalem Minimum

Übersicht über relative Lokalisierung - Parallel/lokale Algorithmen

- Vorteile:
  - Schneller Ablauf, da alles parallel
  - Lokale Fehler werden nicht aufaddiert
  - Dezentral

Übersicht über relative Lokalisierung - Anchor-basierte Algorithmen

- Nachteile:
  - Fehler von Anchors werden auf alle Knoten übertragen
  - Skaliert nicht, falls man obigen Problem entgegen treten will
  - Bei GPS: Import aller GPS-Problemen
  - Probleme bei Ausfällen der Anchors
- Vorteile:
  - Trotz globalen Koordinaten niedrige Kosten
  - Anchors helfen um lokale Minima zu vermeiden

Übersicht über relative Lokalisierung - Anchor-freie Algorithmen

- Vorteile:
  - Keine zusätzlichen Kosten für Anchors
  - Dezentral
  - Jeder Anchor-freie Algorithmus kann in Anchor-basierter Algorithmus umgewandelt werden
- Nachteile:
  - Grundsätzlich grosse Probleme mit lokalen Minima
  - Keine globalen Koordinaten

Relative Lokalisierung konkret: Algorithmus von Priyantha, Balakrishnan, Demaine, Teller [1]

- Ziel: $d'_{ij} = r_{ij}, \forall i,j \in V$
- Distanzmessungen mittels Funk und/oder Schall
- Keine Beweise für
  - Wann und wie oft Algorithmus in einem lokalen Minimum endet
  - Genauigkeit des Algorithmus

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Übersicht

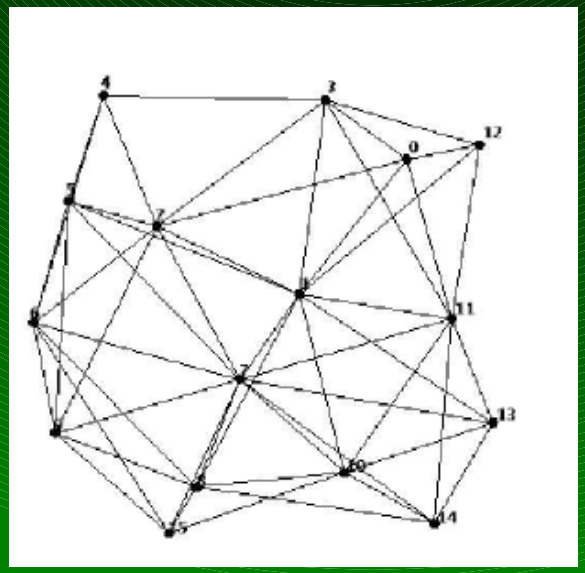
- Teil 1: Graph-Einbettung, „ähnlich“ zum ursprünglichen Graph
- Teil 2: Masse-Feder basierte Optimierung

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller

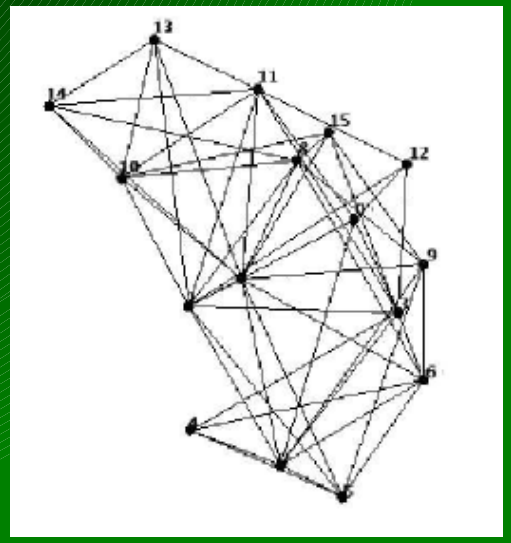
- Probleme:
  - Falsche (lokale) Minima
  - Graph muss rigid sein
  - Performance Metrik

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Probleme

- Falsche (lokale) Minima
  - Meistens als Folge von Masse-Feder Optimierungen
  - Provoziert durch „Faltung“



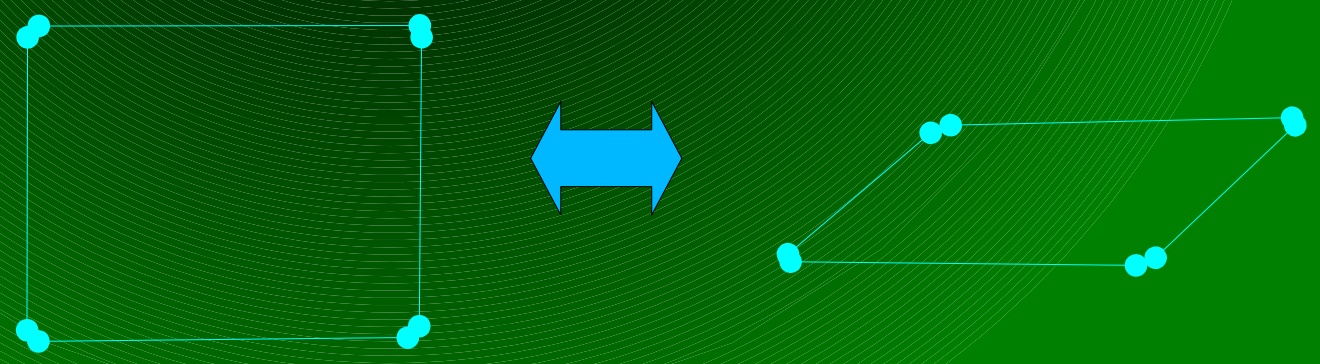
[1] Fig.7



[1] Fig.8

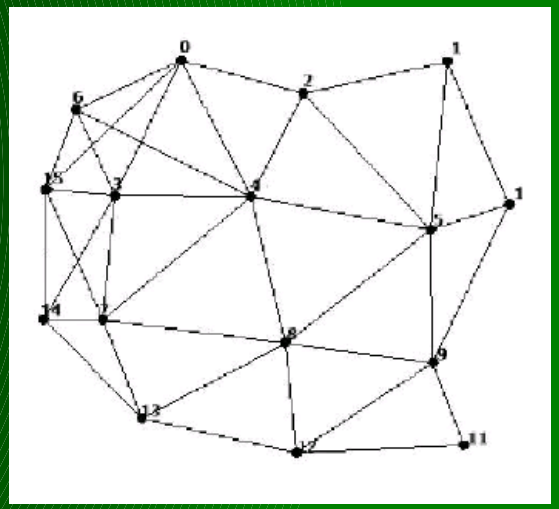
Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Probleme

- Graph muss rigid sein



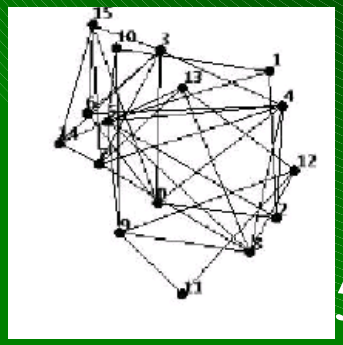
Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Probleme

- Performance Metrik



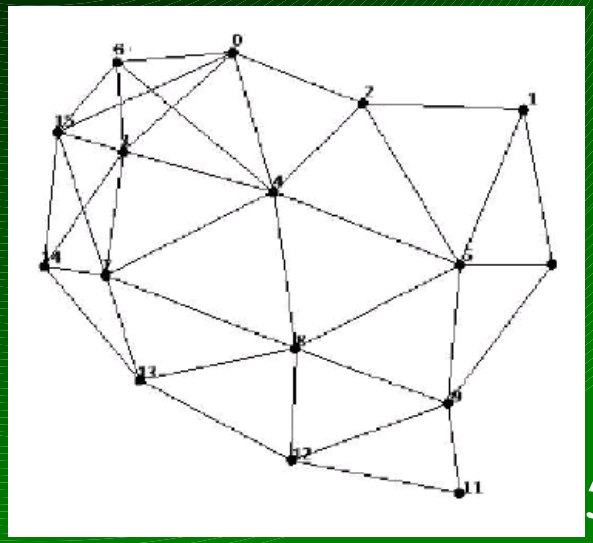
Original

[1] Fig.2



5% error

[1] Fig.3



5% error

[1] Fig.4

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Lösungen

- Falsche (lokale) Minima
  - > Lösung: Fold-freedom

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Lösungen

- Graph muss rigid sein
  - > Connectivity!
- Hendrickson [3] : Graph ist (d+1)-connected -> global rigid
  - Connelly [4] : Stimmt nicht unbedingt bei 3D!
  - Hendrickson: Stimmt aber bei 2D

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Lösungen

- Performance Metrik: Global Energy Ratio / Global Error Ratio (GER)

$$GER = \frac{\sqrt{\sum_{i=0}^{N-1} \sum_{j=i}^{N-1} e'^{2}_{ij}}}{N(N-1)/2}$$

$$e'_{ij} = \frac{d'_{ij} - d_{ij}}{d_{ij}}$$

Teil 1: Graph-Einbettung

1. Wähle n_0; wähle n_1 so dass $h_{0,1}$ möglichst gross
2. Wähle n_2 so dass $h_{1,2}$ möglichst gross
3. Wähle n_3 so dass $(h_{1,3} - h_{2,3})$ möglichst klein und $(h_{1,3} + h_{2,3})$ möglichst gross
4. Wähle n_4 so dass $(h_{1,4} - h_{2,4})$ möglichst klein und $h_{3,4}$ möglichst gross
5. Wähle n_5 so dass $(h_{1,5} - h_{2,5})$ möglichst klein und $(h_{3,5} - h_{4,5})$ möglichst klein

Teil 1: Graph-Einbettung

6. Berechne Koordinaten für alle Knoten:

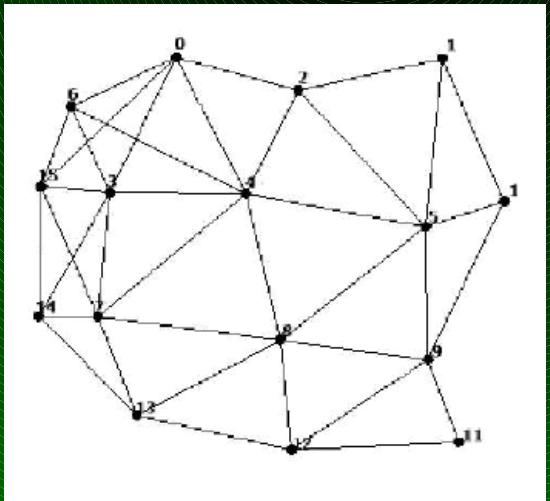
$$\rho_i = h_{5,i} * R$$

$$\theta_i = \tan^{-1}\left(\frac{h_{1,i} - h_{2,i}}{h_{3,i} - h_{4,i}}\right)$$

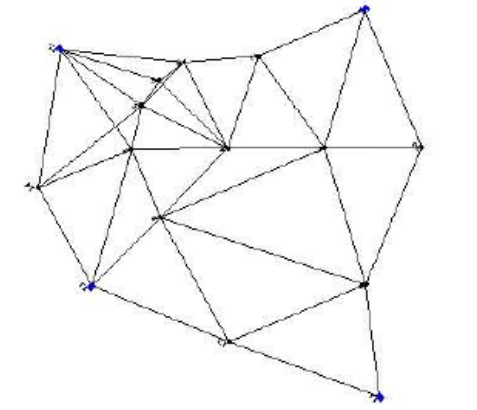
Teil 1: Graph-Einbettung

- Beispiel:

Original-Graph



[1] Fig.2



[1] Fig.14

Graph nach Algorithmus Teil 1

Teil 2: Masse-Feder basierte Optimierung

1. Sende eigene aktuelle Position an alle Nachbarn
2. Falls $F_i = 0$, terminiere
3. Berechne neue Position, sobald neue Position von Nachbarn empfangen:
   1. $F_{i,j} = v'_{i,j}(d'_{i,j} - r_{i,j})$
   2. $F_i = \sum_{i,j} F_{i,j}$
   3. $F'_i = F_i / (2\mathrm{m}_i)$
4. Gehe zu 1.

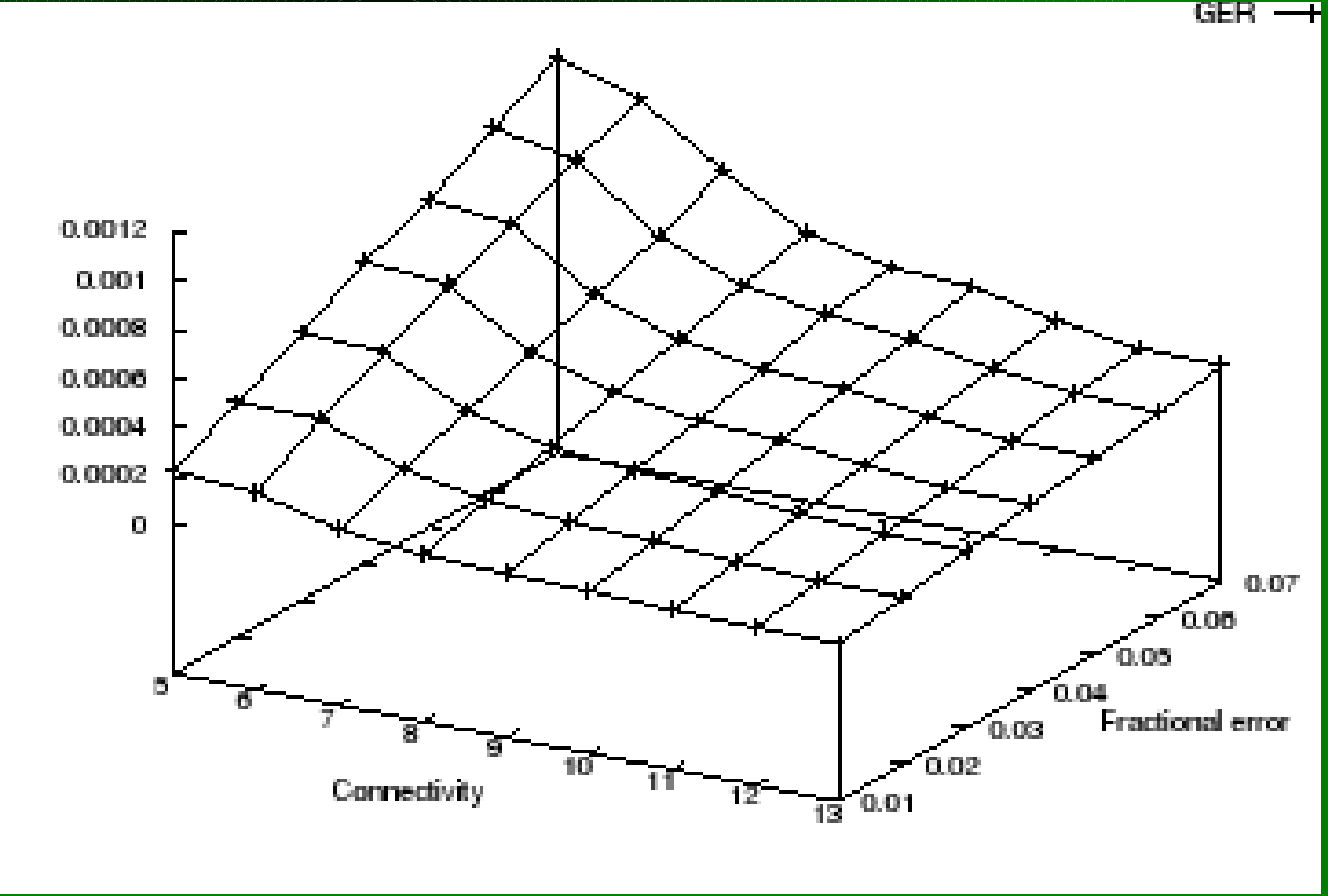
$$E_i = \sum_j E_{i,j} = \sum_j (d'_{i,j} - r_{i,j})^2$$

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Rückblick

- Teil 1: Graph-Einbettung
- Teil 2: Masse-Feder basierte Optimierung

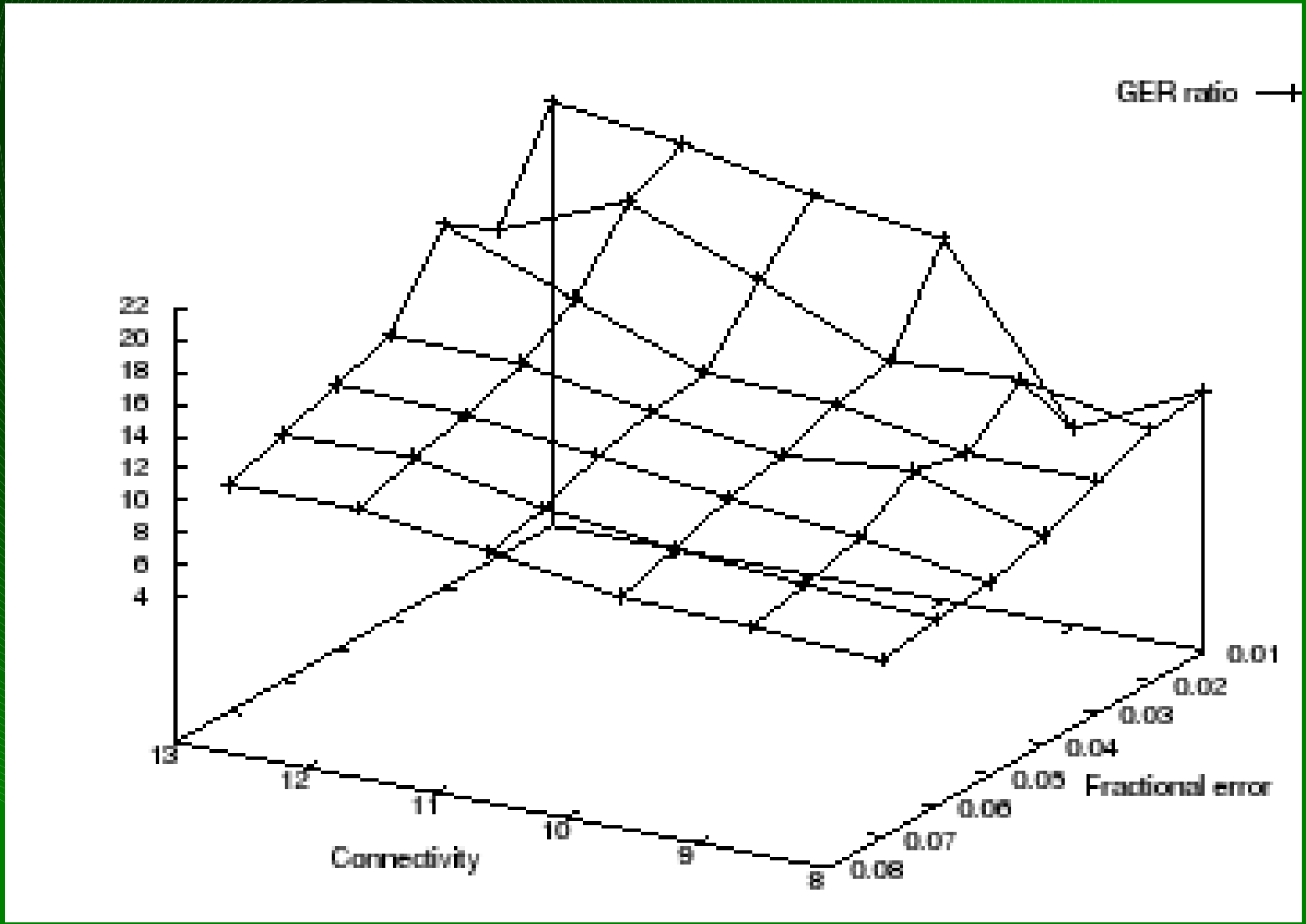
- Probleme: Lokale Minima, „unstabile“ Graphen, Performance-Metrik

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Resultate



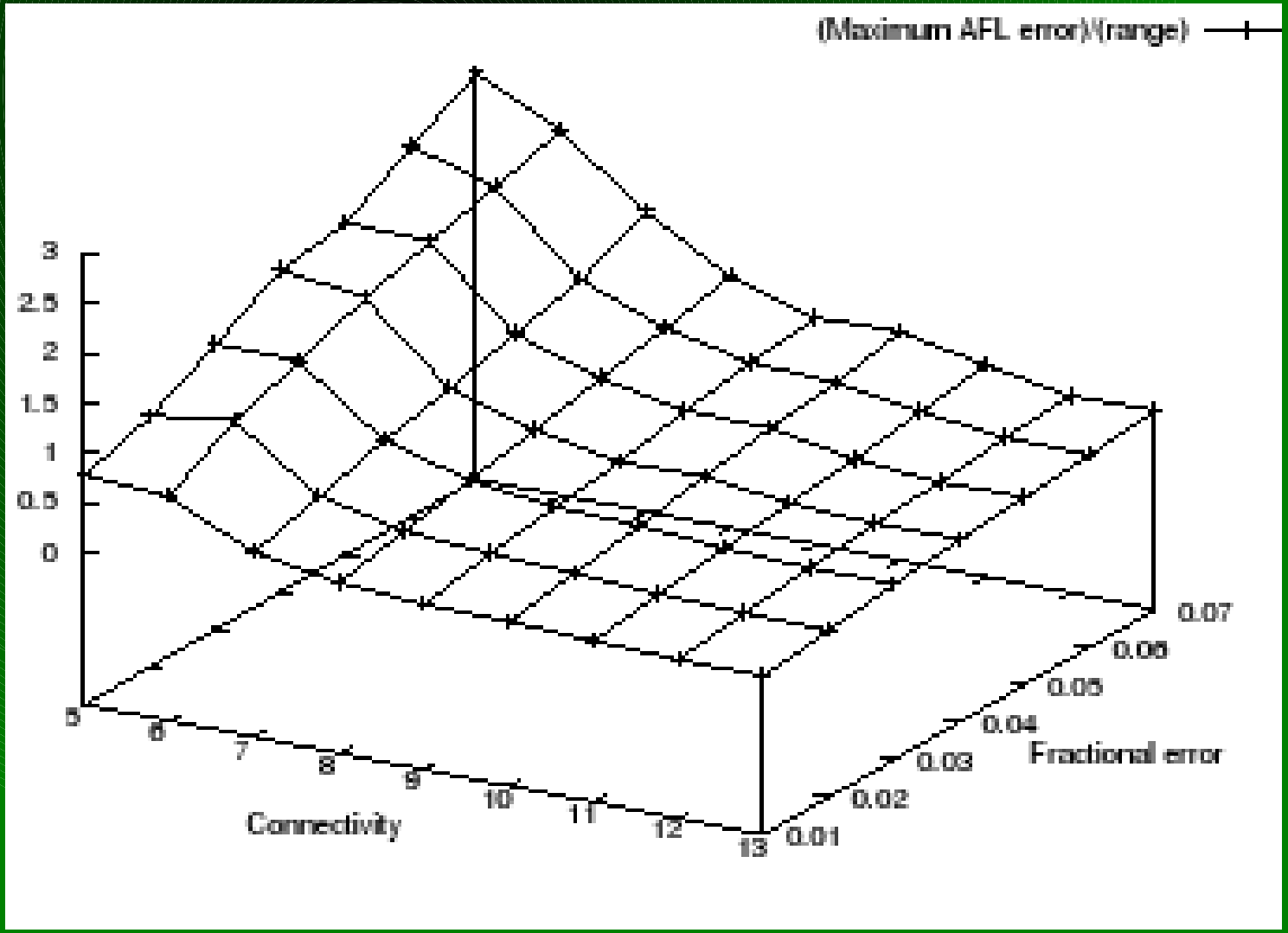
[1] Fig.17

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Resultate



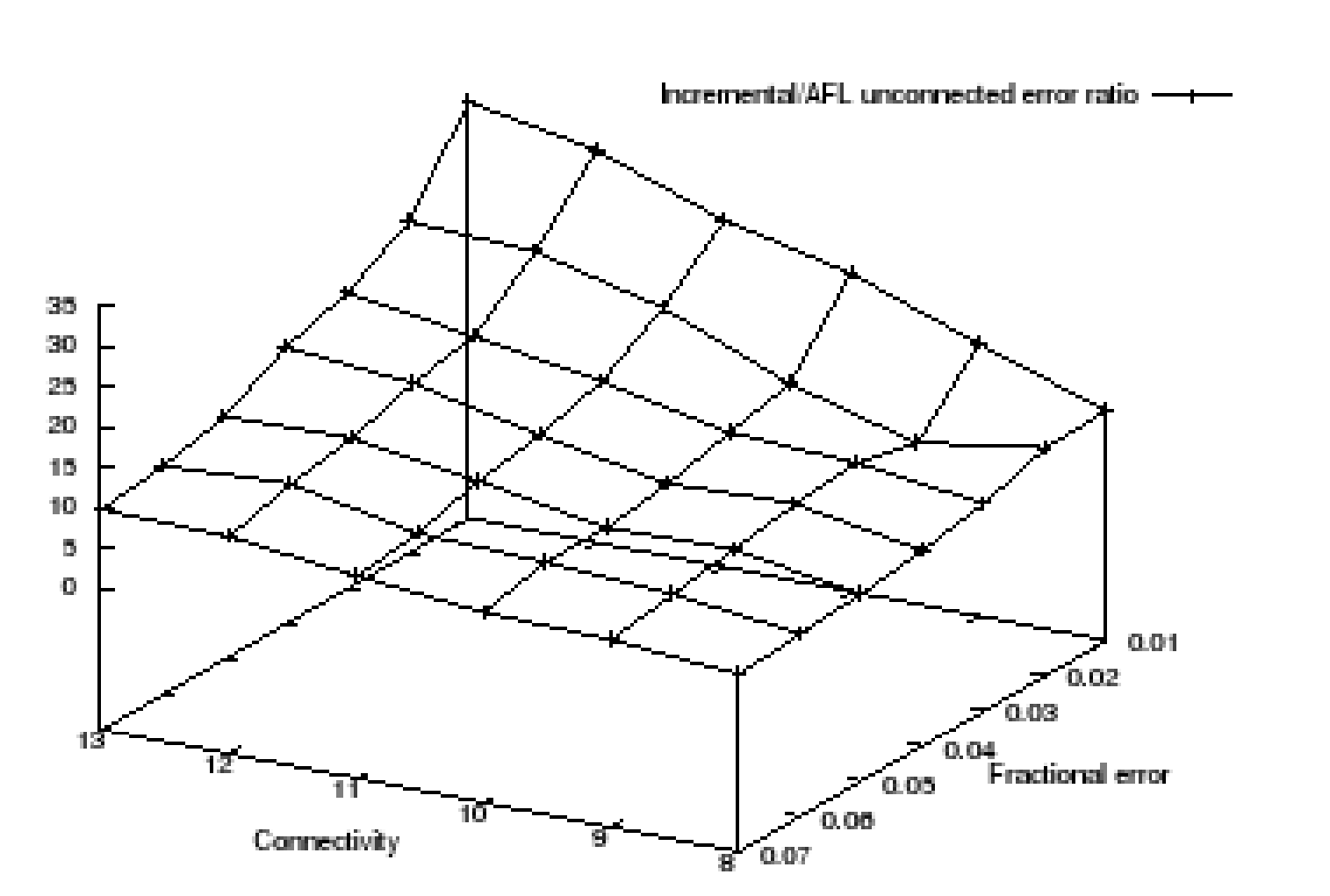
[1] Fig.18

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Resultate



[1] Fig.19

Algorithmus von Priyantha, Balakrishnan, Demaine, Teller - Resultate



[1] Fig.20

(Persönliche) Schlussfolgerungen

- Sehr schöne und gute Idee
- Keine Beweise, „nur“ Simulationen
- GER: nicht messbar in der Realität
- Hohe Connectivity nicht realistisch

Literaturverzeichnis

[1] Anchor-Free Distributed Localization in Sensor Networks [http://nms.lcs.mit.edu/cricket/papers/TechReport892.pdf

[2] On the Computational Complexity of Sensor Network Localization [http://www.dcg.ethz.ch/members/pascal/refs/pos_2004_aspnes.pdf]

[3] HENDRICKSON , B . Conditions for unique graph realizations. SIAM Journal on Computing 21, 1 (1992), 65–84.

[4] CONNELLY, R . On generic global rigidity. In Applied Geom-etry and Discrete Mathematics. American Mathematical Society, 1991, pp. 147–155.