Vorlesung Algorithmische Kartografie

Linienvereinfachung II

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Benjamin Niedermann · **Martin Nöllenburg**

21.04.2015

Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

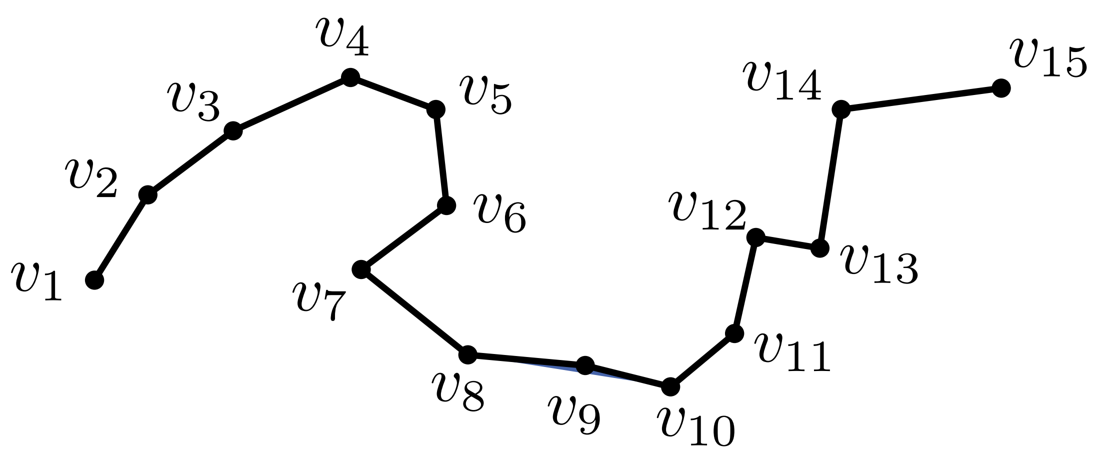
[Visvalingam, Whyatt '93]

Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

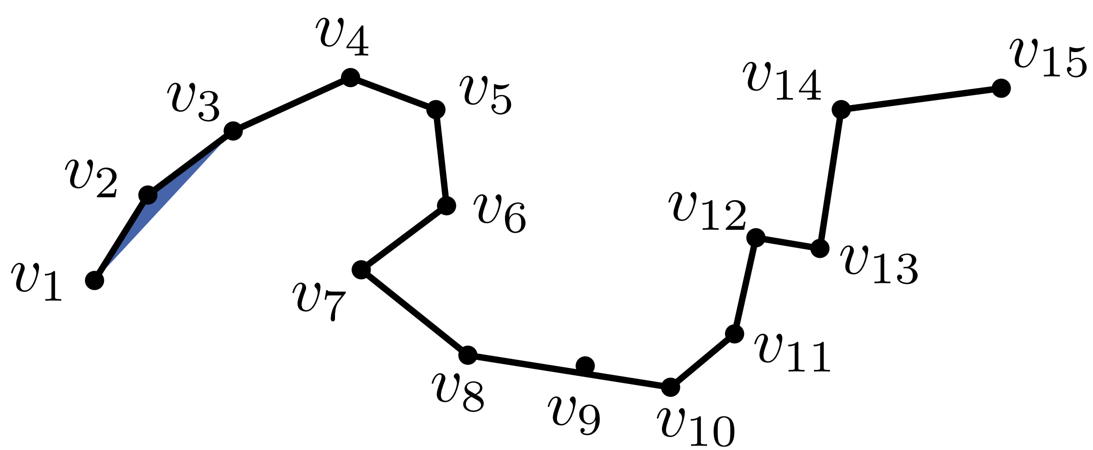


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

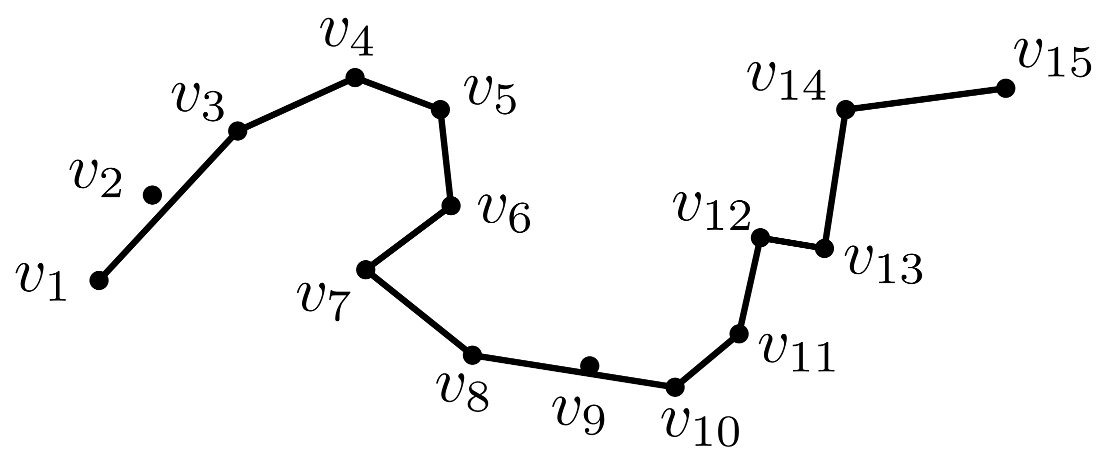


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

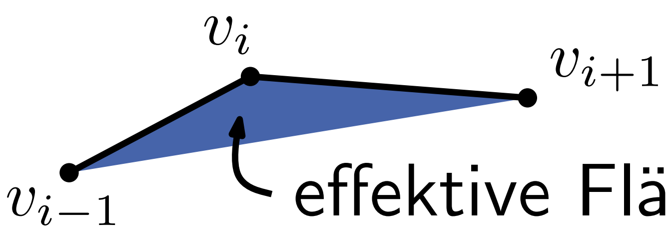


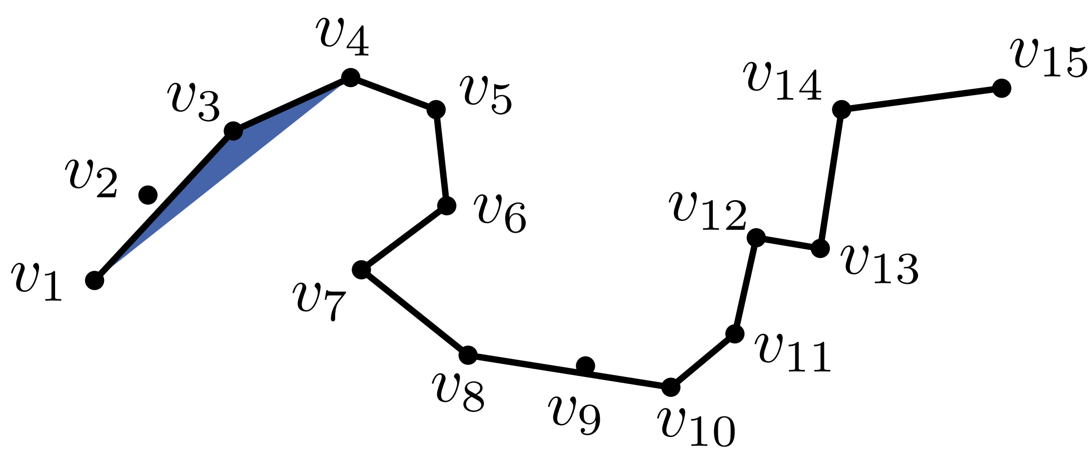
Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]



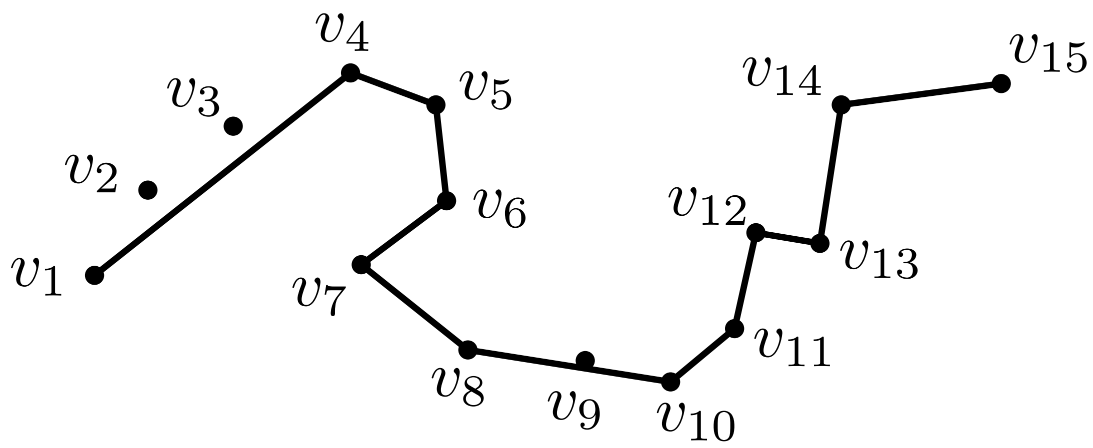


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

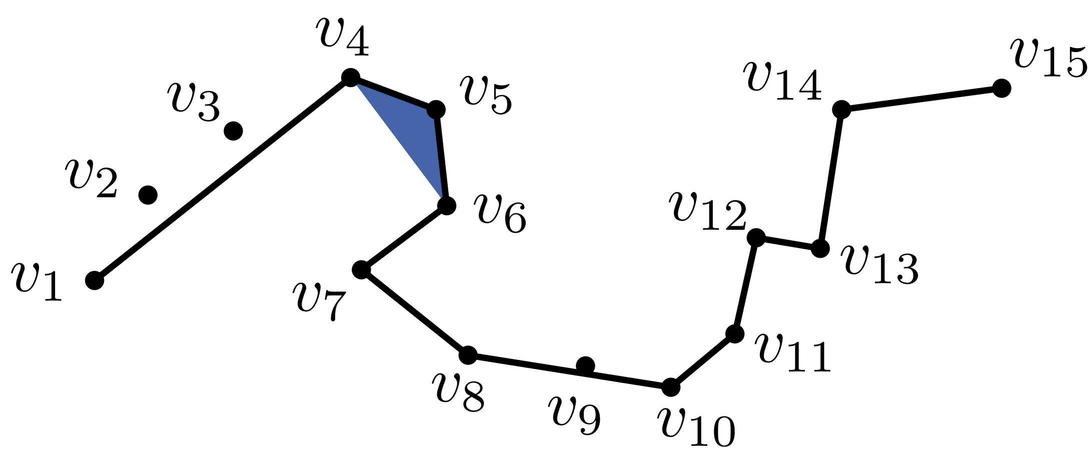


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

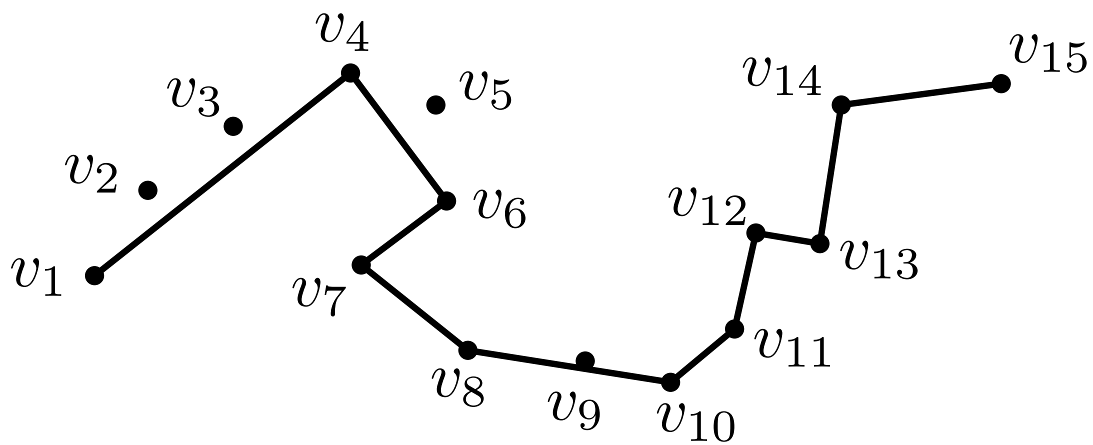


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

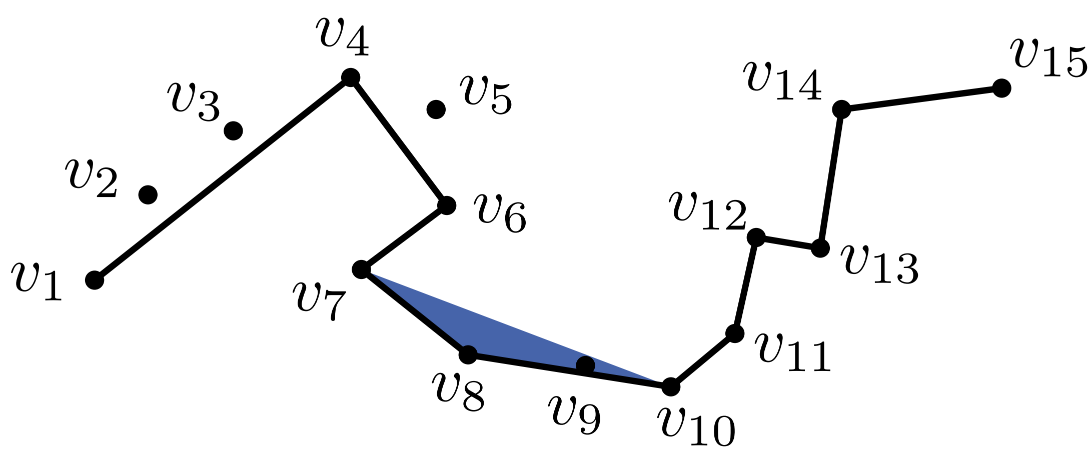


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt ’93]

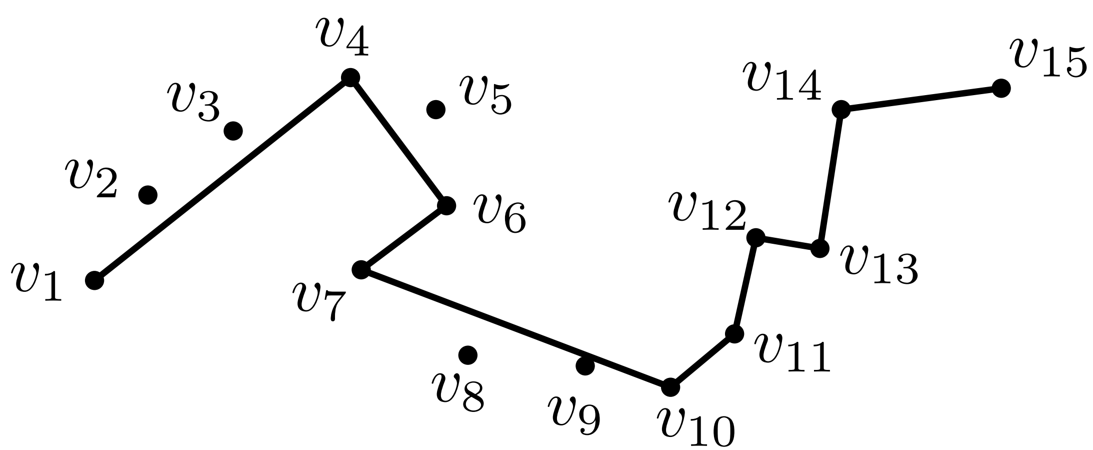


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

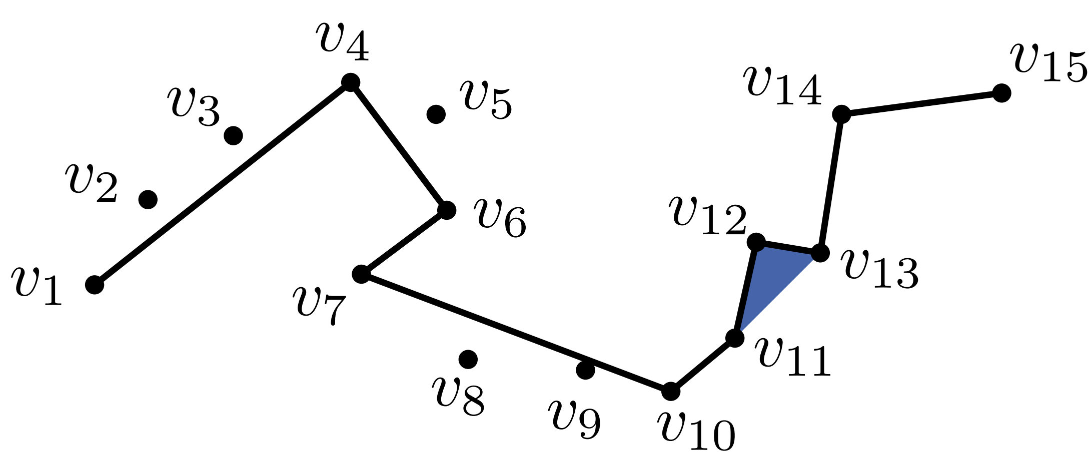


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

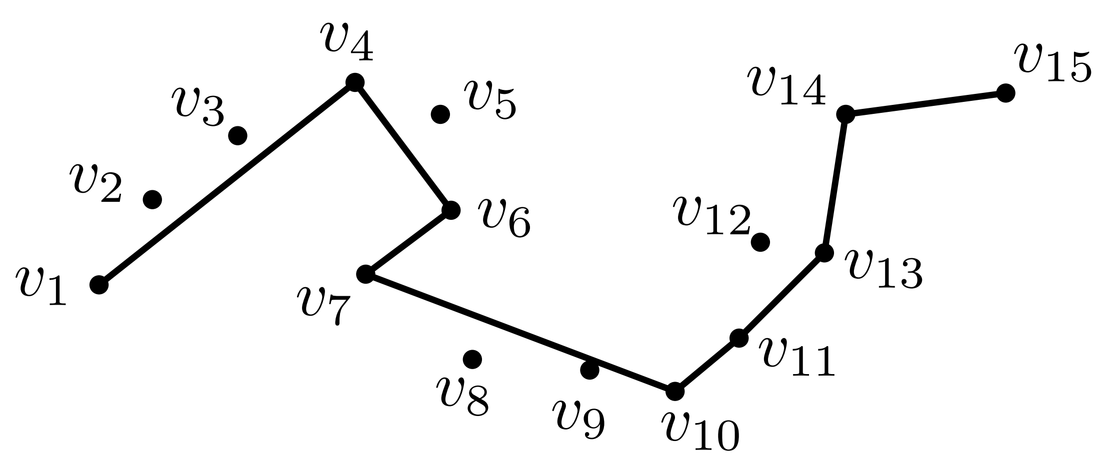


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]

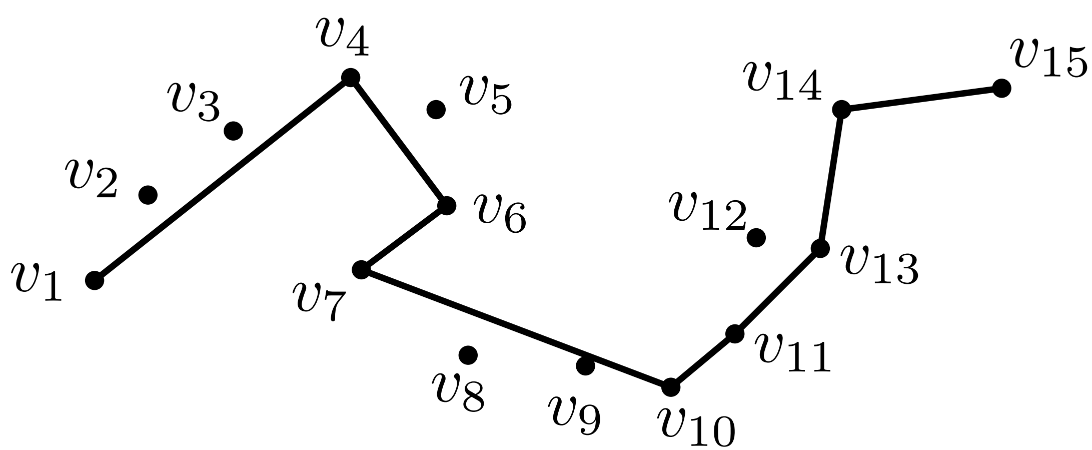


Alternative Punktauswahl

Entferntesten Punkt wie bei DP zu behalten kann Ausreißer betonen.

Betrachte stattdessen **effektive Fläche** jedes Knotens und entferne iterativ Knoten mit kleinster Fläche.

[Visvalingam, Whyatt '93]



Implementierung, Laufzeit?

Vereinfachung als Optimierungsproblem

Bisherige Algorithmen liefern heuristisch gute Approximationen an die Eingabe, jedoch nicht notwendig eine optimale Lösung.

Formulierung als Graphenproblem:

Fall 1: feste Maximallänge, minimiere Fehler

Vereinfachung als Optimierungsproblem

Bisherige Algorithmen liefern heuristisch gute Approximationen an die Eingabe, jedoch nicht notwendig eine optimale Lösung.

Formulierung als Graphenproblem:

Fall 1: feste Maximallänge, minimiere Fehler

- vollständiger, gewichteter DAG G auf $\{v_1, \dots, v_n\}$ mit Kante (v_i, v_j) für alle $i < j$
- Kosten c_{ij} für das Ersetzen des Teilpfads $(v_i, v_{i+1}, \dots, v_j)$ durch Kante (v_i, v_j)
- finde kostenminimalen (kürzesten) Weg von v_1 nach v_n in G mit maximal k Kanten

Vereinfachung als Optimierungsproblem

Bisherige Algorithmen liefern heuristisch gute Approximationen an die Eingabe, jedoch nicht notwendig eine optimale Lösung.

Formulierung als Graphenproblem:

Fall 1: feste Maximallänge, minimiere Fehler

- vollständiger, gewichteter DAG G auf $\{v_1, \ldots, v_n\}$ mit Kante (v_i, v_j) für alle $i < j$
- Kosten c_{ij} für das Ersetzen des Teilpfads $(v_i, v_{i+1}, \ldots, v_j)$ durch Kante (v_i, v_j)
- finde kostenminimalen (kürzesten) Weg von v_1 nach v_n in G mit maximal k Kanten

Solche „constrained shortest path“ Probleme sind i. Allg. NP-schwer, hier jedoch nicht.

$\rightarrow$ mehr dazu gleich!

Vereinfachung als Optimierungsproblem

Bisherige Algorithmen liefern heuristisch gute Approximationen an die Eingabe, jedoch nicht notwendig eine optimale Lösung.

Formulierung als Graphenproblem:

Fall 2: fester maximaler Fehler, minimiere Länge [Imai, Iri '88]

Vereinfachung als Optimierungsproblem

Bisherige Algorithmen liefern heuristisch gute Approximationen an die Eingabe, jedoch nicht notwendig eine optimale Lösung.

Formulierung als Graphenproblem:

Fall 2: fester maximaler Fehler, minimiere Länge [Imai, Iri ’88]

- DAG $G = (V, A)$ mit $V = \{v_1, \ldots, v_n\}$ und Kante $(v_i, v_j) \in A$, $i < j$, gdw. Fehler($\overline{v_i v_j}$) $< \varepsilon$
- finde kürzesten Weg von v_1 nach v_n in G
- topologisches Sortieren in linearer Zeit
- Berechnung von A naiv in $O(n^3)$, optimal in $\Theta(n^2)$

[Chan, Chin ’92]

Übungen

1) Der Douglas-Peucker-Algorithmus garantiert, dass bei der Vereinfachung eines Polygonzugs $P = (p_1, \ldots, p_n)$ in einen Polygonzug Q, der Abstand zwischen P und Q höchstens ε ist. Berechnet der Algorithmus dabei immer ein Q mit minimal vielen Segmenten?

Übungen

1) Der Douglas-Peucker-Algorithmus garantiert, dass bei der Vereinfachung eines Polygonzugs $P = (p_1, \dots, p_n)$ in einen Polygonzug Q, der Abstand zwischen P und Q höchstens ε ist. Berechnet der Algorithmus dabei immer ein Q mit minimal vielen Segmenten?

2) Gegeben sei ein azyklischer, gerichteter Graph $G = (V, E)$ mit Gewichtsfunktion $w \colon E \to \mathbb{R}$. Ein gerichteter st-Pfad $(s = v_0, \dots, v_\ell = t)$ in G heißt k-*minimal*, falls $\ell \leq k$ und für alle anderen st-Pfade $(s = v'_0, \dots, v'_{\ell'} = t)$ mit $\ell' \leq k$ gilt: $\sum_{j=1}^{\ell} w((v_{j-1}, v_j)) \leq \sum_{j=1}^{\ell'} w((v'_{j-1}, v'_j))$.

Entwerfen und analysieren Sie einen möglichst effizienten Algorithmus zur Berechnung von k-minimalen st-Pfaden.