Färbeprobleme und Abschätzung dünnbesetzter Jacobimatrizen

Daniela Dossing

14.08.2006

Sommerschule des Graduiertenkollegs
„Hierarchie und Symmetrie in mathematischen Modellen“
Universitätskolleg Bommerholz, Witten

Übersicht

Motivation

Nicht-lineare Probleme in der Analysis und Optimierung

Motivation

Nicht-lineare Probleme in der Analysis und Optimierung

(1.1) Beispiele

Gegeben sei eine (Vektor-)Funktion f durch $f : \mathbb{R}^n \to \mathbb{R}^m$ mit Komponenten $f_1(x), \ldots, f_m(x)$ und $x \in \mathbb{R}^n$.

Motivation

Nicht-lineare Probleme in der Analysis und Optimierung

(1.1) Beispiele

Gegeben sei eine (Vektor-)Funktion f durch $f : \mathbb{R}^n \to \mathbb{R}^m$ mit Komponenten $f_1(x), \ldots, f_m(x)$ und $x \in \mathbb{R}^n$.

- Lösen eines nicht-linearen Gleichungssystems $f(x) = 0$

Motivation

Nicht-lineare Probleme in der Analysis und Optimierung

(1.1) Beispiele

Gegeben sei eine (Vektor-)Funktion f durch $f : \mathbb{R}^n \to \mathbb{R}^m$ mit Komponenten $f_1(x), \ldots, f_m(x)$ und $x \in \mathbb{R}^n$.

- Lösen eines nicht-linearen Gleichungssystems $f(x) = 0$
- Minimierung der Quadratsumme $\sum_{i=1}^{m} (f_i(x))^2$

Motivation

Nicht-lineare Probleme in der Analysis und Optimierung

(1.1) Beispiele

Gegeben sei eine (Vektor-)Funktion f durch $f : \mathbb{R}^n \to \mathbb{R}^m$ mit Komponenten $f_1(x), \ldots, f_m(x)$ und $x \in \mathbb{R}^n$.

- Lösen eines nicht-linearen Gleichungssystems $f(x) = 0$
- Minimierung der Quadratsumme $\sum_{i=1}^{m} (f_i(x))^2$

Lösungsmethoden erfordern Abschätzung der Jacobimatrix A von f mit $A = \{\frac{\delta f_i}{\delta x_j}\} \in \mathbb{R}^{m \times n}$,

Motivation

Nicht-lineare Probleme in der Analysis und Optimierung

(1.1) Beispiele

Gegeben sei eine (Vektor-)Funktion f durch $f : \mathbb{R}^n \to \mathbb{R}^m$ mit Komponenten $f_1(x), ..., f_m(x)$ und $x \in \mathbb{R}^n$.

- Lösen eines nicht-linearen Gleichungssystems $f(x) = 0$
- Minimierung der Quadratsumme $\sum_{i=1}^m (f_i(x))^2$

Lösungsmethoden erfordern Abschätzung der Jacobimatrix A von f mit $A = \{\frac{\delta f_i}{\delta x_j}\} \in \mathbb{R}^{m \times n}$,
beispielsweise durch **finite Differenzenquotienten**

$$\frac{\delta f}{\delta x_j}(x) \approx \frac{f(x + \epsilon e_j) - f(x)}{\epsilon}$$

Ein einfaches Beispiel

(1.2) Beispiel

Gegeben sei eine Funktion $f : \mathbb{R}^3 \longrightarrow \mathbb{R}^2$ mit

$$f \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \sin(x_1) + x_2 \\ x_2^2 - x_2 \cdot x_3 \end{pmatrix}$$

Ein einfaches Beispiel

(1.2) Beispiel

Gegeben sei eine Funktion $f : \mathbb{R}^3 \to \mathbb{R}^2$ mit

$$f\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \sin(x_1) + x_2 \\ x_2^2 - x_2 \cdot x_3 \end{pmatrix}$$

Die Jacobimatrix $A \in \mathbb{R}^{2\times 3}$ von f berechnet sich zu

$$A = \begin{pmatrix} \cos(x_1) & 1 & 0 \\ 0 & 2x_2 - x_3 & -x_2 \end{pmatrix}$$

Ein einfaches Beispiel

(1.2) Beispiel

Gegeben sei eine Funktion $f : \mathbb{R}^3 \to \mathbb{R}^2$ mit

$$f \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \sin(x_1) + x_2 \\ x_2^2 - x_2 \cdot x_3 \end{pmatrix}$$

Die Jacobimatrix $A \in \mathbb{R}^{2\times 3}$ von f berechnet sich zu

$$A = \begin{pmatrix} \cos(x_1) & 1 & 0 \\ 0 & 2x_2 - x_3 & -x_2 \end{pmatrix}$$

Finiter Differenzenquotient liefert für Jacobi-Vektor-Produkt

$$A(x)p \approx \frac{f(x+p) - f(x)}{\epsilon} \quad \text{mit} \quad p = \epsilon e_j \ (1 \leq j \leq n)$$

Ein einfaches Beispiel (Fortsetzung)

Für $x = (\pi, 2, 1)^T$ und $p = \epsilon e_1$ $(\epsilon = 0,001)$ erhält man näherungsweise die 1. Spalte der Jacobimatrix A:

Ein einfaches Beispiel (Fortsetzung)

Für $x = (\pi, 2, 1)^T$ und $p = \epsilon e_1$ ($\epsilon = 0,001$) erhält man näherungsweise die 1. Spalte der Jacobimatrix A:

$$A\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix}\right) \cdot \begin{pmatrix} 0,001 \\ 0 \\ 0 \end{pmatrix}$$

$$\approx \frac{1}{0,001} \cdot \left(f\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0,001 \\ 0 \\ 0 \end{pmatrix}\right) - f\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix}\right)\right)$$

Ein einfaches Beispiel (Fortsetzung)

Für $x = (\pi, 2, 1)^T$ und $p = \epsilon e_1$ ($\epsilon = 0,001$) erhält man näherungsweise die 1. Spalte der Jacobimatrix A:

$$A\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix}\right) \cdot \begin{pmatrix} 0,001 \\ 0 \\ 0 \end{pmatrix}$$

$$\approx \frac{1}{0,001} \cdot \left(f\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0,001 \\ 0 \\ 0 \end{pmatrix}\right) - f\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix}\right)\right)$$

Analog: Mit $p = \epsilon e_2$ bzw. $p = \epsilon e_3$ erhält man 2. bzw. 3. Spalte

Ein einfaches Beispiel (Fortsetzung)

Für $x = (\pi, 2, 1)^T$ und $p = \epsilon e_1$ ($\epsilon = 0,001$) erhält man näherungsweise die 1. Spalte der Jacobimatrix A:

$$A\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix}\right) \cdot \begin{pmatrix} 0,001 \\ 0 \\ 0 \end{pmatrix}$$

$$\approx \frac{1}{0,001} \cdot \left(f\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0,001 \\ 0 \\ 0 \end{pmatrix}\right) - f\left(\begin{pmatrix} \pi \\ 2 \\ 1 \end{pmatrix}\right)\right)$$

Analog: Mit $p = \epsilon e_2$ bzw. $p = \epsilon e_3$ erhält man 2. bzw. 3. Spalte

Fazit: 4 Funktionsauswertungen zur Bestimmung der Jacobimatrix

Ein einfaches Beispiel (Fortsetzung)

Beobachtungen:

Ein einfaches Beispiel (Fortsetzung)

Beobachtungen:

- Berechnung der 1. und 3. Spalte gleichzeitig durch $p = \epsilon(e_1 + e_3)$

Ein einfaches Beispiel (Fortsetzung)

Beobachtungen:

- Berechnung der 1. und 3. Spalte gleichzeitig durch $p = \epsilon(e_1 + e_3)$
- In keiner gemeinsamen Zeile haben die 1. und 3. Spalte Einträge ungleich Null:

$$A = \begin{pmatrix} x & x & 0 \\ 0 & x & x \end{pmatrix}$$

Ein einfaches Beispiel (Fortsetzung)

- Die Jacobimatrix

$$A = \begin{pmatrix} x & x & 0 \\ 0 & x & x \end{pmatrix}$$

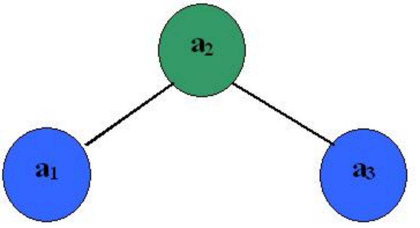
lässt sich als gefärbter Graph darstellen.

Ein einfaches Beispiel (Fortsetzung)

- Die Jacobimatrix

$$A = \begin{pmatrix} x & x & 0 \\ 0 & x & x \end{pmatrix}$$

lässt sich als gefärbter Graph darstellen.



Problemstellung

(1.3) Problem I

Gegeben sei eine Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$.
Wie viele Auswertungen der Funktion f benötigt man im Rahmen des Verfahrens der finiten Differenzenquotienten zur Abschätzung der Jacobimatrix A von f?

Problemstellung

(1.3) Problem I

Gegeben sei eine Funktion $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$.
Wie viele Auswertungen der Funktion f benötigt man im Rahmen des Verfahrens der finiten Differenzenquotienten zur Abschätzung der Jacobimatrix A von f?

(1.4) Beispiel

Ist die Jacobimatrix einer Funktion f tridiagonal, dann benötigt man nur 3 Auswertungen der Funktion f.

Ziel

Problem I lässt sich umformulieren zu

Ziel

Problem I lässt sich umformulieren zu

(1.5) Problem II

Gegeben sei eine Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ mit zugehöriger dünnbesetzter Jacobimatrix $A \in \mathbb{R}^{m \times n}$.
Suche Vektoren $d_1, \ldots, d_p \in \mathbb{R}^{n \times 1}$, so dass die Matrix A direkt und mit kleinstmöglichem p durch die Produkte $Ad_1, \ldots, Ad_p$ eindeutig bestimmt ist.
($\rightarrow$ Curtis, Powell, Reid (1974): CPR-Algorithmus)

Ziel

Problem I lässt sich umformulieren zu

(1.5) Problem II

Gegeben sei eine Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ mit zugehöriger dünnbesetzter Jacobimatrix $A \in \mathbb{R}^{m \times n}$.
Suche Vektoren $d_1, \ldots, d_p \in \mathbb{R}^{n \times 1}$, so dass die Matrix A direkt und mit kleinstmöglichem p durch die Produkte $Ad_1, \ldots, Ad_p$ eindeutig bestimmt ist.
($\to$ Curtis, Powell, Reid (1974): CPR-Algorithmus)

Der CPR-Algorithmus kann als Färbealgorithmus angesehen werden.

Ziel

Problem I lässt sich umformulieren zu

(1.5) Problem II

Gegeben sei eine Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ mit zugehöriger dünnbesetzter Jacobimatrix $A \in \mathbb{R}^{m \times n}$.
Suche Vektoren $d_1, ..., d_p \in \mathbb{R}^{n \times 1}$, so dass die Matrix A direkt und mit kleinstmöglichem p durch die Produkte $Ad_1, ..., Ad_p$ eindeutig bestimmt ist.
(→ Curtis, Powell, Reid (1974): CPR-Algorithmus)

Der CPR-Algorithmus kann als Färbealgorithmus angesehen werden.

(1.6) Ziel

Untersuchung des Zusammenhangs zwischen Färbeproblemen und der Abschätzung dünnbesetzter (Jacobi-)Matrizen.

Übersicht

Definitionen

Definitionen

(2.1) Definition

Eine **Partition** der Spalten von A ist eine Einteilung der Spalten in Gruppen $C_1, \ldots, C_p$, so dass jede Spalte zu genau einer Gruppe C_i gehört, $1 \leq i \leq p$.

Definitionen

(2.1) Definition

Eine **Partition** der Spalten von A ist eine Einteilung der Spalten in Gruppen $C_1, \dots, C_p$, so dass jede Spalte zu genau einer Gruppe C_i gehört, $1 \leq i \leq p$.

(2.2) Definition

Eine Partition heißt **konsistent**, wenn Spalten derselben Gruppe C_i $(1 \leq i \leq p)$ keinen von Null verschiedenen Eintrag in der gleichen Zeile haben.

Partitionen von Spalten

Partitionen von Spalten führen zu einer direkten Bestimmung der Jacobimatrix A:

Partitionen von Spalten

Partitionen von Spalten führen zu einer direkten Bestimmung der Jacobimatrix A:

- Betrachte eine Gruppe C von Spalten a_j der Matrix A $(j = 1, ..., n)$

Partitionen von Spalten

Partitionen von Spalten führen zu einer direkten Bestimmung der Jacobimatrix A:

- Betrachte eine Gruppe C von Spalten a_j der Matrix A $(j = 1, ..., n)$
- Definiere einen Vektor $d \in \mathbb{R}^{n \times 1}$ mit Komponenten δ_j durch

Partitionen von Spalten

Partitionen von Spalten führen zu einer direkten Bestimmung der Jacobimatrix A:

- Betrachte eine Gruppe C von Spalten a_j der Matrix A $(j = 1, ..., n)$
- Definiere einen Vektor $d \in \mathbb{R}^{n \times 1}$ mit Komponenten δ_j durch
  - $\delta_j \neq 0$, falls $j \in C$

Partitionen von Spalten

Partitionen von Spalten führen zu einer direkten Bestimmung der Jacobimatrix A:

- Betrachte eine Gruppe C von Spalten a_j der Matrix A $(j = 1, ..., n)$
- Definiere einen Vektor $d \in \mathbb{R}^{n \times 1}$ mit Komponenten δ_j durch
  - $\delta_j \neq 0$, falls $j \in C$
  - $\delta_j = 0$, sonst

Partitionen von Spalten

Partitionen von Spalten führen zu einer direkten Bestimmung der Jacobimatrix A:

- Betrachte eine Gruppe C von Spalten a_j der Matrix A $(j = 1, ..., n)$
- Definiere einen Vektor $d \in \mathbb{R}^{n \times 1}$ mit Komponenten δ_j durch
  - $\delta_j \neq 0$, falls $j \in C$
  - $\delta_j = 0$, sonst
- Dann gilt:

$$Ad = \sum_{\mathrm{J} \in C} \delta_j a_j$$

Partitionen von Spalten

Partitionen von Spalten führen zu einer direkten Bestimmung der Jacobimatrix A:

- Betrachte eine Gruppe C von Spalten a_j der Matrix A $(j = 1, ..., n)$
- Definiere einen Vektor $d \in \mathbb{R}^{n \times 1}$ mit Komponenten δ_j durch
  - $\delta_j \neq 0$, falls $j \in C$
  - $\delta_j = 0$, sonst
- Dann gilt:

$$Ad = \sum_{j \in C} \delta_j a_j$$

- Enthält C nicht mehr als eine Spalte mit von Null verschiedenen Einträgen in der gleichen Zeile, dann gilt:

$$(Ad)_i = \delta_j a_{ij}$$

Das Partitionsproblem (PP)

(2.3) Problem III

Suche eine konsistente Partition der Spalten von A mit der kleinsten Zahl an Gruppen C_i $(1 \leq i \leq p)$.

Das Partitionsproblem (PP)

(2.3) Problem III

Suche eine konsistente Partition der Spalten von A mit der kleinsten Zahl an Gruppen C_i $(1 \leq i \leq p)$.

(2.4) Definition

$\gamma(A)$ bezeichnet die kleinste Zahl der Gruppen in einer optimalen Partition.

Das Partitionsproblem (PP)

(2.3) Problem III

Suche eine konsistente Partition der Spalten von A mit der kleinsten Zahl an Gruppen C_i $(1 \leq i \leq p)$.

(2.4) Definition

$\gamma(A)$ bezeichnet die kleinste Zahl der Gruppen in einer optimalen Partition.

(2.5) Beispiel, Eisenstat (1980)

$\gamma(A)$ ist nicht notwendig eine untere Schranke, denn:

Das Beispiel von Eisenstat (1980)

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

Das Beispiel von Eisenstat (1980)

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

mit

Das Beispiel von Eisenstat (1980)

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

mit

$$A_1 = \begin{pmatrix} D_1 & D_2 \end{pmatrix} \quad \text{und} \quad A_2 = \begin{pmatrix} z^T & 0 \\ D_3 & B \end{pmatrix}$$

Das Beispiel von Eisenstat (1980)

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

mit

$$A_1 = \begin{pmatrix} D_1 & D_2 \end{pmatrix} \quad \text{und} \quad A_2 = \begin{pmatrix} z^T & 0 \\ D_3 & B \end{pmatrix}$$

Dann gilt: $\gamma(A) = 2n$, $\gamma(A_1) = 2$ und $\gamma(A_2) = n$

Das Beispiel von Eisenstat (1980)

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

mit

$$A_1 = \begin{pmatrix} D_1 & D_2 \end{pmatrix} \quad \text{und} \quad A_2 = \begin{pmatrix} z^T & 0 \\ D_3 & B \end{pmatrix}$$

Dann gilt: $\gamma(A) = 2n$, $\gamma(A_1) = 2$ und $\gamma(A_2) = n$, d.h.

$$\gamma(A) > \gamma(A_1) + \gamma(A_2)$$

Das Beispiel von Eisenstat (1980)

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

mit

$$A_1 = \begin{pmatrix} D_1 & D_2 \end{pmatrix} \quad \text{und} \quad A_2 = \begin{pmatrix} z^T & 0 \\ D_3 & B \end{pmatrix}$$

Dann gilt: $\gamma(A) = 2n$, $\gamma(A_1) = 2$ und $\gamma(A_2) = n$, d.h.

$$\gamma(A) > \gamma(A_1) + \gamma(A_2)$$

Folglich kann A mit $n+2$ Auswertungen von Ad bestimmt werden.

Definitionen, Teil 1

Definitionen, Teil 1

(2.6) Definition

Ein **Graph** G ist ein Tupel (V, E) mit Knotenmenge V und Kantenmenge E.

Definitionen, Teil 1

(2.6) Definition

Ein **Graph** G ist ein Tupel (V, E) mit Knotenmenge V und Kantenmenge E.

(2.7) Definition

Die Knoten $u, v \in V$ heißen **benachbart**, wenn $(u, v) \in E$ eine Kante mit den Endpunkten u und v ist.

Definitionen, Teil 2

(2.8) Definition

Eine **Färbung** eines Graphen G ist eine Funktion $\Phi : V \to \{1, \dots, p\}$, so dass $\Phi(u) \neq \Phi(v)$ für benachbarte Knoten $u, v \in V$ gilt.

Definitionen, Teil 2

(2.8) Definition

Eine **Färbung** eines Graphen G ist eine Funktion
$\Phi : V \to \{1, \ldots, p\}$, so dass $\Phi(u) \neq \Phi(v)$ für benachbarte Knoten $u, v \in V$ gilt.
Wenn G eine Färbung besitzt, dann heißt G **färbbar**.

Definitionen, Teil 2

(2.8) Definition

Eine **Färbung** eines Graphen G ist eine Funktion $\Phi : V \to \{1, ..., p\}$, so dass $\Phi(u) \neq \Phi(v)$ für benachbarte Knoten $u, v \in V$ gilt.
Wenn G eine Färbung besitzt, dann heißt G **färbbar**.
Das kleinste p, für das G färbbar ist, heißt **chromatische Zahl** oder **Färbungszahl** $\chi(G)$ von G.

Definitionen, Teil 2

(2.8) Definition

Eine **Färbung** eines Graphen G ist eine Funktion
$\Phi : V \to \{1, ..., p\}$, so dass $\Phi(u) \neq \Phi(v)$ für benachbarte Knoten $u, v \in V$ gilt.
Wenn G eine Färbung besitzt, dann heißt G **färbbar**.
Das kleinste p, für das G färbbar ist, heißt **chromatische Zahl** oder **Färbungszahl** $\chi(G)$ von G.
Eine Färbung heißt **optimal**, wenn $p = \chi(G)$ gilt.

Färbeprobleme und das Partitionsproblem

Eine Färbung Φ eines Graphen $G = (V, E)$ **induziert** eine Partition der Knoten in die unabhängigen Mengen $C_1, ..., C_p$ mit

$$C_i = \{v \in V : \Phi(v) = i\}, \quad 1 \leq i \leq p$$

Färbeprobleme und das Partitionsproblem

Eine Färbung Φ eines Graphen $G = (V, E)$ **induziert** eine Partition der Knoten in die unabhängigen Mengen $C_1, ..., C_p$ mit

$$C_i = \{v \in V : \Phi(v) = i\}, \quad 1 \leq i \leq p$$

Das Partitionsproblem kann folglich mit dem Färben eines bestimmten Graphen assoziiert werden.

Färbeprobleme und das Partitionsproblem

Eine Färbung Φ eines Graphen $G = (V, E)$ **induziert** eine Partition der Knoten in die unabhängigen Mengen $C_1, \ldots, C_p$ mit

$$C_i = \{v \in V : \Phi(v) = i\}, \quad 1 \leq i \leq p$$

Das Partitionsproblem kann folglich mit dem Färben eines bestimmten Graphen assoziiert werden.

Bestimmung des Graphen: Sei $A \in \mathbb{R}^{m \times n}$ mit Spalten $a_1, \ldots, a_n$. Der Graph $G(A)$ besteht aus der Knotenmenge $V = \{a_1, \ldots, a_n\}$ und den Kanten $(a_i, a_j) \in E \Leftrightarrow a_i \cap a_j \neq \emptyset$.

Ein Färbebeispiel

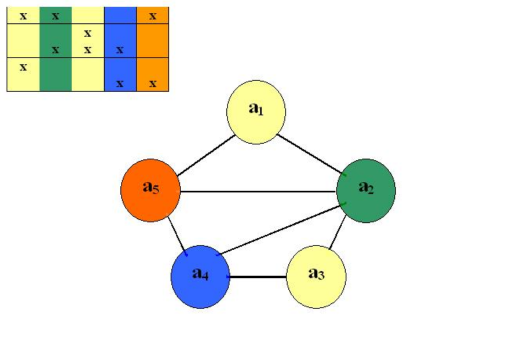
(2.9) Beispiel

Es existiert also eine Kante (a_i, a_j), wenn die i-te und j-te Spalte von A Einträge ungleich Null in wenigstens einer gemeinsamen Zeile haben.

Ein Färbebeispiel

(2.9) Beispiel

Es existiert also eine Kante (a_i, a_j), wenn die i-te und j-te Spalte von A Einträge ungleich Null in wenigstens einer gemeinsamen Zeile haben.



Zentraler Satz

(2.10) Satz

Φ ist eine Färbung von $G(A) \Leftrightarrow \Phi$ induziert eine konsistente Partition der Spalten von A.

Zentraler Satz

(2.10) Satz

Φ ist eine Färbung von $G(A) \Leftrightarrow \Phi$ induziert eine konsistente Partition der Spalten von A.

Daher ist das Partitionsproblem III äquivalent zu

Zentraler Satz

(2.10) Satz

Φ ist eine Färbung von $G(A) \Leftrightarrow \Phi$ induziert eine konsistente Partition der Spalten von A.

Daher ist das Partitionsproblem III äquivalent zu

(2.11) Problem IV (GCP)

Suche eine optimale Färbung für $G(A)$.

Übersicht

Der sequentielle Algorithmus und sein Problem

Gegeben sei ein Graph $G = (V, E)$ mit geordneter Kantenfolge $v_1, ..., v_n \in V$.
Für $k = 1, ..., n$ weist der sequentielle Algorithmus jedem $v_k \in V$ die kleinste Farbe zu.

Der sequentielle Algorithmus und sein Problem

Gegeben sei ein Graph $G = (V, E)$ mit geordneter Kantenfolge $v_1, ..., v_n \in V$.
Für $k = 1, ..., n$ weist der sequentielle Algorithmus jedem $v_k \in V$ die kleinste Farbe zu.

Problem: Zahl der Farben abhängig von der Reihenfolge der Knoten!

Ansätze zur Sortierung der Knoten

Typen von Algorithmen:

Ansätze zur Sortierung der Knoten

Typen von Algorithmen:

- **Largest-First Sortierung**

Ansätze zur Sortierung der Knoten

Typen von Algorithmen:

- **Largest-First Sortierung**
- **Smallest-Last Sortierung**

Ansätze zur Sortierung der Knoten

Typen von Algorithmen:

- **Largest-First Sortierung**
- **Smallest-Last Sortierung**
- **Incidence-Degree Sortierung**

Die Largest-First Sortierung, (WELSH, POWELL)

Die Largest-First Sortierung, (Welsh, Powell)

- Knoten werden ihrem Grad nach absteigend sortiert

Die Largest-First Sortierung, (Welsh, Powell)

- Knoten werden ihrem Grad nach absteigend sortiert
- Sei $d(v)$ der Grad des Knotens $v \in V$, dann erhält man mit einer sequentiellen Färbung

$$\max\{\min(d(v_i) + 1, i) : 1 \leq i \leq n\} \text{ Farben}$$

Die Largest-First Sortierung, (Welsh, Powell)

- Knoten werden ihrem Grad nach absteigend sortiert
- Sei $d(v)$ der Grad des Knotens $v \in V$, dann erhält man mit einer sequentiellen Färbung

$$\max\{\min(d(v_i) + 1, i) : 1 \leq i \leq n\} \text{ Farben}$$

- Nachteil: nicht sehr effizient für bipartite Graphen

Die Largest-First Sortierung, (Welsh, Powell)

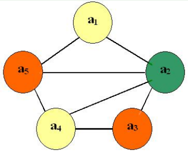
- Knoten werden ihrem Grad nach absteigend sortiert
- Sei $d(v)$ der Grad des Knotens $v \in V$, dann erhält man mit einer sequentiellen Färbung

$$\max\{\min(d(v_i)+1, i) : 1 \leq i \leq n\} \text{ Farben}$$

- Nachteil: nicht sehr effizient für bipartite Graphen

(3.1) Beispiel

Beispiel (2.9): a_2, a_4, a_5, a_1, a_3



Die Smallest-Last Sortierung

beruht auf folgendem Prinzip:

Die Smallest-Last Sortierung

beruht auf folgendem Prinzip:

- Wähle Knoten $v_k \in V$ mit kleinstem Grad

Die Smallest-Last Sortierung

beruht auf folgendem Prinzip:

- Wähle Knoten $v_k \in V$ mit kleinstem Grad
- Wähle anschließend einen benachbarten Knoten $v_{k+1} \in V$, welcher den kleinsten Grad hat

Die Smallest-Last Sortierung

beruht auf folgendem Prinzip:

- Wähle Knoten $v_k \in V$ mit kleinstem Grad
- Wähle anschließend einen benachbarten Knoten $v_{k+1} \in V$, welcher den kleinsten Grad hat
- Suche nun einen Knoten mit kleinstem Grad im induzierten Teilgraphen $V - \{v_{k+1}, ..., v_n\}$

Die Smallest-Last Sortierung

beruht auf folgendem Prinzip:

- Wähle Knoten $v_k \in V$ mit kleinstem Grad
- Wähle anschließend einen benachbarten Knoten $v_{k+1} \in V$, welcher den kleinsten Grad hat
- Suche nun einen Knoten mit kleinstem Grad im induzierten Teilgraphen $V - \{v_{k+1}, ..., v_n\}$
- Setze Sortierung fort

Die Smallest-Last Sortierung

beruht auf folgendem Prinzip:

- Wähle Knoten $v_k \in V$ mit kleinstem Grad
- Wähle anschließend einen benachbarten Knoten $v_{k+1} \in V$, welcher den kleinsten Grad hat
- Suche nun einen Knoten mit kleinstem Grad im induzierten Teilgraphen $V - \{v_{k+1}, ..., v_n\}$
- Setze Sortierung fort

- Man erhält mit einer sequentiellen Färbung

$$\max\{1 + \delta(G_0) : G_0 \text{ ist Teilgraph von } G\} \text{ Farben}$$

Die Smallest-Last Sortierung

beruht auf folgendem Prinzip:

- Wähle Knoten $v_k \in V$ mit kleinstem Grad
- Wähle anschließend einen benachbarten Knoten $v_{k+1} \in V$, welcher den kleinsten Grad hat
- Suche nun einen Knoten mit kleinstem Grad im induzierten Teilgraphen $V - \{v_{k+1}, ..., v_n\}$
- Setze Sortierung fort

- Man erhält mit einer sequentiellen Färbung

$$\max\{1 + \delta(G_0) : G_0 \text{ ist Teilgraph von } G\} \text{ Farben}$$

- Nachteil: nicht sehr effizient für bipartite Graphen

Die Incidence-Degree (ID) Sortierung

(3.2) Definition

Der **Inzidenzgrad** eines Knotens ist gleich der Anzahl der bereits gefärbten Nachbarn.

Die Incidence-Degree (ID) Sortierung

(3.2) Definition

Der **Inzidenzgrad** eines Knotens ist gleich der Anzahl der bereits gefärbten Nachbarn.

Bei dieser Sortierung wird in jedem Schritt der zu färbende Knoten so gewählt, dass er den maximalen Inzidenzgrad hat.

Die Incidence-Degree (ID) Sortierung

(3.2) Definition

Der **Inzidenzgrad** eines Knotens ist gleich der Anzahl der bereits gefärbten Nachbarn.

Bei dieser Sortierung wird in jedem Schritt der zu färbende Knoten so gewählt, dass er den maximalen Inzidenzgrad hat.

(3.3) Satz

Die ID Sortierung ist optimal bei bipartiten Graphen.

Die Incidence-Degree (ID) Sortierung

(3.2) Definition

Der **Inzidenzgrad** eines Knotens ist gleich der Anzahl der bereits gefärbten Nachbarn.

Bei dieser Sortierung wird in jedem Schritt der zu färbende Knoten so gewählt, dass er den maximalen Inzidenzgrad hat.

(3.3) Satz

Die ID Sortierung ist optimal bei bipartiten Graphen.

(3.4) Definition

Der **Sättigungsgrad** eines Knotens ist gleich der Anzahl der unterschiedlich gefärbten Nachbarknoten.

Übersicht

- Rall, L., Corliss, G.: An Introduction to Automatic Differentiation
- Coleman, T., Moré, J.: Estimation of Sparse Jacobian Matrices and Graph Coloring Problems
- Curtis, A., Powell, M., Reid, J.: On the Estimation of Sparse Jacobian Matrices
- Steihaug, T.: Computing a Sparse Jacobian Matrix by Rows and Columns