Lokale Suche

Lokale Suche

Wir betrachten das allgemeine Minimierungsproblem

$$\min_y \quad f(x, y) \quad \text{so dass} \quad L(x, y).$$

Wir nehmen an, dass zu jeder Lösung y auch eine Nachbarschaft $\mathcal{N}(y)$ „benachbarter“ Lösungen gegeben ist.

Die strikte lokale Suche:

(1) Sei $y^{(0)}$ eine Lösung für die Eingabe x. Setze $i = 0$.

(2) Wiederhole solange, bis eine lokal optimale Lösung gefunden ist:

(2a) Bestimme einen Nachbarn $y \in \mathcal{N}(y^{(i)})$, so dass $f(x, y) < f(x, y^{(i)})$ und y eine Lösung ist.

(2b) Setze $y^{(i+1)} = y$ und $i = i + 1$.

Implementierung der lokalen Suche I

- Wie sind die Nachbarschaften $\mathcal{N}(y)$ zu definieren?
  - Wenn nur Elemente aus $\{0,1\}^n$ Lösungen sind, dann ist

$$\mathcal{N}_k(y) = \{y' \in \{0,1\}^n \quad | \quad y \text{ und } y' \text{ unterscheiden sich in in höchstens } k \text{ Positionen}\}$$

    die k**-Flip Nachbarschaft** der Lösung y.
  - Die k-Flip Nachbarschaft besteht aus $\binom{n}{k}$ Elementen und ist für große k zu groß. Deshalb wird häufig $k = 1$ oder $k = 2$ gewählt.
- Mit welcher Anfangslösung $y^{(0)}$ soll begonnen werden?
  - Neben der zufälligen Wahl einer Startlösung
  - werden auch Heuristiken eingesetzt, um mit einer guten Lösung $y^{(0)}$ zu beginnen.
    Achtung: $y^{(0)}$ darf nicht in der Nähe eines lokalen Minimums sein!

Implementierung der lokalen Suche II

- Mit welcher Nachbarlösung soll die Suche fortgesetzt werden?
  - Wenn die Nachbarschaft genügend klein ist, dann liegt die Wahl eines Nachbarn mit kleinstem Zielfunktionswert nahe.
  - Bei zu großen Nachbarschaften wählt man häufig benachbarte Lösungen mit Hilfe einer Heuristik, bzw. man wählt zufällig eine kleine Menge von Nachbarn.

- Die große Schwäche der lokalen Suche:

  Nur Abwärtsbewegungen sind erlaubt.

  Kurz über lang fällt eine Lösung in ein lokales Minimum mit großer Sogwirkung.
- **Auswege:**
  - Im Metropolis Algorithmus und in Simulated Annealing werden auch Aufwärtsbewegungen erlaubt.
  - Die lokale Suche in variabler Tiefe erlaubt ebenfalls Verschlechterungen.

Beispiel: Der Simplex Algorithmus

- Der Lösungsraum des linearen Programmierproblems

$$\min \mathbf{y}^T\mathbf{b}, \text{ so dass } \mathbf{y}^T\mathbf{A} \geqslant \mathbf{c}^T \text{ und } \mathbf{y} \geqslant \mathbf{0}$$

  ist ein Durchschnitt von Halbräumen der Form $\{\mathbf{y} \mid \alpha^T \cdot \mathbf{y} \geqslant \beta\}$.

  Daher liegt ein Optimum in einer „Ecke“ des Lösungsraums.

- Der Simplex-Algorithmus wandert solange von einer Ecke zu einer besseren, benachbarten Ecke, bis keine Verbesserung erreichbar ist.
- Wenn wir also nur Ecken als Lösungen zulassen, dann bildet die lokale Suche die Grundstruktur des sehr erfolgreichen Simplex-Algorithmus.

Beispiel: Minimierung reellwertiger Funktionen

- Die Zielfunktion $f : \mathbb{R}^n \to \mathbb{R}$ des Minimierungsproblems $P = (\min, f, L)$ sei differenzierbar.
- Der Lösungsraum sei eine kompakte Teilmenge des $\mathbb{R}^n$.
- Wenn wir uns gegenwärtig in der Lösung $a \in \mathbb{R}^n$ befinden:
  - In welcher Richtung sollten wir nach besseren Lösungen suchen?
  - In der Richtung des negativen Gradienten!

Die Methode des Gradientenabstiegs

Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine zweimal im Punkt $a \in \mathbb{R}^n$ stetig differenzierbare Funktion mit $\nabla f(a) \neq 0$. Dann gibt es ein $\eta > 0$ mit

$$f\left(a - \eta \cdot \nabla f(a)\right) < f(a),$$

wobei $\nabla f(a)$ der Gradient von f an der Stelle a ist.

- Ersetze a durch $a - \eta \cdot \nabla f(a)$ für ein hinreichend kleines η.

Warum funktioniert der Gradientenabstieg?

- Approximiere die Funktion f durch ihr lineares Taylor-Polynom:

$$f(a+z) = f(a) + \nabla f(a)^T \cdot z + O(z^T \cdot z).$$

  gilt für hinreichend kleines z.

- Für $z = -\eta \cdot \nabla f(a)$ bei hinreichend kleinem $\eta > 0$ ist

$$f(a - \eta \cdot \nabla f(a)) = f(a) - \eta \cdot \ ||\nabla f(a)||^2 + \eta^2 \cdot O(||\nabla f(a)||^2).$$

- Für hinreichend kleines $\eta < 1$ wird somit $f(a - \eta \cdot \nabla f(a))$ kleiner als $f(a)$ sein.

Die Gefahr schlechter lokaler Optima

- Wir betrachten das Vertex Cover Problem VC für einen Graphen $G = (V, E)$.
- Die Nachbarschaft $\mathcal{N}(U)$ einer Teilmenge $U \subseteq V$ besteht aus allen Überdeckungen, die durch Hinzufügen oder Entfernen eines Elementes aus der Überdeckung U entstehen.
- Wir beginnen die lokale Suche mit der Lösung $U = V$ und betrachten einige Beispiele.
  - **Der Graph ohne Kanten:** Die lokale Suche entfernt in jedem Schritt einen Knoten und findet die leere Menge.
  - **Der Sterngraph:**
    - ⋆ Wird im ersten Schritt das Sternzentrum entfernt, dann wird das schlechte lokale Minimum der „Satelliten“ gefunden.
    - ⋆ Ansonsten führt lokale Suche zwangsläufig auf das Sternzentrum.
  - **Der Weg $0 \to 1 \to 2 \to \cdots \to n-2 \to n-1$:** Für gerades n ist $\{0, 2, 4, \ldots, n-2\}$ optimal. Wie sehen die lokalen Minima aus?

Lokale Suche in variabler Tiefe

Lokale Suche in variabler Tiefe

Die Annahme: Der Lösungsraum ist eine Teilmenge von $\{0,1\}^n$.

(1) Setze EINGEFROREN $= \emptyset$, $\mathcal{L} = \{y\}$ und $z = y$.
// EINGEFROREN ist eine Menge von Positionen und
// $\mathcal{L}$ ist eine Menge von Lösungen.

(2) Wiederhole, solange EINGEFROREN $\neq \{1, \ldots, n\}$

(2a) Bestimme eine beste Lösung $z' \neq z$ in der k-Flip Nachbarschaft von z.
// Wechsel an bis zu k Positionen, die nicht in EINGEFROREN sind.

(2b) Friere alle im Wechsel von z nach z' geänderten Positionen ein, füge z' zu $\mathcal{L}$ hinzu und setze $z = z'$.
// Wir erlauben Aufwärtsbewegungen, da z' eine schlechtere
// Lösung als z sein darf. Durch das Einfrieren geänderter
// Positionen wird die Schleife höchstens n Mal durchlaufen.

(3) Gib die beste Lösung in $\mathcal{L}$ als neue Lösung y' aus und wiederhole das Verfahren gegebenenfalls für y'.

Minimum Balanced Cut

- Ein ungerichteter Graph $G = (V, E)$ gegeben.
- **Ziel:** Bestimme eine Zerlegung $V = V_1 \cup V_2$ mit $|V_1| = \lfloor \frac{|V|}{2} \rfloor, |V_2| = \lceil \frac{|V|}{2} \rceil$ so dass die Anzahl kreuzender Kanten, also die Anzahl aller Kanten mit einem Endpunkt in V_1 und einem Endpunkt in V_2,

  minimal ist.

- Eine Anwendung der lokalen Suche in variabler Tiefe wählt die 2-Flip Nachbarschaft.
  - Ein Nachbar einer Zerlegung $V = V_1 \cup V_2$ ist von der Form $V = U_1 \cup U_2$, wobei U_1 durch das Entfernen und das nachfolgende Einfügen eines Elementes aus V_1 entsteht.
- Für Minimum Balanced Cut werden gute experimentelle Ergebnisse berichtet. Gleiches trifft auf das TSP zu.