

Methoden und Anwendungen der Optimierung (MAO)

Kapitel 5: Metaheuristiken – Tabu Search

Univ.-Prof. Dr. Michael Schneider
Christian Schröder

Deutsche Post Chair – Optimization of Distribution Networks (DPO)
RWTH Aachen University

schroeder@dpo.rwth-aachen.de

WS 2017/18



Deutsche Post
Chair - Optimization of
Distribution Networks

RWTHAACHEN
UNIVERSITY

Gesamtgliederung

- 1 Einführung: Heuristiken, Komplexität
- 2 Greedy Algorithmen
- 3 Lösungsqualität und Approximation
- 4 Lokale Suche
- 5 **Metaheuristiken**
 - Einführung
 - ILS
 - VND, VNS
 - **Tabu Search**

Agenda

5 Metaheuristiken – Tabu Search

- Attributives Gedächtnis
- Kurzzeitgedächtnis
- Anspruchskriterien
- Tabu-Search mit Kurzzeitgedächtnis

Tabu Search

Ziele des Kapitels:

- das Prinzip von Tabu Search und mögliche Ausgestaltungen des Gedächtnisses kennen
- die Funktionsweise eines attributiven Gedächtnisses beschreiben können
- sinnvolle Tabu-Bedingungen für gegebene Probleme und Nachbarschaften entwickeln können
- wissen und erklären können, wozu Anspruchskriterien dienen

Grundidee Tabu Search

Merke:

Tabu-Search (TS) ist eine nachbarschaftsbasierte deterministische Metaheuristik. Ihr zentrales Anliegen ist die *Steuerung der Suche zum Verlassen lokaler Optima*.



Informationen über die bisherige Lösungssuche werden in verschiedenen *Gedächtnisspeichern* (kurz: *Gedächtnis*) abgelegt. Eine wesentliche Idee ist die Speicherung von *Attributen* bisher gefundener Lösungen bzw. vollzogener Lösungsübergänge. Vor dem Übergang von einer aktuellen Lösung x zu einer neuen Lösung x' werden die Attribute von x' bzw. $x \rightarrow x'$ mit gespeicherten Attributen im Gedächtnis verglichen. Stimmen die Attribute überein, so wurde die Lösung wahrscheinlich schon besucht und der Übergang zu x' ist *nicht erlaubt (=tabu)*. Andernfalls wird der Übergang erlaubt.

Agenda

5 Metaheuristiken – Tabu Search

■ Attributives Gedächtnis

■ Kurzzeitgedächtnis

■ Anspruchskriterien

■ Tabu-Search mit Kurzzeitgedächtnis

Attributives vs. explizites Gedächtnis I

Attributives Gedächtnis: einzelne Attribute, d.h. Eigenschaften einer Lösung bzw. eines Lösungsübergangs werden gespeichert

Typische Attribute::

- Vorhandensein eines Objekts in einer Lösung (z.B. Standort in einem Standortproblem; Transportverbindung in einem Routing- oder Netzwerk-Design-Problem)
- Anzahl Objekte (z.B. Gegenstände, Behälter, Fahrzeuge, Services) in einer Lösung
- Austausch eines Objekts gegen ein anderes
- Vorteil: relativ geringer Speicherbedarf zum Vermerken von Attributen
- Nachteil: mehrere Lösungen oder Lösungsübergänge können dieselben Attribute aufweisen; in diesem Fall kann eine Lösung tabu gesetzt sein, die bisher nicht untersucht wurden

Attributives vs. explizites Gedächtnis II

Explizites Gedächtnis:

- Speicherung kompletter Lösungen
- wird typischerweise zur Verwaltung bester Lösungen (Elite-Lösungen) verwendet
- beste oder sehr attraktive Lösungen können auch zur Steuerung der Lösungssuche verwendet werden (→path relinking).
- Vorteil: keine Gefahr des Tabu-Setzens von bisher nicht besuchten Lösungen
- Nachteil: relativ hoher Speicheraufwand

Attributives Gedächtnis I

Angenommen, man hat ein *binäres Programmierungsproblem* mit n Entscheidungsvariablen (z.B. ein Rucksackproblem oder ein Set-Covering-Problem) zu lösen und stellt die einzelnen Lösungen durch Angabe der Werte der Entscheidungsvariablen in einem Vektor dar.

Eine Nachbarschaft im Sinne der lokalen Suche kann darin bestehen, die Werte zweier *Entscheidungsvariablen* x_i und x_j

- mit aktuellem Wert $x_i = 0$ und $x_j = 1$
- mittels $x'_i := 1$ und $x'_j := 0$ zu invertieren.

Attributives Gedächtnis II

Die folgenden *Attribute* könnten z.B. zur *Charakterisierung des Übergangs* von einer Lösung zur nächsten herangezogen werden:

Nr.	Basis-Attribut	abgeleitete Tabu-Restriktion
1	$Inc(i)$	Verbiere das Verringern von x_i auf 0
2	$Dec(j)$	Verbiere das Erhöhen von x_j auf 1
3	$Inc(i)$ und $Dec(j)$	Verbiere Schritte, bei denen 1 oder 2 (oder beides) auftritt
4	$Swap(+i, -j)$	Verbiere den umgekehrten Schritt $Swap(+j, -i)$
5	$c(x') - c(x)$	Verbiere Schritte mit Zielfunktionsänderung $c(x) - c(x')$
6	$g(x') - g(x)$	Verbiere Schritte mit Funktionsänderung $g(x) - g(x')$

Die Tabu-Bedingung 3 ist restriktiver als Bedingung 1 und Bedingung 2, während die Tabu-Bedingung 4 jeweils weniger restriktiv ist als die Bedingungen 1, 2 und 3.

Attributives Gedächtnis III

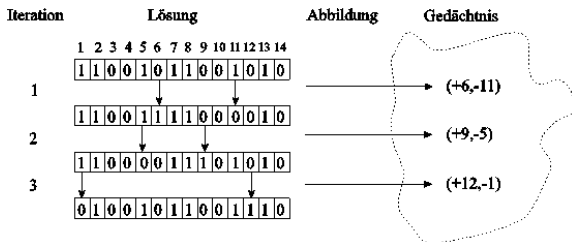
Tabu-Restriktionen können mehr oder weniger *restriktiv* sein.

- restriktive Tabu-Restriktionen haben zur Folge, dass eine Vielzahl von Nachbarlösungen tabu wird
- um Zyklen in der Suche mit Hilfe von Tabu-Restriktionen zu vermeiden, wird man entweder solche Nachbarlösungen verbieten, die das im Gedächtnis vorhandene Attribut aufweisen (Attribute von Lösungen) oder solche Übergänge zu Nachbarlösungen verbieten, die die Inversion eines aktuell abgespeicherten Attributs zur Folge hätten (Attribute von Lösungsübergängen)

Attributives Gedächtnis IV

Das *Attribut Swap*($+i, -j$) soll nun zum Speichern des bisherigen Lösungsverlaufs verwendet werden.

In der folgende Abbildung sind beispielhaft drei Iterationen eines Lokalen Suchverfahrens für einen binären Vektor x und die Entwicklung des Gedächtnisses dargestellt. Dies verdeutlicht den Unterschied zwischen dem Abspeichern der kompletten Lösung und der Nutzung eines attributiven Speichers.



Attributives Gedächtnis V

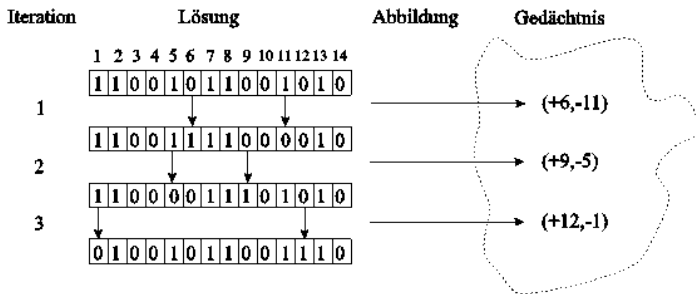
Wie kann nun mit Hilfe des *Gedächtnisses* erreicht werden, dass es bei der Suche nicht zu *Zyklen* kommt?

Dazu definiert man eine Menge von sog. *Tabu-Restriktionen*:

- Eine Tabu-Restriktion bestimmt ausgehend von der aktuellen Lösung x^t , einer Nachbarlösung $x' \in \mathcal{N}^t(x^t)$ und dem aktuellen Status des attributiven Gedächtnisses, ob die Nachbarlösung im aktuellen Iterationsschritt t *tabu* ist.
- Ist die Nachbarlösung tabu, so darf sie im nächsten Schritt nicht gewählt werden. Auf Ausnahmen von dieser Regel wird im folgenden explizit hingewiesen (\rightarrow *Anspruchskriterien*).

Attributives Gedächtnis VI

Der Effekt der Tabu-Restriktionen 1 bis 4 soll nun anhand des Gedächtnisspeichers aus der Abbildung von Folie 9 verdeutlicht werden. Dazu nehmen wir an, dass Nachbarlösungen im vierten Iterationsschritt bezüglich ihres Tabu-Status zu bewerten sind.



Attributives Gedächtnis VII

Die folgende Tabelle zeigt, welche Übergänge zu einer Nachbarlösung tabu sind:

Tabu- Restriktion	verbotene Übergänge
1	Setzen der Variablen x_6 , x_9 und x_{12} auf 0
2	Setzen der Variablen x_{11} , x_5 und x_1 auf 1
3	Setzen der Variablen x_6 , x_9 und x_{12} auf 0 oder der Variablen x_{11} , x_5 und x_1 auf 1
4	Gleichzeitiges Setzen der Variablen $x_6 = 0$ und $x_{11} = 1$ oder $x_9 = 0$ und $x_5 = 1$ oder $x_{12} = 0$ und $x_1 = 1$

Agenda

5 Metaheuristiken – Tabu Search

- Attributives Gedächtnis
- Kurzzeitgedächtnis
- Anspruchskriterien
- Tabu-Search mit Kurzzeitgedächtnis

Kurzzeitgedächtnis I

Die *einfachsten Implementationen von Tabu-Search* verwenden ausschließlich das *Kurzzeitgedächtnis (recency-based memory)*:

- das Kurzzeitgedächtnis speichert die Attribute der zuletzt untersuchten Lösungen oder getätigten Lösungsübergänge in der sog. *Tabu-Liste*
- eine Tabu-Liste der *Länge* k enthält die Attribute der k letzten Lösungen bzw. Lösungsübergänge (man kann sie sich wie eine FIFO-Warteschlange mit begrenzter Länge vorstellen)
- befinden sich beim Hinzufügen eines neuen Attributs bereits k Attribute in der Schlange, so wird das erste Attribut gelöscht; dadurch werden die Attribute derjenigen Lösungen „vergessen“, deren Untersuchung im Rahmen der lokalen Suche weiter zurückliegt

Kurzzeitgedächtnis II

Die *Länge k der Tabu-Liste* legt fest, wie schnell der Prozess des Vergessens ist:

- man muss dabei einerseits abwägen zwischen dem Risiko, dass aufgrund einer zu kurzen Tabu-Liste Zyklen bei der Suche entstehen
- andererseits besteht bei zu langer Tabu-Liste die Möglichkeit, dass die Suche unnötig stark eingeschränkt wird
- bezüglich der optimalen Länge der Tabu-Liste lassen sich keine allgemeinen Aussagen machen
 - in den ersten (frühen) Implementationen wurden sehr oft statische Tabu-Listen der Länge 7 ± 2 verwendet
 - es gibt jedoch erfolgreiche Implementationen mit sehr viel längeren Tabu-Listen

Kurzzeitgedächtnis II

Länge k der Tabu-Liste (Forts.):

- beliebt ist auch der Ansatz, die Länge der Tabu-Liste aufgrund von charakteristischen Daten der Instanz zu bestimmen oder die Länge innerhalb eines Intervalls zufällig zu variieren
- beim reaktiven Tabu-Search (engl.: reactive tabu search, RTS) geschieht die Anpassung aufgrund von Beobachtungen, wie flexibel die Suche in der Vergangenheit war bzw. wie oft sich Lösungen in der näheren Vergangenheit wiederholt haben

Agenda

5 Metaheuristiken – Tabu Search

- Attributives Gedächtnis
- Kurzzeitgedächtnis
- Anspruchskriterien
- Tabu-Search mit Kurzzeitgedächtnis

Anspruchskriterien I

Abhängig von der Länge der Tabu-Liste und der gewählten Tabu-Restriktion kann es vorkommen, dass in einem Iterationsschritt alle Nachbarlösungen tabu sind. Will man nun die Suche mit der gleichen Nachbarschaft fortsetzen, so muss eine Nachbarlösung gewählt werden, die tabu ist: Hierzu dienen sog. *Anspruchskriterien* (engl: aspiration criteria).

Merke: (Anspruchskriterium)

Ein Anspruchskriterium ist eine Bedingung, die den Tabu-Status bestimmter Tabu-Restriktionen aufhebt.



Anspruchskriterien können nicht nur dann aktiviert werden, wenn alle Nachbarlösungen tabu sind, sondern – abhängig von ihrer Ausgestaltung – während des ganzen Verfahrens.

Anspruchskriterien II

Anspruchskriterium	Bedeutung
Anspruch in Ermangelung (aspiration by default)	falls alle Nachbarlösungen tabu sind, wird diejenige gewählt, die nach einem bestimmten Maß „am wenigsten tabu“ ist
Anspruch durch Zielfunktion (aspiration by objective)	falls der Zielfunktionswert besser ist als der bisher beste Wert, ist die Lösung nicht mehr tabu Variante: „besser“ bezieht sich auf die aktuelle Region des Lösungsraums
Anspruch durch Suchrichtung (aspiration by search direction)	falls das zur Restriktion führende Attribut in einer Verbesserungsphase gesetzt wurde und der Schritt zur Nachbarlösung ebenfalls verbessernd ist, wird der Tabu-Status aufgehoben

Agenda

5 Metaheuristiken – Tabu Search

- Attributives Gedächtnis
- Kurzzeitgedächtnis
- Anspruchskriterien
- Tabu-Search mit Kurzzeitgedächtnis

Tabu-Search mit Kurzzeitgedächtnis I

- Tabu-Search benutzt (in der Regel) eine *Bestensuche* (→lokale Suche). Dies wird in der Literatur auch als *aggressive Suche* in der Nachbarschaft bezeichnet.
- Das Verfahren benötigt ein *Stopp-Kriterium*, welches die Suche terminiert.
- Nachfolgend wird eine einfache Variante eines Tabu-Search-Algorithmus vorgestellt, die *ausschließlich das Kurzzeitgedächtnis* verwendet. Dieser Algorithmus kann um zusätzliche Bestandteile erweitert werden (siehe Glover und Laguna, 1997).

Tabu-Search mit Kurzzeitgedächtnis II

Algorithmus 1 : Tabu-Search mit Kurzzeitgedächtnis

```
// Input: Zulässige Lösung  $x^0$ 
SETZE den Iterationszähler  $t := 0$ .
repeat
    // (Bestensuche)
    Durchsuche die gesamte Nachbarschaft  $\mathcal{N}^t(x^t)$  und wähle eine
    bezüglich einer Bewertungsfunktion beste Lösung  $x' \in \mathcal{N}^t(x^t)$ , die
    nicht tabu ist oder tabu ist und ein Anspruchskriterium erfüllt.
    // Iteration
    SETZE  $x^{t+1} := x'$  und  $t := t + 1$ 
    Aktualisiere das Kurzzeitgedächtnis.
until ein Stoppkriterium ist erfüllt
// Output: Lösung  $x^t$  bzw. beste gefunden Lösung  $x^{best}$ 
```

Tabu-Search mit Kurzzeitgedächtnis III

Bemerkungen:

- Die Nachbarschaft kann im Lauf des Verfahrens zur *Intensivierung* und *Diversifikation* variiert werden.
- Da das *Stoppkriterium* nicht sichert, dass das Verfahren mit der besten bisher gefundenen Lösung endet, sollten eine beste oder mehrere „sehr gute“ Lösungen (explizit) gespeichert werden.
- Im einfachsten Fall stimmt die *Bewertungsfunktion* mit der Zielfunktion überein. Sie kann auch Aspekte wie
 - die Zulässigkeit der Lösung (evtl. Strafkosten)
 - den Einfluss der Lösung/des Übergangs auf die Qualität/Strukturberücksichtigen.

Tabu-Search mit Kurzzeitgedächtnis IV

Bemerkungen (Forts.):

- Man kann die *Tabu-Restriktionen* auch als Operatoren auffassen, die die zugrunde liegende Nachbarschaft \mathcal{N} in jeden Iterationsschritt zu einer Nachbarschaft \mathcal{N}^t modifizieren, welche von der aktuellen Iteration t und den im Gedächtnis aufgezeichneten Informationen abhängt.

Zur Vertiefung...



- (Glover und Laguna, 1997)

[Glover und Laguna 1997] Glover, Fred ; Laguna, Manuel: *Tabu search*. Boston : Kluwer Academic Publishers, 1997. – ISBN 9780792399650