ÜBUNG 8

TABU SUCHE

- Ziel: Verlassen lokaler Optima
- Idee: Gedächtnis:
  - Speicherung von Attributen bisher gefundener Lösungen bzw. vollzogener Lösungsübergängen
  - Vor dem Übergang von Lösung x zu x': Attribute von x' bzw $x \to x'$ mit gespeicherten Attributen vergleichen

  → stimmen Attribute überein, so wurde Lös wahrscheinlich schon besucht und der Übergang zu x' ist nicht erlaubt (tabu)

Attributives Gedächtnis	explizites Gedächtnis
• einzelne Eigenschaften einer Lös. bzw. eines Lösungsüberg. werden gespeichert	• Speicherung kompl. Lösungen
• typische Attribute:	• Verwaltung bester Lösung
– Vorhandensein eines Objekts (Standort bei: Standortplanung, Transport v. a. b. in einem Routingproblem)	• Steuerung der Lösungssuche durch gesp. gute Lösungen
– Anzahl von Objekten (Behälter, Fahrzeugen, ...)	
– Austausch eines Objekts ggn ein anderes	
Vorteil: geringer Speicherbedarf	Vorteil: Keine Gefahr des Tabu-Setzens von bisher nicht besuchten Lösungen
Nachteil: mehrere Lös. können selbe Attribute haben. In diesem Fall wird Lös. tabu gesetzt, die bisher nicht untersucht wurde	Nachteil: Speicheraufwand, Rechenaufwand

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

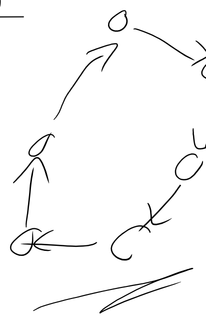
Bsp.: $x = (1,0,1,1,0,1)$

$x' = (1,1,0,1,0,1)$

Attribute:

$Inc(2)$, $Dec(3)$, $Swap(+2,-3)$

$x'' \in N(x')$		$Inc(2)$	$Dec(3)$	$Swap(+2,-3)$
(i)	(1 0 1 1 0 1)	tabu	tabu	tabu
(j,i)	(1 0 0 1 0 1)	tabu	ok	ok
(i,i)	(1 1 1 1 0 1)	ok	tabu	ok



Tabu-Restriktion:

- Bestimmt ausgehend von Lös x^t, Nachbar $x' \in N(x^t)$, gespeicherten Attributen, ob x' tabu ist.

Kurzzeitgedächtnis:

- speichert nur die letzten k Attribute von Lös. bzw. Lösungsüberg.
- Balance finden: k zu klein ⇒ Gefahr Zyklen zu durchlaufen

  k zu groß ⇒ unnötig große Einschränkungen der Suche

Anspruchskriterium:

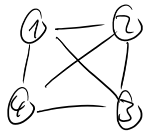
- Bedingung, die den Tabu-Status bestimmter Tabu-Restriktionen aufhebt
- z.B. Akzeptanz von Lösungen, die besser sind als die bisher beste Lösung

Aufgabe 1 (Tabu-Suche):

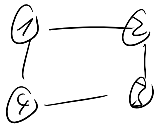
Gegeben sei die in Abb. 1 dargestellte Instanz eines STSP in der euklidischen Ebene. Die Länge einer Kante zwischen zwei Knoten ist durch die euklidische Distanz gegeben. Es soll eine Tabu-Suche mit lokaler Bestensuche unter Verwendung der in Abb. 2 dargestellten Nachbarschaftsoperatoren implementiert werden.

a) Welche Attribute eines Lösungsübergangs halten Sie beim STSP für ein attributives Gedächtnis für geeignet?

Bsp.:

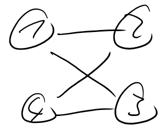


$$x^1 = \boxed{\begin{pmatrix} 0 & 1 & 0 & 1 \\ & 0 & 1 & 0 \\ & & 0 & 1 \\ & & & 0 \end{pmatrix}} \hat{=}$$



EV $x_{ij} = \begin{cases} 1 & \text{falls } j \text{ Nachfolger von } i \\ 0 & \text{sonst} \end{cases}$

$$N(x^1) \ni x^2 = \boxed{\begin{pmatrix} 0 & 1 & 1 & 0 \\ & 0 & 0 & 1 \\ & & 0 & 1 \\ & & & 0 \end{pmatrix}} \hat{=}$$



Analog zum obigen Bsp lassen sich die inc-, dec- und swap Attribute auf die EV x_{ij} anwenden.

inc(1,3), inc(2,4) dec(2,3), dec(1,4)

swap(+(1,3), −(1,4)) swap(+(1,3), −(2,3))

swap(+(2,4), −(1,4)) swap(+(2,4), −(2,3))

b) Was sind beim STSP die Vor- und Nachteile eines expliziten gegenüber eines attributiven Gedächtnisses?

Vorteil:

Keine Gefahr des Tabu-Setzens von bisher nicht besuchten Lös.

Nachteil:

enorme Speicherbedarf, sehr hohe Rechenaufwand

c) Nennen Sie mögliche Tabu-Restriktionen.

Klassisch: gelöschte Kanten werden nicht wieder eingefügt (dec) bzw. eingefügte Kanten nicht gelöscht (inc)

denkbar, aber oft nicht restriktiv genug:

Der gleiche Satz von gelöschten (eingefügten) Kanten darf nicht eingefügt (gelöscht) werden

(ähnlich zu swap)

d) Welche Struktur würden Sie von einer guten Lösung bei der vorliegenden Instanzstruktur erwarten?

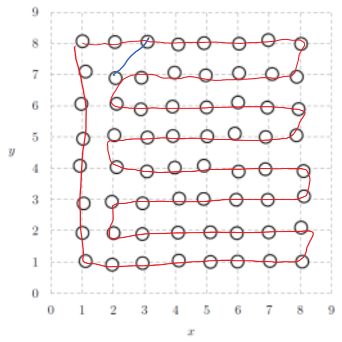
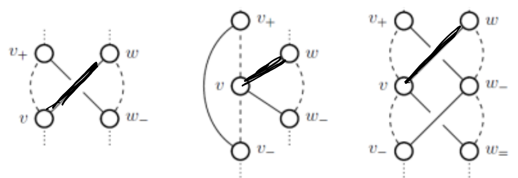


Abbildung 1: STSP Beispielinstanz

Bei Gleichverteilung der Knoten ist davon auszugehen, dass eine gute Lösung nur kurze Kanten des vollst. Graphen enthält. Hier sollten nur horizontale oder vert. Kanten benutzt werden.

e) Wie können sie diese Eigenschaft verwenden, um die Laufzeitkomplexität der lokalen Suche zu verbessern? Welche Laufzeitkomplexität der lokalen Suche könnte unter sinnvollen Annahmen erreicht werden?

Hinweis: Nutzen Sie das Konzept der Generatorkanten (v, w) bei der Definition der Nachbarschaftsoperatoren, um die Nachbarschaft einzuschränken und die Tabu-Suche in vielversprechende Bereiche des Lösungsraums zu lenken.



(a) 2-Opt (b) Relocate (c) Exchange

Abbildung 2: Nachbarschaftsoperatoren für das STSP.

Generatorkante (v,w): Wird durch Op. in die Lösung eingefügt.

Will man die Op. anwenden und gleichzeitig nur horizontale und vert. Kanten in die Lös. einfügen, ist es sinnvoll als Generator-kanten nur die horizontalen und vertikalen Kanten zu nehmen. Damit verringert sich die Komplexität der lok. Suche auf $O(N)$, wobei Beeinträchtigung der Lösungsqualität minimal ausfällt.

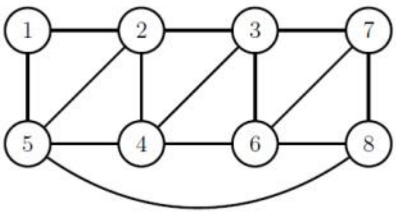
Eine solche Tabu-Suche nennt man auch Granulare Tabu Suche. Oft. liefert GTS sogar bessere Ergebnisse als Tabu Suche mit vollst. Nachbarschaft. Dies liegt daran, dass in gleicher Zeit ein viel größerer Teil des vielversprechenden Suchraums erkundet werden kann.

Aufgabe 2 (Tabu-Suche):

Gegeben seien die in Tab. 1 dargestellten Lösungen x mit Attributen $b_i \in \{0,1\}$, $i = 1,\dots,5$ und Kosten $c(x)$. Der zugehörige Nachbarschaftsgraph ist in Abb. 3 dargestellt. Führen Sie eine Tabu-Suche mit lokaler Bestensuche durch, welche bei $x = 1$ startet. Als Tabu-Kriterium wird das inc-Attribut $(+)$ und das dec-Attribut $(-)$ verwendet: Ändert sich bei einem Lösungsübergang das Attribut b_i, so ist die Umkehrung des Wertes von b_i für die nächsten zwei Iterationen tabu. Ein Übergang, welcher den besten gefundenen Zielfunktionswert verbessert, ist immer erlaubt. Brechen Sie die Suche nach 5 Übergängen ohne Verbesserung ab.

x	b_4	b_3	b_2	b_1	b_0	$c(x)$
1	0	0	0	0	1	50
2	0	0	1	1	1	51
3	0	0	0	1	1	46
4	1	0	1	1	1	47
5	1	1	1	1	0	52
6	0	0	1	1	0	48
7	0	1	0	1	0	49
8	1	1	0	1	0	61

Tabelle 1: Lösungen mit Attributen b_i.



x	$c(x)$	Tabuliste	$N(x)$	x'	x^*
1	50	—	$\{2,5\}$	2	1
2	51	inc(b_2) inc(b_1)	$\{\cancel{1},5,4,3!\}$	3	3
3	46	dec(b_2) ; inc(b_2,b_1)	$\{\cancel{2},\cancel{4},\cancel{6},7\}$	7	3
7	49	inc(b_3) dec(b_0) dec(b_2)	$\{\cancel{3},\cancel{6},8\}$	8	3
8	61	inc(b_3) dec(b_0) inc(b_4)	$\{\cancel{6},\cancel{7},5\}$	5	3
5	52	inc(b_4) inc(b_2)	$\{\cancel{1},\cancel{8},4\}$	4	3
4	47	inc(b_2) dec(b_3) inc(b_0)	$\{2,\cancel{8},\cancel{5},\cancel{6}\}$	2	3

Output $x^* = 3$