Heuristische Optimierungsverfahren

2.Programmierübung

Johannes Reiter - 0625101, Christian Gruber - 0625102

26.Jänner 2009

1 Minimum Energy Broadcast Problem

Das *Minimum Energy Broadcast Problem (MEBP)* ist ein NP-hartes Optimierungsproblem fr das Verbinden von mobilen Geräten in einem drahtlosen Adhoc Netzwerk. Es gilt eine Konfiguration fr das Netzwerk zu finden, bei der die Sendeleitung und die nötige Energie fr die Kommunikation möglichst minimal ist. Die Aufgabenstellung ist nun so, dass eine Nachricht ausgehend von einem vorgegebenen Knoten an alle anderen weitergegeben werden soll. Dabei können Knoten, die die Nachricht schon erhalten haben, auch als Zwischensender dienen, da dies eine energiesparendere Kommunikation ermöglicht. Gesucht sind nun konkrete Pfade, wie die einzelnen Knoten an den Startknoten angebunden werden.
Ein bisschen abstrakter haben wir einen vollständigen gerichteten Graph $G = (V, E, d)$ gegeben. Eine Lösung wird nun durch einen zusammenhängenden gerichteten Spannbaum $T = (V, E_T)$ repräsentiert, in der die Gesamtübertragungsleitung

$$c(P) = \sum_{i \in V} \max_{(i,j) \in E_T} d_{ij}^{l}$$

minimiert wird. In unserem Fall ist $l = 3$. Ein Knoten j bekommt nun die Nachricht von dem Knoten i, falls es eine Kante $(i, j) \in E_T$ gibt.

1.1 Ant Colony Optimization

Wir verwenden den *MAX-MIN Ant System (MMAS)* als ACO Algorithmus. Dabei werden von 10 künstlichen Ameisen Lösungen generiert, die mittels einer lokalen Suche anschlieend noch wenn möglich verbessert werden. Das Pheromon-Modell sieht bei uns so aus, dass die Variable τ_{ij} den Wunsch vom Knoten i zum Knoten j die Nachricht zu senden ausdrückt. Zu Beginn und bei jedem Neustart nach einem bereits konvergierten Zustand werden alle $\tau_e = 0.5$ gesetzt. Beim Konstruieren einer Lösung verwendet jede Ameise eine eingeschränkte Kandidatenliste E_{cand}, von der sie über die Wahrscheinlichkeiten

$$p(e) = \frac{\tau_e \cdot \eta(e)}{\sum_{e' \in E_{cand}} \tau_{e'} \cdot \eta(e')}$$

der einzelnen möglichen Kandidaten einen auswählen. Ermöglicht dieser Schritt weitere Knoten ohne zusätzliche Kosten der Teillösungen hinzuzufügen, wird das anschlieend erledigt. Danach wird der nächste Knoten wieder mittels der Wahrscheinlichkeiten ausgewählt.
Nachdem alle Ameisen ihre Lösungen generiert haben, wird das Pheromon-Update über die Formel

$$\tau_e = \min(\max(\tau_{min}, \tau_e + p \cdot (\xi_e - \tau_e)), \tau_{max})$$

durchgeführt, wobei das $\xi_e = \frac{2}{3}$ falls e in der besten Lösung aus dieser Iteration, $\xi_e = \frac{1}{3}$ falls e in der besten Lösung seit dem letzten Neustart oder $\xi_e = 1$ falls e in beiden besten Lösungen enthalten ist.

1.2 Verbesserungsheuristik

1.3 Vergleich der verschiedenen Varianten für die lokale Suche

Testinstanz	Nachbarschaft	Schrittfunktion	Mittelwert
$4\zeta_{10}^{10}$	move	random	12.9
	move	next	11.2
	move	best	11.2
	2-exchange	random	13.2
	2-exchange	next	11.1
	2-exchange	best	11.6
	split	random	13.1
	split	next	11.6
	split	best	11.6
	rotate subsequence	random	12.8
	rotate subsequence	next	11.3
	rotate subsequence	best	11.2
$15\zeta_{40}^{30}$	move	random	157.6
	move	next	119.6
	move	best	119.5
	2-exchange	random	158
	2-exchange	next	122.7
	2-exchange	best	124
	split	random	158.6
	split	next	134.2
	split	best	133.8
	rotate subsequence	random	158.1
	rotate subsequence	next	118.9
	rotate subsequence	best	118.5

Testinstanz	Nachbarschaft	Schrittfunktion	Mittelwert
$20\zeta_{60}^{40}$	move	random	283.2
	move	next	196.2
	move	best	196.5
	2-exchange	random	283.8
	2-exchange	next	203.8
	2-exchange	best	203.5
	split	random	284.5
	split	next	228.8
	split	best	227.5
	rotate subsequence	random	281.2
	rotate subsequence	next	192.8
	rotate subsequence	best	192.2

1.4 Variable Neighborhood Descent (VND)

Wir haben mit der VND jeweils 30 runs gemacht und als Schrittfunktion ist best verwendet worden. Sehr schön kann man hier erkennen, wie man aus einem lokalen Optimum für eine Nachbarschaft durch einen Wechsel der Nachbarschaftsstruktur wieder heraus finden kann und dadurch das Ergebnis weiter verbessert wird.

Testinstanz	Mittelwert	Bestes Ergebnis	Standardabweichung
$4\zeta_{10}^{10}$	11	10	0.8
$15\zeta_{40}^{30}$	117.2	114	2.3
$20\zeta_{60}^{40}$	193.9	188	4

1.5 Generalized Variable Neighborhood Search (GVNS)

Wir haben mit der GVNS jeweils 30 runs gemacht und als Schrittfunktion haben wir best verwendet. Man kann aus den erzielten Ergebnissen relativ eindeutig die Verbesserung durch die GVNS erkennen. In den log-Dateien sieht man auch wie sehr dieses „shaking“ helfen kann, aus einem lokalen Optimum wieder heraus zu finden.

Testinstanz	Mittelwert	Bestes Ergebnis	Standardabweichung
$4\zeta_{10}^{10}$	10.5	10	0.5
$15\zeta_{40}^{30}$	113.7	111	2
$20\zeta_{60}^{40}$	185.2	181	3.0