Exact Algorithms for the Quadratic Linear Ordering Problem

Martin Gronemann
Bernd Zey

Übersicht

- Wdhlg. Lineare Programmierung
- Wdhlg. Linear Ordering Problem
- Quadratische Programmierung
- Experimentelle Ergebnisse

Lineare Programmierung

- Lineare Zielfunktion $\max \sum_{i=1}^{n} c_i x_i$
- Lineare Nebenbedingungen $a_i^T x \leqslant b_i$
- (Ganzzahligkeitsbedingung $x_i \in \{0, 1\}$)

Linear Ordering Problem (LOP)

$$\max \sum_{i \neq j} c_{ij} x_{ij}$$

$$x_{ij} + x_{ji} = 1$$ 2-Kreis-Ungleichung

$$0 \leq x_{ij} + x_{jk} + x_{ki} \leq 2$$ 3-Kreis-Ungleichung

$$x_{ij} \in \{0, 1\}$$

Linear Ordering Problem (LOP)

$$\max \sum_{i<j} c'_{ij} x_{ij}$$

$$\text{Projektion } x_{ij} = 1 - x_{ji} \qquad c'_{ij} := c_{ij} - c_{ji}$$

$$0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1$$

$$x_{ij} \in \{0, 1\}$$

Quadratische Programmierung (QP)

- Quadratische Zielfunktion

$$\max \; x^T C x \;, \text{C ist } n \times n \text{ Matrix}$$

- Lineare Nebenbedingungen $a_i^T x \leqslant b_i$
- (Ganzzahligkeitsbedinung $x_{ij} \in \{0, 1\}$)

Quadratisches LOP (QLOP)

- Quadratische Zielfunktion

$$\max\ x^T Cx, \quad \text{C ist } \binom{n}{2} \times \binom{n}{2} \text{ Matrix}$$

$$\text{s.t.} \quad x \in \{0,1\}^{\binom{n}{2}} \cap \text{LOP Polytop}$$

QLOP Linearisierung

- Ersetze $x_{ij} \cdot x_{kl}$ durch y_{ijkl}

- max $\sum_{(i,j,k,l) \in I} (c_{ijkl} + c_{klij}) y_{ijkl} + \sum_{i<j} c_{ij} x_{ij}$

$$y_{ijkl} \leq x_{ij}, x_{kl}$$
$$y_{ijkl} \geq x_{ij} + x_{kl} - 1$$
$$y_{ijkl} \in \{0,1\}$$
$$x_{ij} \in \{0,1\}$$

QLOP Ungleichung

- Linearisierte QLOP Ungleichung

$$x_{ik} - y_{ijik} - y_{ikjk} + y_{ijjk} = 0$$

- Ersetzt die 3-Kreis-Ungleichung
- Gleichungssystem ist minimal

Quadratisches LOP

- Quadratisches Programm **ohne LOP Nebenbedingungen** „entspricht“ dem **MaxCut Problem**
- Für MaxCut Polytop gute Ungleichungen bekannt
- Branch & Cut: Erst MaxCut Ungleichungen, dann erst die LOP Ungleichungen

Lösen von QP

- QP ist in Polynomialzeit lösbar, gdw. Matrix *C* positiv-semidefinit (PSD) ist
- Matrix *C* ist PSD, gdw.
  $\forall\, x \in \mathbb{R}^n \setminus 0 \;:\; x^T C\, x \;\in \mathbb{R}^{>0}$
- Andernfalls ist QP NP-schwierig

Experimentelle Ergebnisse

- Algorithmen:
  - **JM**: *2-layer straightline crossing minimization: performance of exact and heuristic algorithms*
  - **CPLEX**: MIP-Solver (default Parameter)
  - **ILP**: Branch & Cut – ILP
  - **SDP**: semidefinite Programmierung
- Instanzen zufällig generiert mit ***random_bigraph*** der *Stanford GraphBase*
- Knotenanzahl n, $n=10,...,18$
- Kantenanzahl $d*n^2 / 100$, $d=10,...,90$

Experimentelle Ergebnisse (2)

n	d	Durchschn.Grad	JM	CPLEX		ILP		SDP	
			time	time	nodes	time	nodes	time	nodes
10	10	1,0	**0,02**	0,70	1,40	0,29	1,00	1,57	1,00
10	20	2,0	**0,07**	2,77	3,30	1,69	1,00	1,72	1,00
10	30	3,0	**0,20**	20,16	92,30	7,89	1,00	3,46	1,00
10	40	4,0	**0,44**	73,51	569,30	23,09	1,00	3,65	1,00
10	50	5,0	**0,80**	207,00	1365,60	43,30	1,00	3,38	1,00
10	60	6,0	**1,50**	831,01	6020,50	95,21	1,00	3,76	1,00
10	70	7,0	**3,06**	1996,00	11440,20	195,93	1,00	4,72	1,00
10	80	8,0	5,29	1060,00	4159,90	176,03	1,00	**3,61**	1,00
10	90	9,0	11,59	202,00	418,40	329,85	1,50	**4,84**	1,00
12	10	1,2	**0,26**	7,22	4,90	8,64	1,00	12,81	1,00
12	20	2,4	**1,94**	33,57	55,10	31,37	1,00	13,80	1,00
12	30	3,6	**5,92**	242,54	847,60	72,24	1,00	15,16	1,00
12	40	4,8	21,27	2826,62	9288,00	248,65	1,00	**17,49**	1,00
12	50	6,0	74,36			1175,62	1,00	**31,20**	1,00
12	60	7,2	134,40					**32,17**	1,00
12	70	8,4	281,69					**26,89**	1,00
12	80	9,6	682,36					**19,95**	1,00
12	90	10,8	1373,39					**15,75**	1,00
14	10	1,4	19,87	**14,47**	3,90	28,12	1,00	51,82	1,00
14	20	2,8	142,11	347,06	510,40	263,99	1,00	**61,85**	1,00
14	30	4,2	963,12			1339,48	1,00	**93,19**	1,00
14	40	5,6	2299,98					**97,93**	1,00
14	50	7,0						**117,55**	1,00
14	60	8,4						**95,86**	1,00
14	70	9,8						**101,83**	1,00
14	80	11,2						**101,85**	1,00
14	90	12,6						**56,30**	1,00

Experimentelle Ergebnisse (2)

n	d	Durchschn.Grad	JM	CPLEX		ILP		SDP	
			time	time	nodes	time	nodes	time	nodes
16	10	1,6		**101,30**	12,80	415,55	1,00	119,14	1,00
16	20	3,2				1953,26	1,00	**200,87**	1,00
16	30	4,8						**432,85**	1,20
16	40	6,4						**1432,02**	2,80
16	50	8,0						**1181,20**	2,40
16	60	9,6						**1186,81**	2,20
16	70	11,2						**916,86**	1,80
16	80	12,8						**444,92**	1,20
16	90	14,4						**224,14**	1,00
18	10	1,8		**314,60**	44,80	1145,60	1,00	343,23	1,00
18	20	3,6						**491,87**	1,00
18	30	5,4						**1233,73**	1,00
18	40	7,2							
18	50	9,0							
18	60	10,8							
18	70	12,6						**2694,98**	2,00
18	80	14,4						**2523,04**	2,00
18	90	16,2						**601,30**	1,00

Fazit

- Bei Branch & Cut werden nur wenige Branch-Knoten erzeugt (gute Qualität der MaxCut Constraints)
- Semidefinite Programmierung ist bei dichten Graphen wesentlich schneller