

Einführung Optimierungssoftware

Walter S. Gisler

22. Oktober 2021

walter@gotpro.com

Ein minimales Stundenplanungsproblem

- Wir betrachten nur 2 Klassen
- Wir haben eine gegebene Liste von Lektionen pro Klasse, welche eingeplant werden müssen
- Die beiden Klassen haben eine gemeinsame Lektion
- Wir ignorieren die Raumzuteilung
- Wir müssen sicherstellen, dass ein Lehrer nicht mehr als eine Lektion gleichzeitig geben muss
- Freier Mittwochnachmittag

Ein minimales Stundenplanungsproblem

U21a Klassenlehrperson: **bip** Prorektorat: **sup**

	Mo	Di	Mi	Do	Fr
8:00-8:45	bip T.5 SH 1)	gej S0.3 TG 2)	scn P.20 DE	frc R3.7 IN	bip R2.8 GG
8:50-9:35				scn P.12 DE	
9:55-10:40	arj V1.27 NT	ebm S2.4 FR	bip P.10 KS	dua R3.3 GS	ale B1.7 MA
10:45-11:30			bip P.10 BL		
11:40-12:25			ale B1.7 MA	bip T.3 SH 3)	dea R1.5 EN
12:30-13:15					
13:30-14:15	blb B3.7 RE	dea R1.5 EN		ale B1.7 MA	
14:20-15:05					ebm S2.4 FR
15:20-16:05	scn P.20 DE	vom S0.9 MU			gej Z0.12 BG
16:10-16:55					
17:00-17:45					

U21b Klassenlehrperson: **frm** Prorektorat: **sup**

	Mo	Di	Mi	Do	Fr
8:00-8:45	bip T.5 SH 1)	frc R3.7 IN	gyg Z0.12 BG	jeg B0.9 FR	buk B2.9 MA
8:50-9:35		frm P.10 DE			
9:55-10:40	buk B2.9 MA	frm P.10 KS	buk B2.9 MA	hlu M0.1 MU	arj V1.27 NT
10:45-11:30	gyg S0.3 TG 2)				
11:40-12:25			frm P.10 DE	bip T.3 SH 3)	jeg S2.3 FR
12:30-13:15					
13:30-14:15	dua R3.3 GS	fsl P.23 EN		blb B3.7 RE	frm P.10 DE
14:20-15:05					
15:20-16:05	frm P.10 BL	buk R2.8 GG			fsl P.23 EN
16:10-16:55					
17:00-17:45					

	A	B	C	D	E	F
1	ID	Fach	Lehrperson	Stunden	Klasse	Overlap
2	1	SH	bip	2	U21a	23
3	2	NT	arj	2	U21a	0
4	3	RE	blb	2	U21a	0
5	4	DE	scn	2	U21a	0
6	5	TG	gej	2	U21a	0
7	6	FR	ebm	2	U21a	0
8	7	EN	dea	2	U21a	0
9	8	MU	vom	2	U21a	0
10	9	DE	scn	2	U21a	0
11	10	KS	bip	1	U21a	0
12	11	BL	bip	1	U21a	0
13	12	MA	ale	1	U21a	0
14	13	IN	frc	1	U21a	0
15	14	DE	scn	1	U21a	0
16	15	GS	dua	2	U21a	0
17	16	SH	bip	1	U21a	38
18	17	MA	ale	2	U21a	0
19	18	GG	bip	2	U21a	0
20	19	MA	ale	2	U21a	0
21	20	EN	dea	1	U21a	0
22	21	FR	ebm	1	U21a	0
23	22	BG	gej	2	U21a	0
24	23	SH	bip	2	U21b	1
25	24	MA	buk	1	U21b	0
26	25	TG	gyg	2	U21b	0
27	26	GS	dua	2	U21b	0

Was macht einen guten Stundenplan aus?

- Keine Lücken zwischen Lektionen
- Falls ein Fach in einer Woche mehrfach unterrichtet wird, soll zwischen den Lektionen ein gewisser Abstand liegen
- Nicht zu viele Stunden an einem Tag
- Abwechslung: z.B. nicht zwei Sprachfächer am selben Morgen/Nachmittag, schwierige Fächer und leichtere Fächer (z.B. Sport) abwechseln

Zwei Programmierparadigmen

- Gemischt ganzzahlige Programmierung (MIP)
- Constraintprogramming (CP)

Allgemeine Vorgehensweise

- Planungsentscheide mit Entscheidungsvariablen abbilden:
 - Hier: zu welcher Zeit findet welche Lektion statt?
- Constraints mit den Entscheidungsvariablen formulieren
- Zielfunktion definieren: je grösser (alternativ: kleiner) der Zielfunktionswert, desto besser die Lösung
- Typischerweise stellen Sie das Modell zuerst mathematisch auf und implementieren es dann in einem späteren Schritt

Entscheidungsvariablen: MIP

		Mo	Di	Mi	Do	Fr
		0	1	2	3	4
08:00 - 08:45	0					
08:50 - 09:35	1					
09:55 - 10:40	2					
10:45 - 11:30	3					
11:40 - 12:25	4					
12:30 - 13:15	5					
13:30 - 14:15	6					
14:20 - 15:05	7					
15:20 - 16:05	8					
16:10 - 16:55	9					
17:00 - 17:45	10					

Ja/Nein-Entscheidung: Lektion a findet am Tag b im Zeitslot c statt/ nicht statt

	A	B	C	D	E	F
1	ID	Fach	Lehrperson	Stunden	Klasse	Overlap
2	1	SH	bip	2	U21a	23
3	2	NT	arj	2	U21a	0
4	3	RE	blb	2	U21a	0
5	4	DE	scn	2	U21a	0
6	5	TG	gej	2	U21a	0
7	6	FR	ebm	2	U21a	0
8	7	EN	dea	2	U21a	0
9	8	MU	vom	2	U21a	0
10	9	DE	scn	2	U21a	0
11	10	KS	bip	1	U21a	0
12	11	BL	bip	1	U21a	0
13	12	MA	ale	1	U21a	0
14	13	IN	frc	1	U21a	0
15	14	DE	scn	1	U21a	0
16	15	GS	dua	2	U21a	0
17	16	SH	bip	1	U21a	38
18	17	MA	ale	2	U21a	0
19	18	GG	bip	2	U21a	0
20	19	MA	ale	2	U21a	0
21	20	EN	dea	1	U21a	0
22	21	FR	ebm	1	U21a	0
23	22	BG	gej	2	U21a	0
24	23	SH	bip	2	U21b	1
25	24	MA	buk	1	U21b	0
26	25	TG	gyg	2	U21b	0
27	26	GS	dua	2	U21b	0

Entscheidungsvariablen: MIP

		Mo	Di	Mi	Do	Fr
		0	1	2	3	4
08:00 - 08:45	0					
08:50 - 09:35	1					
09:55 - 10:40	2					
10:45 - 11:30	3					
11:40 - 12:25	4					
12:30 - 13:15	5					
13:30 - 14:15	6					
14:20 - 15:05	7					
15:20 - 16:05	8					
16:10 - 16:55	9					
17:00 - 17:45	10					

MU/vom

$$X[8,2,5] = 1$$

	A	B	C	D	E	F
1	ID	Fach	Lehrperson	Stunden	Klasse	Overlap
2	1	SH	bip	2	U21a	23
3	2	NT	arj	2	U21a	0
4	3	RE	blb	2	U21a	0
5	4	DE	scn	2	U21a	0
6	5	TG	gej	2	U21a	0
7	6	FR	ebm	2	U21a	0
8	7	EN	dea	2	U21a	0
9	8	MU	vom	2	U21a	0
10	9	DE	scn	2	U21a	0
11	10	KS	bip	1	U21a	0
12	11	BL	bip	1	U21a	0
13	12	MA	ale	1	U21a	0
14	13	IN	frc	1	U21a	0
15	14	DE	scn	1	U21a	0
16	15	GS	dua	2	U21a	0
17	16	SH	bip	1	U21a	38
18	17	MA	ale	2	U21a	0
19	18	GG	bip	2	U21a	0
20	19	MA	ale	2	U21a	0
21	20	EN	dea	1	U21a	0
22	21	FR	ebm	1	U21a	0
23	22	BG	gej	2	U21a	0
24	23	SH	bip	2	U21b	1
25	24	MA	buk	1	U21b	0
26	25	TG	gyg	2	U21b	0
27	26	GS	dua	2	U21b	0

Implementation

- Demo: siehe Jupyter Notebook

Bemerkungen:

- Es gibt immer verschiedene richtige Modellierungsansätze
- Das Beispiel ist hauptsächlich darauf ausgelegt, möglichst einfach verständlich zu sein. In der Realität sollten Sie sich Gedanken machen, wie sie das ganze objektorientiert gestalten können und wie sie verschiedene Komponenten abgrenzen und wiederverwendbar machen: Constraints, Input/Output, Variablen etc.

Ressourcen

- Optimierer:
 - Cplex: <https://www.ibm.com/academic/home>
 - Fico Xpress: <https://community.fico.com/s/academic-programs>
 - OR Tools: <https://developers.google.com/optimization>
- Modellierungs-Libraries
 - Pyomo: https://pyomo.readthedocs.io/en/stable/pyomo_overview/simple_examples.html
 - PuLP: <https://github.com/coin-or/pulp>
 - JuMP: <https://jump.dev>

Ressourcen

- Beispiele (Xpress):
 - <https://github.com/RichardOberdieck/practical-optimization-course/tree/master/Exercises/Jupyter%20notebooks>
- Linking Object Oriented Programming with Optimization:
 - <https://github.com/RichardOberdieck/optimization-blog/blob/master/Linking%20object-oriented%20programming%20and%20optimization.ipynb>
- Testing:
 - <https://github.com/RichardOberdieck/optimization-blog/blob/master/Testing%20optimization%20problems.ipynb>