

36. BWInf Aufgabe 2

Schwimmbad

Dokumentation

Kamal Abdellatif

27. November 2017

Das Problem lässt sich als klassisches ILP-Problem¹ darstellen. Das Ziel ist es den Gesamtpreis in Abhängigkeit der ganzzahligen Parameter, der Wahl von Tickets, zu optimieren. Die Parameter sind durch Bedingungen eingeschränkt, welche aus dem Wortlaut der in der Aufgabenstellung formulierten Regeln hervorgehen. ILP-Probleme sind im Allgemeinen NP-vollständig². Es wird nach Brute-Force vorgegangen. Das Programm probiert systematisch alle möglichen Angebotsauswahlen mit den entsprechenden Gesamtpreisen aus, und wählt die des kleinsten Preises.

Folgende Größen sind die Argumente von der Zielfunktion des Gesamtpreises P_{ges}

$e \in \mathbb{N}$: Anzahl von Erwachsenen	$j \in \mathbb{N}$: Anzahl von Jugendlichen
$k \in \mathbb{N}$: Anzahl von Kleinkindern	$g \in \mathbb{N}$: Anzahl von Gutscheinen

Kleinkinder sind definiert als Personen mit Alter unter 4 Jahren, Jugendliche zwischen einschließlich 4 und einschließlich 16 Jahren, und Erwachsene über ausschließlich 16 Jahre. Da Kinder sich nicht auf die Preise an sich auswirken, werden diese zunächst vernachlässigt.

Person	Alter (Intervall)
Kleinkind	$[0; 4[$
Jugendlicher	$[4; 16]$
Erwachsener ³	$]16; \infty[$

¹engl. *Integer Linear Programming*

²<https://people.eecs.berkeley.edu/~luca/cs172/karp.pdf>

³Unsterbliche werden nicht als erwachsen gezählt.

Der zeitliche Rahmen ist gegeben durch zwei Wahrheitswerte W, F .

$W \in \{0, 1\} : 1$, wenn es sich um einen Arbeitstag handelt, sonst 0

$F \in \{0, 1\} : 1$, wenn Ferien sind, sonst 0

Die ersten Parameter sind die Anzahlen von Personen, die von einem bestimmten Angebot betroffen sind und die Anzahl von Familien

d : Anz. P. die durch einen Gutschein ermäßigt sind

h : Anz. P. die durch einen Gutschein kostenlos eintreten

t : Anz. P. die unter einer Tageskarte eintreten

f_{22} : Anz. von Familien mit 2 Erwachsenen und 2 Jugendlichen

f_{13} : Anz. von Familien mit 1 Erwachsenen und 3 Jugendlichen

$$(d, h, t, f_{22}, f_{13} \in \mathbb{N})$$

Die ersten drei Parameter sind jeweils durch die Gesamtanzahl von Personen $e + j$ nach oben begrenzt. h kann noch weiter eingegrenzt werden, sobald die Anzahl der verfügbaren Gutscheine g kleiner als die Anzahl von Personen wird. Wenn keine Gutscheine vorhanden sind, so kann weiterhin keine Person durch einen solchen ermäßigt werden. Da Tageskarten nur wochentags erlaubt sind, ist die maximale Anzahl am Wochenende ($\neg W$) null. Die Bedingung der Ferien ist gleichzusetzen mit dem Ausdruck $F \Rightarrow g = 0$, da Gutscheine nicht in den Ferien gelten, und so als „nicht vorhanden“ angesehen werden können. Dies spiegelt sich in den logischen Ausdrücken wider. Die Anzahl von möglichen Familien geht aus der Verteilung von Kindern und Erwachsenen hervor.

$$\begin{aligned} 0 \leq d &\leq \begin{cases} e + j & (g > 0 \wedge \neg F) \\ 0 & (g = 0 \vee F) \end{cases} \\ 0 \leq h &\leq \begin{cases} \min(g, e + j) & (\neg F) \\ 0 & (F) \end{cases} \\ 0 \leq t &\leq \begin{cases} e + j & (W) \\ 0 & (\neg W) \end{cases} \\ 0 \leq f_{22} &\leq \min\left(\frac{1}{2}e, \frac{1}{2}j\right) \\ 0 \leq f_{13} &\leq \min\left(e, \frac{1}{3}j\right) \end{aligned}$$

Die Intervalle stellen die jeweilige Menge an möglichen Werten an, die ein Parameter annehmen kann.

$$d \in D \quad h \in H \quad t \in T \quad f_{22} \in F_{22} \quad f_{13} \in F_{13}$$

Jede mögliche Einstellung von allen Parametern, formal jedes mögliche Tupel, wird als das kartesische Produkt der einzelnen Möglichkeitsmengen erzeugt.

$$(d, h, t, f_{22}, f_{13}) \in D \times H \times T \times F_{22} \times F_{13}$$

Zu jedem Tupel $(d, h, t, f_{22}, f_{13})$ von Personenanzahlen des jeweiligen Angebots gibt es mögliche Verteilungen. Eine Verteilung besteht aus den einzelnen Anzahlen von Jugendlichen und Erwachsenen des entsprechenden Angebots. Die Anzahl von Jugendlichen x_j eines Angebots x muss dabei mit der Anzahl von Erwachsenen x_e in der Summe die Gesamtanzahl von Personen des Angebots ergeben.

$$\begin{array}{lll} d = d_e + d_j & h = h_e + h_j & t = t_e + t_j \\ 0 \leq d_e, d_j \leq d & 0 \leq h_e, h_j \leq h & 0 \leq t_e, t_j \leq t \end{array}$$

$$(d_e, d_j, h_e, h_j, t_e, t_j \in \mathbb{N})$$

Da die Summe $x_e + x_j$ konstant sein muss, ergibt sich so für jedes x_e genau ein x_j . So kann die Menge der möglichen Verteilungen wie bei den Gesamtanzahlen über das kartesische Produkt der Möglichkeiten erzeugt werden. Die jeweils korrespondierende Größe kann durch die Differenz mit der gegebenen Summe eindeutig ergänzt werden.

$$(d_e, h_e, t_e) \in [0; d] \times [0; h] \times [0; t]$$

$$d_j = d - d_e \quad h_j = h - h_e \quad t_j = t - t_e$$

Die einzig fehlenden Größen sind die Anzahl von Jugendlichen g_j und Erwachsenen g_e , die ohne besonderes Angebot den normalen Basispreis entsprechend des zeitlichen Rahmens zahlen. Da die Gesamtanzahl von Jugendlichen j und Erwachsenen e gegeben sind, sind diese Werte eindeutig festgelegt, da sie in der Summe mit den jeweiligen Anzahlen der einzelnen Angebotsteilnehmer die vorgegebene Gesamtsumme ergeben müssen. Erfordert dies jedoch, dass g_e oder g_j negative Werte annehmen müssen, bedeutet das, dass die Einstellung ungültig ist, da die Angebotsbelegung die Anzahl von Personen überschreitet. Diese Fälle werden ignoriert.

$$\begin{array}{l} e = g_e + d_e + h_e + t_e + 1 \cdot f_{13} + 2 \cdot f_{22} \\ j = g_j + d_j + h_j + t_j + 3 \cdot f_{13} + 2 \cdot f_{22} \end{array}$$

So sind alle benötigten Anzahlen für die Preisberechnung vorhanden. Für jedes Angebot gibt es einen vorgeschriebenen Preis, welcher zum Teil vom zeitlichen Rahmen abhängt. Der Gesamtpreis errechnet sich aus der Summe aller Produkte von Angebotspreisen mit

der jeweiligen Anzahl. Da der Preis minimiert werden soll, kann für die eigentliche Anzahl von verwendeten Tageskarten K die Anzahl von Karten eingesetzt werden, die mindestens gebraucht wird, um alle Personen mit Tageskarten-Ermäßigung abzudecken.

$$K = \left\lceil \frac{t}{6} \right\rceil$$

$$\text{Grundpreis Erwachsener : } P_{ge} = 3,50e \cdot \begin{cases} 1 - 20\% & (W) \\ 1 & (\neg W) \end{cases}$$

$$\text{Grundpreis Jugendlicher : } P_{gj} = 2,50e \cdot \begin{cases} 1 - 20\% & (W) \\ 1 & (\neg W) \end{cases}$$

$$\text{Tageskarte : } P_t = 11,00e$$

$$\text{Familie : } P_f = 8,00e$$

$$\text{Erwachsener mit Gutscheinerem. : } P_{de} = P_{ge} \cdot (1 - 10\%)$$

$$\text{Jugendlicher mit Gutscheinerem. : } P_{dj} = P_{gj} \cdot (1 - 10\%)$$

Preisfunktion

$$P_{ges} = P_{ge} \cdot g_e + P_{gj} \cdot g_j + P_t \cdot K + P_f \cdot (f_{22} + f_{13}) + P_{de} \cdot d_e + P_{dj} \cdot d_j$$

Programm

Das Programm geht in der selben Reihenfolge wie oben beschrieben vor. Für jede mögliche Kombination von Personenanzahlen der einzelnen Angebote wird jede mögliche Verteilung dieser Personen auf die Angebote getestet. Für jeden Testfall wird der Gesamtpreis ermittelt. Falls der bisherige Minimal-Gesamtpreis unterboten wurde, so wird dieser durch den Preis ersetzt. So erhält nach vollständigem Durchtesten den Minimalpreis mit der Minimalbelegung.

Kinder wurden bisher vernachlässigt. Ein Kind kann kostenlos eintreten, jedoch nur in Begleitung eines Erwachsenen. Daher kann jedes Kind einer beliebigen erwachsenen Begleitperson zugeordnet werden, insofern diese existiert. Dies hat keine Auswirkungen auf den Preis. Eine Belegung kann nur valide sein, wenn bei Vorhandensein von Kindern mindestens ein Erwachsener in der Gruppe enthalten ist. Es muss folglich die Bedingung $k > 0 \Rightarrow e > 0$ gelten, sonst ist keine gültige Belegung möglich.

Für die in der Aufgabe formulierte Menge an Angeboten können bei genauerer Betrachtung von Spezialfällen Vereinfachungen gemacht werden, welche den Umfang der nötigen Berechnungen stark reduzieren würden. Dies ist jedoch immer abhängig von

den in der Aufgabe gegebenen Bedingungen, und kann in Komplexität und überhaupt in Umsetzbarkeit stark variieren. Daher habe ich mich für eine verallgemeinerte Lösung entschieden, welche in Anbetracht des zu erwartenden Umfangs der Eingabedaten ohne Ausnutzung spezieller Eigenschaften der vorgegebenen Bedingungen in akzeptabler (polynomieller) Laufzeit immer ein zuverlässiges Ergebnis liefert.

Testfälle

Das Eingabeformat erfolgt als Textdatei. In der ersten und zweiten Zeile stehen die Wahrheitswerte **True** oder **False** für Woche (*W*) und Ferien (*F*). Die dritte Zeile enthält die Anzahl verfügbarer Gutscheine. Die darauf folgenden Zeilen sind die Auflistung des Alters aller Personen. Das Programm wird über die Kommandozeile aufgerufen. Die Benutzung wird hier am gegebenen Beispiel in der Aufgabe gezeigt.

```
$ cat eingaben/beispiel0.txt
False # Wochenende
True  # in den Ferien
1    # Coupons
13   # Antonia und 4 Freunde
13
13
13
2    # Schwester
45   # Mutter
$ ./schwimmbad.py eingaben/beispiel.txt
```

	Angebot	Erwachsene	Jugendliche
	-----+-----+-----		
	Basispreis	0	1
Erm. durch Gutschein		0	0
Freie Eintritte		0	0
Erm. durch Tageskarte		0	0

	Anzahl
-----+-----	
Tageskarten	0
2-2-Familien	0
1-3-Familien	1

Gesamtpreis: 10.50€

\$

Die verwendeten Eingabedateien der folgenden Tests sind unter dem Ordner **eingaben/** zu finden. Es steht zuerst der Dateiname, gefolgt von Dateiinhalt und Programmausgabe mit optionalem Fazit.

```

beispiel0.txt
False # Wochenende
True  # in den Ferien
1    # Coupons
13   # Antonia und 4 Freunde
13
13
13
2    # Schwester
45   # Mutter

```

Angebot	Erwachsene	Jugendliche
Basispreis	0	1
Erm. durch Gutschein	0	0
Freie Eintritte	0	0
Erm. durch Tageskarte	0	0

	Anzahl
Tageskarten	0
2-2-Familien	0
1-3-Familien	1

Gesamtpreis: 10.50€

[illegible]

Angebot	Erwachsene	Jugendliche
Basispreis	3	0
Erm. durch Gutschein	0	0
Freie Eintritte	0	0
Erm. durch Tageskarte	24	0
	Anzahl	
Tageskarten	4	
2-2-Familien	0	
1-3-Familien	0	

Gesamtpreis: 52.40€

[illegible]

Angebot	Erwachsene	Jugendliche
Basispreis	0	0
Erm. durch Gutschein	25	0
Freie Eintritte	2	0
Erm. durch Tageskarte	0	0
	Anzahl	
Tageskarten	0	
2-2-Familien	0	
1-3-Familien	0	

Gesamtpreis: 78.75€

Fazit:

```
# Wenn die Klasse versucht, am Wochenende in der Schulzeit zu gehen, ist es
# sogar erheblich teurer.
```



```

beispiel2a.txt
True # Wochentag
True
0
3
7
10
14
17
15
40
40

```

Angebot	Erwachsene	Jugendliche
Basispreis	0	1
Erm. durch Gutschein	0	0
Freie Eintritte	0	0
Erm. durch Tageskarte	3	3

	Anzahl
Tageskarten	1
2-2-Familien	0
1-3-Familien	0

Gesamtpreis: 13.00€

```

beispiel2b.txt
False # Wochenende
True
0
3
7
10
14
17
15
40
40

```

Angebot	Erwachsene	Jugendliche
Basispreis	1	2
Erm. durch Gutschein	0	0
Freie Eintritte	0	0
Erm. durch Tageskarte	0	0

	Anzahl
Tageskarten	0
2-2-Familien	1
1-3-Familien	0

Gesamtpreis: 16.50€

beispiel3a.txt

False

False

1 # Es wird angenommen dass jeder Verein einen Gutschein hat

18 # 14 Erw.

18

18

18

18

18

18

18

18

18

18

18

18

18

18

Angebot	Erwachsene	Jugendliche
-----+-----+-----		
Basispreis	0	0
Erm. durch Gutschein	14	0
Freie Eintritte	0	0
Erm. durch Tageskarte	0	0

	Anzahl
-----+-----	
Tageskarten	0
2-2-Familien	0
1-3-Familien	0

Gesamtpreis: 44.10€

beispiel3b.txt

False

False

1 # Es wird angenommen dass jeder Verein einen Gutschein hat

18 # 18 Erw.

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

Angebot	Erwachsene	Jugendliche
-----+-----+-----		
Basispreis	0	0
Erm. durch Gutschein	18	0
Freie Eintritte	0	0
Erm. durch Tageskarte	0	0

	Anzahl
-----+-----	
Tageskarten	0
2-2-Familien	0
1-3-Familien	0

Gesamtpreis: 56.70€

beispiel3c.txt

False

False

2

18 # 32 = 14 + 18 Erwachsene

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

Angebot	Erwachsene	Jugendliche
-----+-----+-----		
Basispreis	0	0
Erm. durch Gutscheine	31	0
Freie Eintritte	1	0
Erm. durch Tageskarte	0	0

	Anzahl
-----+-----	
Tageskarten	0
2-2-Familien	0
1-3-Familien	0

Gesamtpreis: 97.65€

Fazit:

$44.10\text{€} + 56.70\text{€} - 97.65\text{€} = 3.15\text{€}$
Die Vereine sparen zusammen 3.15€.

Quellcode

```
#!/usr/bin/python3

# 36. BWInf
# 2. Aufgabe - Schwimmbad
# Kamal Abdellatif

from itertools import product
from math import ceil
import sys

COUPON, DAYCARD, FAMILY, CHILD, TEEN, ADULT = range(6)

week_discount = .2 # 20% Wochenrabatt
coupon_discount = .1 # 10% Couponrabatt

prices = { # Preise in Euro
    TEEN: 2.5,
    ADULT: 3.5,
    DAYCARD: 11.0,
    FAMILY: 8.0,
}

# Implementation des kartesischen Produkts von Intervallen ]0;n_i] mit Obergrenzen n_0, ...,
combinations = lambda *sizes: product( *(range(n+1) for n in sizes) )

def minimum_price(teens, adults, children, coupons, week=True, holiday=False):
    """Gibt die Belegung fuer eine gegebene Situation aus"""
    if children and not adults:
        return None # Keine Aufsichtspersonen

    week_f = 1 - (week_discount if week else 0) # Wochenrabatt
    coupon_f = 1 - coupon_discount # Couponrabatt
    people = teens + adults # Anzahl von Personen
    # Obere Grenzen der Anzahlen von betreffenden Personen bezgl. eines Angebots
    parameters = (
        people if coupons > 0 and not holiday else 0, # d
        min(people, coupons) if not holiday else 0, # h
        people if week else 0, # t
        min(adults//2, teens//2), # f_22
        min(adults, teens//3), # f_13
    )
    min_price = float("inf")
    min_combination = None
    for discounts, freepasses, daycards, _22families, _13families in combinations(*parameters):
        if discounts > 0:
            freepasses -= 1 # Ein Coupon wird durch die ermaessigung genutzt.

    for discount_a, freepass_a, daycard_a in combinations(discounts, freepasses, daycards):
        # _a fuer betreffliche Erwachsene
        # _t fuer betreffliche Jugendliche
```

```

discount_t = discounts - discount_a
freepass_t = freepasses - freepass_a
daycard_t = daycards - daycard_a
base_a = adults - (2*_22families+1*_13families+discount_a+freepass_a+daycard_a)
base_t = teens - (2*_22families+3*_13families+discount_t+freepass_t+daycard_t)

# Testen ob die Anzahl von Personen nicht ueberschritten wird
if base_a >= 0 and base_t >= 0:
    # Implementation der Preisfunktion
    price = base_a*prices[ADULT]*week_f + base_t*prices[TEEN]*week_f \
        + (_22families+_13families)*prices[FAMILY] + ceil(daycards/6)*prices[DAYCARD] \
        + (discount_a*prices[ADULT]+discount_t*prices[TEEN])*coupon_f*week_f

    if price < min_price:
        min_price = price
        min_combination = {
            "price": price,
            "discount_a": discount_a,
            "discount_t": discount_t,
            "freepass_a": freepass_a,
            "freepass_t": freepass_t,
            "daycard_a": daycard_a,
            "daycard_t": daycard_t,
            "_22families": _22families,
            "_13families": _13families,
            "base_a": base_a,
            "base_t": base_t,
            "daycards": ceil(daycards/6)
        }

return min_combination

if __name__ == "__main__":
    # Einlesen der Datei
    with open(sys.argv[1], 'r') as fobj:
        adults = 0
        teens = 0
        children = 0

        week = eval(fobj.readline())
        holiday = eval(fobj.readline())
        coupons = eval(fobj.readline()) # eval statt int fuer Kommentare

        for age_s in fobj.readlines():
            if not age_s:
                break
            age = eval(age_s)
            if age < 4:
                children += 1

            elif age <= 16:
                teens += 1

```



```

else:
    adults += 1

# Haupt-Funktionsaufruf
solution = minimum_price(teens, adults, children, coupons, week, holiday)

fmt_template = """
        Angebot / Erwachsene / Jugendliche
-----+-----+-----
        Basispreis / {base_a:^10d} / {base_t:^11d}
Erm. durch Gutschein / {discount_a:^10d} / {discount_t:^11d}
        Freie Eintritte / {freepass_a:^10d} / {freepass_t:^11d}
Erm. durch Tageskarte / {daycard_a:^10d} / {daycard_t:^11d}

                / Anzahl
-----+-----
        Tageskarten / {daycards:^6d}
        2-2-Familien / {_22families:^6d}
        1-3-Familien / {_13families:^6d}

Gesamtpreis: {price:.2f}€
"""

# Ausgeben der Loesung
if solution is None:
    print("Kinder bleiben unbeaufsichtigt! Kein Eintritt.")

else:
    print(fmt_template.format(**solution))

```
