DATV: Introduktion til optimering og operationsanalyse, 2007

Bin Packing Problemet

David Pisinger, *Projektopgave 2*

Dette er den anden obligatoriske projektopgave på kurset “DATV: Introduktion til optimering og operationsanalyse”. Opgaven stilles fredag 9. marts 2007 og skal afleveres senest tirsdag 20. marts 2007 kl. 11.00 i DIKU’s studieadministration. For at blive godkendt skal der være gjort et reelt forsøg på at løse samtlige spørgsmål. Besvarelsen skal udarbejdes i grupper på to til tre deltagere. Grupper med én deltager kræver skriftlig accept fra instruktoren mindst en uge før opgaven skal afleveres. Læs venligst hele opgaveformuleringen igennem inden du går i gang. Hints til opgaverne kan fås ved øvelserne, hvor der er afsat tid til at arbejde med projektopgaven.

Indledning

Bin packing problemet er et vigtigt optimeringsproblem indenfor produktionsplanlægning, pakning m.m. Bin packing problemet er NP-hårdt at løse [1].

Formelt kan *bin packing problemet* defineres på følgende vis: Lad der være givet n genstande som hver har en tilknyttet vægt w_j. Lad der endvidere være givet et uendeligt antal beholdere (*bins*) der hver kan rumme vægten c (*kapaciteten*). Opgaven er nu at fordele genstandene i beholdere, så ingen beholderes kapacitet overskrides, og således at der benyttes færrest mulige beholdere.

Hvis vi bruger binære variable x_{ij} til at angive om en genstand j anbringes i beholder i, og v_i til at angive om beholder i benyttes, får vi følgende matematiske formulering af problemet:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} v_i && (1)\\
\text{subject to} \quad & \sum_{j=1}^{n} w_j x_{ij} \le c v_i, \quad i = 1,\ldots,n && (2)\\
& \sum_{i=1}^{n} x_{ij} \ge 1, \quad j = 1,\ldots,n && (3)\\
& x_{ij} \in \{0,1\}, \quad i,j = 1,\ldots,n && (4)\\
& v_i \in \{0,1\}, \quad i = 1,\ldots,n && (5)
\end{aligned}$$

Vi vil betegne formuleringen (1) – (5) for en *simple formulering*. Objektfunktionen (1) angiver at vi skal bruge færrest mulige beholdere, mens begrænsning (2) angiver at hvis $v_i = 0$ så kan der ikke pakkes noget i beholderen og hvis $v_i = 1$ så kan der pakkes kapaciteten c. Begrænsning (3) sikrer at hver genstand j bliver pakket, og begrænsningerne (4) og (5) sikrer at alle beslutningsvariable er boolske.

Det antages normalt at alle koefficienter w_j og c er positive heltal. Det antages endvidere at $w_j \le c$ for alle j da man ellers ikke kan finde en lovlig løsning.

Eksempel 1 I det følgende eksempel er $c = 9$ og der er givet $n = 7$ genstande med følgende vægte:

j	1	2	3	4	5	6	7
w_j	2	4	6	5	8	4	6

■

Opgave 1 Løs instansen fra eksempel 1 til optimalitet ved brug af CPLEX. Rapporter den fundne løsning samt antal branch-and-bound knuder som CPLEX brugte (angiv *Nodes* og *Iterations*). ■

En god heuristik til at løse bin packing problemet er *first fit decreasing*. Her sorters genstandene efter aftagende vægt, og alle beholdere er til at begynde med tomme. Nu betragtes genstandene i den sorterede rækkefølge. Hver genstand j anbringes i den første beholder (dvs. den beholder med lavest nummer) hvor der er plads.

Opgave 2 Anvend first fit decreasing heuristikken på eksempel 1 og vis at man herved finder følgende løsning

bin	genstande	vægt
1	5	8
2	3,1	6+2
3	7	6
4	4,2	5+4
5	6	4

Giv tilstrækkeligt med detaljer til at man kan følge algoritmen. ■

Hvis man kender en øvre grænse m for antal beholdere, der skal benyttes i en optimal løsning, kan dette udnyttes til at begrænse antal beslutningsvariable v_j således at man kun har disse variable for $j = 1, \dots, m$. Dette kan reducere størrelsen af modellen (1) – (5) betragteligt.

Grænseværdier

Hvis vi LP-relaxerer bin packing problemet (1) – (5) får vi følgende problem

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} v_i && (6)\\
\text{subject to} \quad & \sum_{j=1}^{n} w_j x_{ij} \le c v_i, \quad i = 1, \dots, n && (7)\\
& \sum_{i=1}^{n} x_{ij} \ge 1, \quad j = 1, \dots, n && (8)\\
& 0 \le x_{ij} \le 1, \quad i, j = 1, \dots, n && (9)\\
& 0 \le v_i \le 1, \quad i = 1, \dots, n && (10)
\end{aligned}$$

Løsningsværdien til LP-relaxeringen betegnes z^{LP}.

Opgave 3 Find z^{LP} for instansen fra eksempel 1. ■

Det er oplagt at z^{LP} giver en nedre grænseværdi for bin packing problemet. Men desværre er kvaliteten af denne relativt dårlig. Man kan opnå en strammere grænseværdi ved at omskrive problemet som følger:

Opgave 4 Opskriv for instansen i eksempel 1 samtlige måder en beholder kan pakkes på. (Dvs. angiv i tabel-form de valgte genstande, og deres vægt-sum). ■

Antag at der findes to lovlige pakninger P_1 og P_2 hvor samtlige genstande i pakning P_1 også indgår i pakning P_2, men pakning P_2 yderligere indeholder en eller flere genstande. Så kan vi slette pakning P_1 idet vi ikke er dårligere stillet ved at anvende pakning P_2 i stedet. (Vi kan altid smide ekstra genstande væk i en løsning).

Opgave 5 Brug denne observation til at fjerne nogle pakninger fra forrige opgave og opskriv de tilbageværende pakninger. ■

Lad R betegne mængden af pakninger af en enkelt beholder. Lad endvidere a_{ij} angive om genstand j indgår i pakning i. Vi kan benytte dette til at opskrive en alternativ formulering af bin packing problemet, som vi betegner *kolonne formuleringen*.

Lad x'_i angive om pakning $i \in R$ benyttes. Dermed bliver modellen:

$$\min \quad \sum_{i \in R} x'_i \tag{11}$$

$$\text{s.t} \quad \sum_{i \in R} a_{ij} x'_i \geq 1 \quad j = 1, \ldots, n \tag{12}$$

$$x'_i \in \{0, 1\} \quad i \in R \tag{13}$$

Her sikrer (12) at hver genstand j bliver pakket, mens (13) sikrer at hver pakning bliver brugt nul eller en gang.

Opgave 6 Opskriv kolonne formuleringen af eksempel 1 (gerne reduceret i henhold til opgave 5) ■

Opgave 7 Løs kolonne formuleringen af eksempel 1 til LP-optimalitet med CPLEX. ■

Opgave 8 Løs kolonne formuleringen af eksempel 1 til IP-optimalitet med CPLEX. Rapporter antal branch-and-bound knuder som CPLEX brugte (angiv *Nodes* og *Iterations*). ■

Opgave 9 Bevis at man får strammere grænseværdier ved at løse LP-relaxeringen af kolonne formuleringen end ved at løse LP-relaxeringen af den simple formulering. (Hint: Idet de to formuleringer har samme objektfunktion skal man blot vise at løsningsrummet for det ene problem er skarpt indeholdt i det andet problem. Man skal m.a.o. vise at enhver løsning til kolonne formuleringen også er en løsning til den simple formulering. Endvidere skal man vise at der findes mindst en instans hvor kolonne formuleringen giver en strammere grænseværdi end den simple formulering). ■

Desværre kan der være eksponentielt mange måder hvorpå en enkelt beholder kan pakkes, og dermed bliver kolonne formuleringen i praksis uløselig grundet det store antal beslutningsvariable. Hvis man kun er interesseret i at løse LP-relaxeringen af kolonne formuleringen (11) – (13), kan problemet løses ved at generere kolonnerne efterhånden som der bliver brug for dem. Man kan da håbe på at problemet kan løses til LP-optimalitet uden at generere mere end en lille brøkdel af kolonnerne.

For at illustrere princippet, betragt igen den LP-relaxerede kolonne formulering af eksempel 1, hvor vi kun betragter pakningerne fra first fit decreasing heuristikken. Dette fører til følgende LP-model

$$\begin{array}{lcccccccccccc} \min & x_1' & + & x_2' & + & x_3' & + & x_4' & + & x_5' & & \\ \text{s.t.} & & & x_2' & & & & & & & \geq & 1 \\ & & & & & & & x_4' & & & \geq & 1 \\ & & & x_2' & & & & & & & \geq & 1 \\ & & & & & & & x_4' & & & \geq & 1 \\ & x_1' & & & & & & & & & \geq & 1 \\ & & & & & & & & & x_5' & \geq & 1 \\ & & & & & x_3' & & & & & \geq & 1 \end{array} \tag{14}$$

Lad y_j betegne den duale variabel svarende til begrænsning (12) for genstand j. De duale variable for ovenstående eksempel er $y_1 = y_2 = y_5 = y_6 = y_7 = 1$ mens $y_3 = y_4 = 0$.

Opgave 10 Giv en intuitiv fortolkning af de duale variable y_j. ■

Opgave 11 Idet den intuitive fortolkning af de duale variable udnyttes, find ved inspektion den mest lovende pakning $i \in R$, fra opgave 5 som skal tilføjes modellen. ■

Man behøver ikke at kende mængden R for at løse problemet fra opgave 11.

Opgave 12 Vis at problemet med at finde den mest lovende pakning $i \in R$ som skal tilføjes til modellen kan formuleres som et knapsack problem. Dette problem kaldes *pricing problemet*. ■

Opgave 13 Angiv et kriterie for hvornår det ikke kan betale sig at medtage en ny pakning af en beholder (Hint: betragt den reducerede omkostning af en pakning). ■

Opgave 14 Udvid ovenstående kolonne formulering (14) med den nye pakning som blev fundet i opgave 11. Løs det udvidede problem (14) med CPLEX og bestem de nye duale variable $y_1, \ldots, y_7$. ■

Opgave 15 Gentag processen med at finde den mest lovende pakning ved brug af metoden fra opgave 12, tilføj den til modellen (14) og bestem de nye duale variable. Stands processen når stop-kriteriet fra opgave 13 nås. Giv tilstrækkeligt med detaljer i hvert skridt. ■

Opgave 16 Sammenlign de to løsninger fra opgave 7 og 15 med hensyn til LP-løsningsværdi og størrelse af den endelige model. ■

Noter

Til opgaven benyttes CPLEX. Da der kun er nogle få CPLEX-licenser til rådighed på DIKU bedes man logge ud fra CPLEX relativt hurtigt efter at have kørt sin instans. CPLEX licenser er tilgængelige på `kand` og `bach` maskinerne.

CPLEX kan drille når man vil finde de duale variable: Hvis man bruger formuleringen

```
bounds
0 <= x1 <= 1
```

vil CPLEX have en usynlig dual variabel knyttet til begrænsningen. Dette kan medføre at de reducerede omkostninger udregnes forkert.

Det nemmeste er helt at udelade grænser på variablene. Det er ikke nødvendigt at sikre $x_i' \leq 1$ (da vi minimerer), og $x_i' \geq 0$ er underforstået for LP-problemer.

Litteratur

[1] http://en.wikipedia.org/wiki/Bin_packing_problem

[2] L. A. Wolsey, Integer Programming, Wiley, Chichester, UK, 1999.