

Projektinformation

TAOP61 Optimering av realistiska sammansatta system

Projekt 1: Modellering och lösning av ett komplext planeringsproblem

1 Uppgift

Uppgiften i projekt 1 är modellering och lösning av ett komplext planeringsproblem. Ni ska använda paketet GLPK, vilket innehåller modelleringspråket GMPL samt lösaren glpsol.

Ni ska alltså formulera problemet som en matematisk optimeringsmodell, med variabeldefinitioner, bivillkor och målfunktion. Därefter skriver ni in modellen på fil i GMPL-format.

2 Att göra

1. Studera beskrivningen av problemet.
2. Studera informationen om GMPL och GLPK.
3. Formulera problemet som en matematisk optimeringsmodell.
4. Lös problemen som beskrives nedan.
5. Redovisa genom att skriva en rapport om resultaten som svarar på alla frågor och kan läsas av både en som inte kan optimering och en som kan det och är nyfiken på hur ni gjorde. Inkludera er matematiska modell.
6. Ev: Förbered en redovisning av resultaten/rapporten.

3 Problembeskrivning

Det allmänna situationen för Returpack AB beskrivs separat.

I detta projekt ska man göra en matematisk formulering av ett något förenklat problem. Vi betraktar bara varuflödet, inte penningaffödet. Dessutom bortser vi från att flödet egentligen består av olika sorters förpackningar (metallburkar och petflaskor), utan ser flödet av tomförpackningar som *en* sorts flöde. Dock sker en omvandling till råmaterial vid råvarutillverkarna, samt en omvandling från råmaterial till nya förpackningar vid förpackningstillverkarna. Vid dessa omvandlingar används vissa omvandlingsfaktorer, en

enhet flöde in blir inte exakt en enhet flöde ut. En liknande omvandlingsfaktor uppstår då Returpack komprimerar tomförpackningarna.

Rent konkret ska vi modellera följande situation. Vid butikerna uppstår viss efterfrågan av drycker. Motsvarande mängd tomförpackningar lämnas in till butikerna, minus de tomförpackningar som slängs på andra sätt, vilket vi kallar spill. Tomförpackningarna skickas till grossister eller bryggerier, och grossisterna och bryggerierna skickar tomförpackningarna till Returpacks huvudanläggning. Där komprimeras allt och skickas vidare till råvarutillverkare. Råvarutillverkarna producerar nytt material, som skickas till förpackningstillverkare, som producerar nya förpackningar som skickas till bryggerier, där de fylls med dryck och skickas vidare till butikerna, ev. via grossisterna.

Man funderar på att inrätta nya mellanlager, som ersätter grossister och/eller bryggerier, så att tomförpackningar skickas direkt från butiker till mellanlager och sedan vidare direkt till Returpack. Ett antal möjligheter för mellanlager har tagits fram, och frågan är vilka av dessa möjligheter som ska realiseras. Poängen är att transportererna blir billigare och kortare, vilket är bra för miljön.

Vi har alltså ett cirkulerande flöde, som dock bryts av tomförpackningar som inte pantas, samt av omvandlingsfaktorerna. Av denna anledning behöver råvarutillverkarna köpa nytt material för att fylla på med så att efterfrågan kan tillgodoses.

Grossister, bryggerier och mellanlager har begränsade kapaciteter för tomförpackningshantering, samt s.k. fasta kostnader för hanteringen. En sådan kostnad är inte proportionell mot den hanterade mängden, men försvinner om inga tomförpackningar alls hanteras där. För övrigt arbetar vi med linjära kostnader för flöde, proportionellt mot avståndet mellan platserna, för att minimera avgaser och andra miljömässiga nackdelar med transportererna.

Huvuduppgiften är att jämföra situationen med och utan mellanlager. Vi förutsätter att transportererna sker så att kostnaderna minimeras. Man vill veta vilka mellanlager som skall byggas, så det finns ett inslag av lokaliseringsproblem i det hela.

Man är också intresserad av hur känslig lösningen är för förändringar i spillet. Ska man satsa på att försöka minska spillet i återvinningen?

Ett flertal instanser av olika storlekar ska lösas, från små exempel som kan användas för att kontrollera modellen, till stora, riktigt svåra instanser. (Alla är dock mindre än det riktiga problemet.)

Först löser man problemet utan mellanlager (genom att sätta kapaciterna för mellanlager till noll i modellfilen), vilket ger en bild av nuläget (eftersom vi förutsätter att varorna skickas så att kostnaden minimeras). Därefter löses problemet med möjliga mellanlager och optimeringen bestämmer vilka lager som ska byggas.

Det viktigaste slutresultatet är hur mycket man tjänar på att införa mellanlager och vilka som ska byggas, samt den miljömässiga vinsten, som ges av målfunktionsvärdet för den linjära delen av problemet (dvs. förutom de fasta kostnaderna).

Den andra frågan gäller hur förändringar i spillet förändrar totalkostnaden och inköpet av råvaror. För att få viss kunskap om detta, kan man lösa problemet för det givna värdet på spillet samt för inget spill alls. (Det senare görs enkelt genom att multiplicera spillet i modellen med koefficienten noll.)

3.1 Indata

Indata ges i datafiler i GMPL-format. Följande parametrar ges: **nbutik**: antal butiker, **ngross**: antal grossister, **nbrygg**: antal bryggerier, **nmatr**: antal råvarutillverkare, **nfpack**: antal förpackningstillverkare, **nlager**: antal möjliga mellanlager.

tomkost: faktor att multiplicera avståndet med för att få kostnaden för att transportera en tomförpackning (dvs. flödet från butikerna till Returpack), **fullkost**: faktor att multiplicera avståndet med för att få kostnaden för att transportera en enhet material eller fylld förpackning (dvs. flödet efter Returpack tillbaka till butikerna), **lagkost**: faktor att multiplicera avståndet med för att få kostnaden för att transportera en tomförpackning från butiker till mellanlager och sedan vidare till Returpack (den nya bilarna ger billigare transporter, vilket ses i skillnaden mellan **lagkost** och **tomkost**), **bristkost**: kostnad (per enhet) som uppstår då efterfrågan inte kan tillgodoses (används ej), **matrkop**: kostnad att köpa en enhet råvarumaterial, **maxlager**: det maximala antal mellanlager som får byggas.

behov: efterfrågan hos varje butik, **netto**: antal tomförpackningar som försvinner, dvs. inte lämnas tillbaka till butiken (spillet).

gkaptom: kapaciteten för tomförpackningshantering hos varje grossist, **gkapfull**: kapaciteten för hantering av fyllda förpackningar hos varje grossist (används ej), **bkaptom**: kapaciteten för tomförpackningshantering hos varje bryggeri, **bkapfull**: kapaciteten för hantering av fyllda förpackningar hos varje bryggeri (används ej), **lkaptom**: kapaciteten för tomförpackningshantering hos varje möjligt mellanlager. (Kapaciteterna för transporter av fulla förpackningar ska ej ingå i modellen.)

fkostG: en fast kostnad för tomförpackningshantering hos varje grossist, **fkostB**: en fast kostnad för tomförpackningshantering hos varje bryggeri, **fkostL**: en fast kostnad för tomförpackningshantering hos varje mellanlager, vilket inkluderar kostnaden för att bygga/inrätta lagret (diskonterat). Pga. fördyringar ska **fkostL** multipliceras med 10.

faktR: kompressionsfaktor vid Returpack (utflödet är **faktR** gånger inflödet), **faktM**: omvandlingsfaktor hos råvarutillverkarna (utflödet är **faktM** gånger inflödet), **faktF**: omvandlingsfaktor hos tomförpackningstillverkarna (utflödet är **faktF** gånger inflödet).

distBuG: avståndet mellan butik och grossist, **distBuB**: avståndet mellan butik och bryggeri, **distGR**: avståndet från grossist till Returpack, **distBR**: avståndet från bryggeri till Returpack, **distTB**: avståndet från tomförpackningstillverkare till bryggeri, **distGB**: avståndet mellan grossist och bryggeri, **distRM**: avståndet från Returpack till råmaterialtillverkare, **distMT**: avståndet från råvarutillverkare till tomförpackningstillverkare, **distBuL**: avståndet från butik till mellanlager, **distLR**: avståndet från mellanlager till Returpack.

Studera datafilen för exemplet *ex1.dat* för att se formatet på dessa data.

I filen *initmodel.mod* ges en föreslagen början på modellfilen. Där definieras indexmängder samt formatet på alla indata.

Det som återstår är att definiera variablerna, samt att teckna målfunktion och bivillkor.

3.2 Instanser

Ett antal olika instanser av problemet har tagits fram: ($|N|$ är antalet noder i nätverket, $|A|$ är antalet bågar och n är antalet variabler i problemet)

Problem	nbutik	ngross	nbrygg	nmatr	nfpack	nlager	$ N $	$ A $	n
ex1	4	3	2	2	2	3			90
ex3	5	3	2	2	2	4	19	70	111
ex5	20	5	3	3	3	4	39	288	483
ex6	100	10	5	3	3	10	132	2602	4230
ex7	500	30	5	3	5	20	564	27 748	45 806
ex9	1000	30	5	3	5	100	1144	135 328	171 466
ex10	1000	30	5	3	5	100	1144	135 328	171 466

Exemplet *ex1.dat* kommer inte från en geografisk karta, så avstånden är inte naturliga. I exemplen *ex9.dat* och *ex10.dat* har man tagit fram betydligt flera alternativ för mellanlager än man behöver, för att ha något att välja mellan. (Det gör dock problemet svårare att lösa.)

Praktiskt tips: De största instanserna är svåra att lösa. Låt inte programmet hålla på för länge. En uppsnabbning kan ske genom att lösa LP-relaxationen (m.h.a. *--nomip*) och avrunda resultatet, eller genom att tillåta approximativa kapningar i trädsökningen (m.h.a. t.ex. *--mipgap 0.01*). Man får dock då inte exakt optimum.

4 Deluppgifter

1. Lös det lilla testproblemet *ex1.dat* för att se att allt fungerar korrekt. (Optimalt målfunktionsvärde är 104850.6467, och man ska bygga mellanlager 2.)
2. Lös de större instanserna. Gör följande för varje exempel.

Notera antalet trädsökningsnoder och antal simplexiterationer, dels för första LP-problemet och dels totalt för att lösa problemet. Notera tiden det tog. Notera antal bryggerier, grossister och mellanlager som används för tomförpackningshantering. Notera målfunktionsvärde (med och utan fasta kostnader).

- a) Lös problemet utan mellanlager (sätt kapaciteten till noll i modellfilen).
- b) Lös problemet med mellanlager. Notera vilka mellanlager som ska byggas och om några grossister och bryggerier används för tomförpackningshantering. Notera även hur mycket nytt råvarumaterial som köpes in.
- c) Ange hur mycket man tjänar på att bygga mellanlager, dels totalt och dels uppdelat i fasta kostnader och linjära kostnader (vilket motsvarar miljöeffekten).
- d) Lös problemet med spill noll (och med mellanlager). Notera målfunktionsvärde samt hur mycket nytt råvarumaterial som köpes in.
- e) Ange hur mycket man skulle tjäna på att få ner spillet till noll, dels totalt och dels miljömässigt. Förändras slutsatsen om råvarupriset multipliceras med 100?
- f) Om det tog för lång tid att lösa problemet exakt, ange hur ni gjorde och vilken osäkerhet resultatet kan ha.