

# TAOP61 Projekt 1

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringspråket GMPL samt lösaren glpsol.

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringsspråket GMPL samt lösaren glpsol.

Material: Problembeskrivning: Returpack.

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringsspråket GMPL samt lösaren glpsol.

Material: Problembeskrivning: Returpack.

Uppgifter:

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringsspråket GMPL samt lösaren glpsol.

Material: Problembeskrivning: Returpack.

Uppgifter:

Formulera problemet som en matematisk optimeringsmodell.

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringsspråket GMPL samt lösaren glpsol.

Material: Problembeskrivning: Returpack.

Uppgifter:

Formulera problemet som en matematisk optimeringsmodell.

Skriv in modellen på fil i GMPL-format.

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringsspråket GMPL samt lösaren glpsol.

Material: Problembeskrivning: Returpack.

Uppgifter:

Formulera problemet som en matematisk optimeringsmodell.

Skriv in modellen på fil i GMPL-format.

Lös det lilla testproblemet för att se att allt fungerar korrekt.



# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringsspråket GMPL samt lösaren glpsol.

Material: Problembeskrivning: Returpack.

Uppgifter:

Formulera problemet som en matematisk optimeringsmodell.

Skriv in modellen på fil i GMPL-format.

Lös det lilla testproblemet för att se att allt fungerar korrekt.

Lös de stora problemen.

# TAOP61 Projekt 1

Modellering och lösning av ett komplext planeringsproblem.

Programvara: GLPK, dvs. modelleringsspråket GMPL samt lösaren glpsol.

Material: Problembeskrivning: Returpack.

Uppgifter:

Formulera problemet som en matematisk optimeringsmodell.

Skriv in modellen på fil i GMPL-format.

Lös det lilla testproblemet för att se att allt fungerar korrekt.

Lös de stora problemen.

Skriv en rapport om resultaten.

## TAOP61 Projekt 1: Returpack

Konsumenter lämnar in tomburkar och flaskor i butiker. Dessa förpackas och skickas till en grossist. Därefter skickas varorna vidare till Returpack.

## TAOP61 Projekt 1: Returpack

Konsumenter lämnar in tomburkar och flaskor i butiker. Dessa förpackas och skickas till en grossist. Därefter skickas varorna vidare till Returpack.

Ett visst flöde går även från butik till bryggeri, som sedan levererar till Returpack.

## TAOP61 Projekt 1: Returpack

Konsumenter lämnar in tomburkar och flaskor i butiker. Dessa förpackas och skickas till en grossist. Därefter skickas varorna vidare till Returpack.

Ett visst flöde går även från butik till bryggeri, som sedan levererar till Returpack.

Returpack skickar varorna till råmaterialtillverkare, som återanvänder materialet, och skickar det till förpackningstillverkare som gör nya förpackningar. Förpackningarna levereras till bryggerier, som skickar dem (med dryck i) till grossister och/eller direkt till butiker. Där inhandlas de av konsumenter, och innehållet dricks upp.

## TAOP61 Projekt 1: Returpack

Konsumenter lämnar in tomburkar och flaskor i butiker. Dessa förpackas och skickas till en grossist. Därefter skickas varorna vidare till Returpack.

Ett visst flöde går även från butik till bryggeri, som sedan levererar till Returpack.

Returpack skickar varorna till råmaterialtillverkare, som återanvänder materialet, och skickar det till förpackningstillverkare som gör nya förpackningar. Förpackningarna levereras till bryggerier, som skickar dem (med dryck i) till grossister och/eller direkt till butiker. Där inhandlas de av konsumenter, och innehållet dricks upp.

Returpack funderar på att ersätta grossisterna roll i flödet med egna mellanlager. Dessutom funderar man på att inte transportera tomförpackningarna i kartonger, utan i plastsäckar, som kan hämtas med något som liknar en sopbil.

## TAOP61 Projekt 1: Returpack

Grossister, bryggerier och mellanlager har begränsade kapaciteter för tomförpackningshantering, samt s.k. *fasta kostnader* för hanteringen.

## TAOP61 Projekt 1: Returpack

Grossister, bryggerier och mellanlager har begränsade kapaciteter för tomförpackningshantering, samt s.k. *fasta kostnader* för hanteringen.

För övrigt arbetar vi med linjära kostnader för flöde, proportionellt mot avståndet mellan platserna, för att minimera avgaser och andra miljömässiga nackdelar med transporterna.



## TAOP61 Projekt 1: Returpack

Grossister, bryggerier och mellanlager har begränsade kapaciteter för tomförpackningshantering, samt s.k. *fasta kostnader* för hanteringen. För övrigt arbetar vi med linjära kostnader för flöde, proportionellt mot avståndet mellan platserna, för att minimera avgaser och andra miljömässiga nackdelar med transporterna.

Fasta kostnader motsvarar alltså investeringskostnader, medan linjära kostnader motsvarar miljöpåverkan.

## TAOP61 Projekt 1: Returpack

Grossister, bryggerier och mellanlager har begränsade kapaciteter för tomförpackningshantering, samt s.k. *fasta kostnader* för hanteringen.

För övrigt arbetar vi med linjära kostnader för flöde, proportionellt mot avståndet mellan platserna, för att minimera avgaser och andra miljömässiga nackdelar med transporterna.

Fasta kostnader motsvarar alltså investeringskostnader, medan linjära kostnader motsvarar miljöpåverkan.

Huvuduppgiften är att jämföra situationen med och utan mellanlager. Man vill veta vilka mellanlager som skall byggas.

## TAOP61 Projekt 1: Returpack

Grossister, bryggerier och mellanlager har begränsade kapaciteter för tomförpackningshantering, samt s.k. *fasta kostnader* för hanteringen. För övrigt arbetar vi med linjära kostnader för flöde, proportionellt mot avståndet mellan platserna, för att minimera avgaser och andra miljömässiga nackdelar med transporterna.

Fasta kostnader motsvarar alltså investeringskostnader, medan linjära kostnader motsvarar miljöpåverkan.

Huvuduppgiften är att jämföra situationen med och utan mellanlager. Man vill veta vilka mellanlager som skall byggas.

Först löser man problemet utan mellanlager. Därefter löses problemet med möjliga mellanlager och optimeringen bestämmer vilka lager som ska byggas.

## TAOP61 Projekt 1: Returpack

Grossister, bryggerier och mellanlager har begränsade kapaciteter för tomförpackningshantering, samt s.k. *fasta kostnader* för hanteringen.

För övrigt arbetar vi med linjära kostnader för flöde, proportionellt mot avståndet mellan platserna, för att minimera avgaser och andra miljömässiga nackdelar med transporterna.

Fasta kostnader motsvarar alltså investeringskostnader, medan linjära kostnader motsvarar miljöpåverkan.

Huvuduppgiften är att jämföra situationen med och utan mellanlager. Man vill veta vilka mellanlager som skall byggas.

Först löser man problemet utan mellanlager. Därefter löses problemet med möjliga mellanlager och optimeringen bestämmer vilka lager som ska byggas.

Man vill också veta effekten av spill, dvs. tomburkar som inte lämnas till återvinning. Därför löses problemet även utan spill.

## TAOP61 Projekt 1: Indata

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

## TAOP61 Projekt 1: Indata

Indata ges i datafiler i GMP-format.  $n_{butik}$ : antal butiker,  $n_{gross}$ : antal grossister,  $n_{brygg}$ : antal bryggerier,  $n_{matr}$ : antal råvarutillverkare,  $n_{fpack}$ : antal förpackningstillverkare,  $n_{lager}$ : 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,  $matrkop$ : kostnad att köpa en enhet råvarumaterial,  $maxlager$ : det maximala antal mellanlager som får byggas.

## TAOP61 Projekt 1: Indata

behov: efterfrågan hos varje butik, netto: spill, antal tomförpackningar som försvinner, dvs. inte lämnas tillbaka till butiken.

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.

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). Ska multipliceras med 10.

## TAOP61 Projekt 1: Indata

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 Returpak till råmaterialtillverkare, distMT: avståndet från råvarutillverkare till tomförpackningstillverkare, distBuL: avståndet från butik till mellanlager, distLR: avståndet från mellanlager till Returpack.



## TAOP61 Projekt 1: Indata

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 Returpak till råmaterialtillverkare, distMT: avståndet från råvarutillverkare till tomförpackningstillverkare, distBuL: avståndet från butik till mellanlager, distLR: avståndet från mellanlager till Returpack.

I filen *initmodel.mod* ges en föreslagen början på modellfilen.

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

# TAOP61 Projekt 1: Indata

```
param nbutik := 4; param ngross := 3; param nbrygg := 2; param nmatr := 2; param nfpack :=
2; param nlager := 3;
param tomkost := 2; param fullkost := 3; param lagkost := 1; param bristkost := 99999999;
param matrkop := 1; param maxlager := 4;
param : behov := 1 100 2 100 3 100 4 100; param : netto := 1 1 2 2 3 1 4 2;
param : gkaptom := 1 500 2 500 3 500; param : gkapfull := 1 500 2 500 3 500;
param : bkaptom := 1 500 2 500; param : bkapfull := 1 500 2 500;
param : lkaptom := 1 700 2 700 3 700;
param : fkostG := 1 100 2 100 3 100; param : fkostB := 1 30 2 10;
param : fkostL := 1 300 2 100 3 200;
param faktR := 0.7; param faktM := 0.9; param faktF := 1.5;
param distBuG : 1 2 3 := 1 10 13 22 2 13 13 26 3 20 23 29 4 17 23 12;
param distBuB : 1 2 := 1 20 43 2 33 53 3 40 23 4 17 23;
param : distGR := 1 12 2 22 3 14;
param : distBR := 1 23 2 15;
param distTB : 1 2 := 1 20 33 2 33 26;
param distGB : 1 2 := 1 20 22 2 33 26 3 40 29;
param : distRM := 1 23 2 18;
param distMT : 1 2 := 1 20 43 2 33 53;
param distBuL : 1 2 3 := 1 10 13 22 2 13 13 26 3 20 23 29 4 17 23 12;
param : distLR := 1 23 2 15 3 16;

end;
```

# TAOP61 Projekt 1: Instanser

Problem	nbu	ngr	nbr	nma	nfp	nla	$ N $	$ A $	$n$
ex1	4	3	2	2	2	3			90
ex2	10	3	3	2	2	4			213
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

## TAOP61 Projekt 1: Instanser

Problem	nbu	ngr	nbr	nma	nfp	nla	$ N $	$ A $	$n$
ex1	4	3	2	2	2	3			90
ex2	10	3	3	2	2	4			213
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

Exemplen *ex1.dat* och *ex2.dat* kommer inte från geografiska kartor, så avstånden är inte naturliga.

## TAOP61 Projekt 1: Instanser

Problem	nbu	ngr	nbr	nma	nfp	nla	$ N $	$ A $	$n$
ex1	4	3	2	2	2	3			90
ex2	10	3	3	2	2	4			213
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

Exemplen *ex1.dat* och *ex2.dat* kommer inte från geografiska kartor, så avstånden är inte naturliga.

I *ex9.dat* och *ex10.dat* har man tagit fram betydligt flera alternativ för mellanlager än man behöver.

# TAOP61 Projekt 1

Praktiskt tips 1: De största instanserna är svåra att lösa. Låt inte programmet hålla på för länge.

# TAOP61 Projekt 1

Praktiskt tips 1: De största instanserna är svåra att lösa. Låt inte programmet hålla på för länge.

Alt 1: Lös LP-relaxationen (m.h.a. `--nomip`) och avrunda resultatet.

# TAOP61 Projekt 1

Praktiskt tips 1: De största instanserna är svåra att lösa. Låt inte programmet hålla på för länge.

Alt 1: Lös LP-relaxationen (m.h.a. `--nomip`) och avrunda resultatet.

Alt 2: Tillåta approximativa kapningar i trädsökningen (m.h.a. t.ex. `--mipgap 0.01`).



# TAOP61 Projekt 1

Praktiskt tips 1: De största instanserna är svåra att lösa. Låt inte programmet hålla på för länge.

Alt 1: Lös LP-relaxationen (m.h.a. `--nomip`) och avrunda resultatet.

Alt 2: Tillåta approximativa kapningar i trädsökningen (m.h.a. t.ex. `--mipgap 0.01`).

Praktiskt tips 2: Om man vill hålla reda på separata delar av målfunktionsvärdet (linjära, fasta, material), är det smart att införa en variabel för varje del och minimera summan av dem.

# TAOP61 Projekt 1

Praktiskt tips 1: De största instanserna är svåra att lösa. Låt inte programmet hålla på för länge.

Alt 1: Lös LP-relaxationen (m.h.a. `--nomip`) och avrunda resultatet.

Alt 2: Tillåta approximativa kapningar i trädsökningen (m.h.a. t.ex. `--mipgap 0.01`).

Praktiskt tips 2: Om man vill hålla reda på separata delar av målfunktionsvärdet (linjära, fasta, material), är det smart att införa en variabel för varje del och minimera summan av dem.

Detta gäller även andra mängder man kan vilja hålla reda på.

# TAOP61 Projekt 1

Praktiskt tips 1: De största instanserna är svåra att lösa. Låt inte programmet hålla på för länge.

Alt 1: Lös LP-relaxationen (m.h.a. `--nomip`) och avrunda resultatet.

Alt 2: Tillåta approximativa kapningar i trädsökningen (m.h.a. t.ex. `--mipgap 0.01`).

Praktiskt tips 2: Om man vill hålla reda på separata delar av målfunktionsvärdet (linjära, fasta, material), är det smart att införa en variabel för varje del och minimera summan av dem.

Detta gäller även andra mängder man kan vilja hålla reda på.

Låt datorn räkna, så man själv slipper.

# TAOP61 Projekt 1

- ① Lös det lilla testproblemet *ex1.dat* för att se att allt fungerar korrekt.
- ② Lös de större. Notera följande: antalet trädsökningsnoder och antal simplexiterationer, dels för första LP-problemet och dels totalt, tiden det tog, antal bryggerier, grossister och mellanlager som används, 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 vilka grossister och bryggerier som används samt 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 (miljöeffekten).
  - d) Lös problemet med spill noll (och mellanlager). Notera 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) Ange om och hur problemet inte löstes exakt och vilken osäkerhet resultatet kan ha.