

TAOP61 Projekt 2

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

Material: Fallbeskrivning: Laddhybrid.

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

Material: Fallbeskrivning: Laddhybrid.

Uppgifter:

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

Material: Fallbeskrivning: Laddhybrid.

Uppgifter:

Formulera problemet för lösning med dynamisk programmering.

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

Material: Fallbeskrivning: Laddhybrid.

Uppgifter:

Formulera problemet för lösning med dynamisk programmering.

Gör ett program som löser problemet med dynamisk programmering.

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

Material: Fallbeskrivning: Laddhybrid.

Uppgifter:

Formulera problemet för lösning med dynamisk programmering.

Gör ett program som löser problemet med dynamisk programmering.

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

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

Material: Fallbeskrivning: Laddhybrid.

Uppgifter:

Formulera problemet för lösning med dynamisk programmering.

Gör ett program som löser problemet med dynamisk programmering.

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

Lös de stora problemen.

TAOP61 Projekt 2

Optimering av elmotorutnyttjandet i en laddhybrid med hjälp av dynamisk programmering.

Programvara: Matlab/Octave/Python, eget program.

Material: Fallbeskrivning: Laddhybrid.

Uppgifter:

Formulera problemet för lösning med dynamisk programmering.

Gör ett program som löser problemet med dynamisk programmering.

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

Lös de stora problemen.

Skriv en rapport om resultaten.

Laddhybrid



Laddhybrid



TAOP61 Projekt 2: Laddhybrid

Vi har fyra inställningar för framdrivnings sättet:

1. Bara el.
2. Mest el, med bidrag av bensin.
3. Lite el, med mera bensin.
4. Bara bensin.

TAOP61 Projekt 2: Laddhybrid

Vi har fyra inställningar för framdrivnings sättet:

1. Bara el.
2. Mest el, med bidrag av bensin.
3. Lite el, med mera bensin.
4. Bara bensin.

Vägarna delas upp i fem kategorier:

1. Motorväg.
2. Landsväg.
3. Mindre väg, backig och/eller kurvig.
4. Väg i gles tätort (liten stad eller förort).
5. Väg i innerstad (trafikljus, många start och stopp).

TAOP61 Projekt 2: Laddhybrid

Vi förutsätter att man har bestämt vilken väg man ska ta. Hela vägen delas upp i n stycken vägavsnitt av olika typer och olika längder. Vägavsnitt j har längden l_j (m) och är av typen t_j .

TAOP61 Projekt 2: Laddhybrid

Vi förutsätter att man har bestämt vilken väg man ska ta. Hela vägen delas upp i n stycken vägavsnitt av olika typer och olika längder. Vägavsnitt j har längden l_j (m) och är av typen t_j .

Baserat på förväntad medelhastighet har man uppskattat hur mycket batterinivån sjunker om man färdas en meter på en väg av kategori t med framdrivningssätt k , och det betecknas med b_{kt} . Dessutom har man beräknat en kostnad (inkluderande allt, dvs. bensin, slitage, miljöpåverkan etc) för att färdas en meter på en väg av kategori t med framdrivningssätt k , och den betecknas med d_{kt} .

TAOP61 Projekt 2: Laddhybrid

Vi förutsätter att man har bestämt vilken väg man ska ta. Hela vägen delas upp i n stycken vägavsnitt av olika typer och olika längder. Vägavsnitt j har längden l_j (m) och är av typen t_j .

Baserat på förväntad medelhastighet har man uppskattat hur mycket batterinivån sjunker om man färdas en meter på en väg av kategori t med framdrivningssätt k , och det betecknas med b_{kt} . Dessutom har man beräknat en kostnad (inkluderande allt, dvs. bensin, slitage, miljöpåverkan etc) för att färdas en meter på en väg av kategori t med framdrivningssätt k , och den betecknas med d_{kt} .

Eftersom vägavsnitt j har längden l_j , sjunker batterinivån med $a_{jk} = l_j b_{kt_j}$ om man kör den med inställning k , och kostnaden blir $c_{jk} = l_j d_{kt_j}$.

TAOP61 Projekt 2: Laddhybrid

Vi förutsätter att man har bestämt vilken väg man ska ta. Hela vägen delas upp i n stycken vägavsnitt av olika typer och olika längder. Vägavsnitt j har längden l_j (m) och är av typen t_j .

Baserat på förväntad medelhastighet har man uppskattat hur mycket batterinivån sjunker om man färdas en meter på en väg av kategori t med framdrivningssätt k , och det betecknas med b_{kt} . Dessutom har man beräknat en kostnad (inkluderande allt, dvs. bensin, slitage, miljöpåverkan etc) för att färdas en meter på en väg av kategori t med framdrivningssätt k , och den betecknas med d_{kt} .

Eftersom vägavsnitt j har längden l_j , sjunker batterinivån med $a_{jk} = l_j b_{kt_j}$ om man kör den med inställning k , och kostnaden blir $c_{jk} = l_j d_{kt_j}$.

Man kan använda sig av variablerna $x_{jk} = 1$ om vi kör vägavsnitt j med inställning k , och 0 om inte.

TAOP61 Projekt 2: Laddhybrid

Vi förutsätter att man har bestämt vilken väg man ska ta. Hela vägen delas upp i n stycken vägavsnitt av olika typer och olika längder. Vägavsnitt j har längden l_j (m) och är av typen t_j .

Baserat på förväntad medelhastighet har man uppskattat hur mycket batterinivån sjunker om man färdas en meter på en väg av kategori t med framdrivningssätt k , och det betecknas med b_{kt} . Dessutom har man beräknat en kostnad (inkluderande allt, dvs. bensin, slitage, miljöpåverkan etc) för att färdas en meter på en väg av kategori t med framdrivningssätt k , och den betecknas med d_{kt} .

Eftersom vägavsnitt j har längden l_j , sjunker batterinivån med $a_{jk} = l_j b_{kt_j}$ om man kör den med inställning k , och kostnaden blir $c_{jk} = l_j d_{kt_j}$.

Man kan använda sig av variablerna $x_{jk} = 1$ om vi kör vägavsnitt j med inställning k , och 0 om inte.

(Om alla koefficienter för en variabel blir noll, kommer den att elimineras.)

Kappsäcksproblem:

$$\begin{aligned} \min \quad & \sum_j \sum_k c_{jk} x_{jk} \\ & \sum_k x_{jk} = 1 \text{ för alla } j \\ & \sum_j \sum_k a_{jk} x_{jk} \leq L_0 \\ & x_{jk} \in \{0, 1\} \text{ för alla } j \text{ och } k \end{aligned}$$

Kappsäcksproblem:

$$\begin{aligned} \min \quad & \sum_j \sum_k c_{jk} x_{jk} \\ & \sum_k x_{jk} = 1 \text{ för alla } j \\ & \sum_j \sum_k a_{jk} x_{jk} \leq L_0 \\ & x_{jk} \in \{0, 1\} \text{ för alla } j \text{ och } k \end{aligned}$$

Notera att antal vägvagnsnummer (j) avgör antalet iterationer, medan högerledet L_0 avgör storleken av tabellen i dynamisk programmering.

TAOP61 Projekt 2: Indata

Data till detta problem består av koefficienterna b_{kt} och d_{kt} . Dessa data återfinnes på filerna *linkbatt.txt* och *linkcost.txt*.

TAOP61 Projekt 2: Indata

Data till detta problem består av koefficienterna b_{kt} och d_{kt} . Dessa data återfinnes på filerna *linkbatt.txt* och *linkcost.txt*.

Dessutom behövs längd l_j och typ t_j för varje vägavsnitt, vilket kan vara många. Dessa data återfinnes på filerna *path0.txt*, *path1.txt* etc. Slutligen behövs startladdningen, L_0 .

TAOP61 Projekt 2: Indata

Data till detta problem består av koefficienterna b_{kt} och d_{kt} . Dessa data återfinnes på filerna *linkbatt.txt* och *linkcost.txt*.

Dessutom behövs längd l_j och typ t_j för varje vägavsnitt, vilket kan vara många. Dessa data återfinnes på filerna *path0.txt*, *path1.txt* etc. Slutligen behövs startladdningen, L_0 .

Programmet läser in dessa data, och beräknar koefficienterna a och c .

TAOP61 Projekt 2: Indata

linkbatt.txt:

12 8 5 0

11 7 4 0

10 6 3 0

10 6 3 0

12 8 5 0

linkcost.txt:

2 4 7 10

1 4 7 10

1 5 8 14

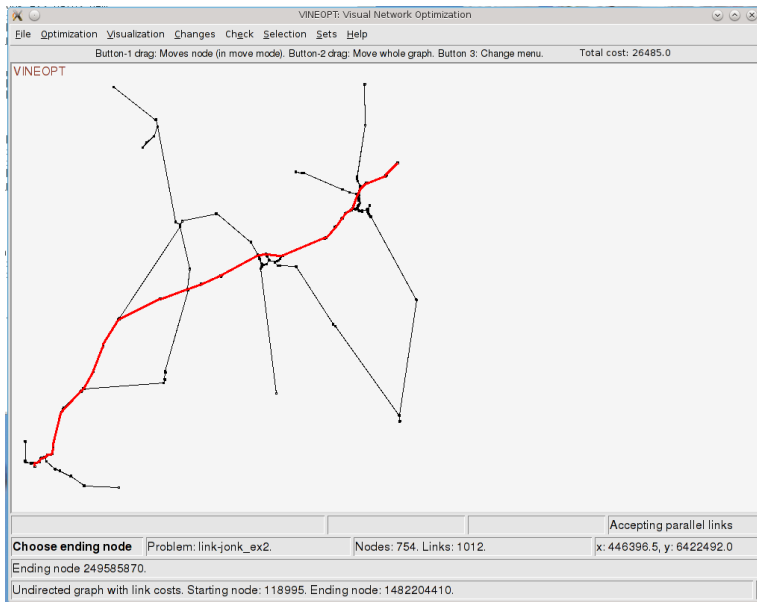
1 6 10 17

1 7 15 20

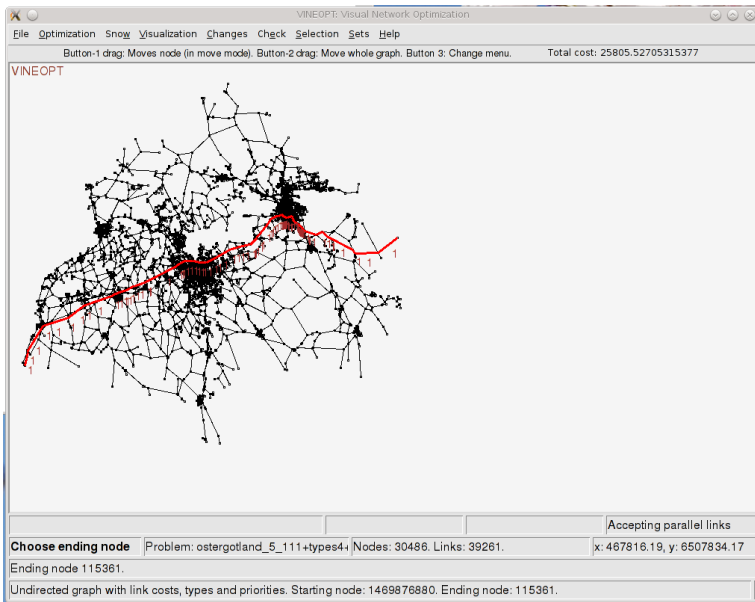
TAOP61 Projekt 2: Instanser

Instans	Antal sträckor	L_0	Vad?
path1	3	10	Litet test
path2	3	4000	Mellantest
path3	5	4000	Större test
path4	20	1000	Från slumpad karta
path6	74	200000	Linköping - Jönköping
path7	74	200000	Linköping - Jönköping
path8	94	100000	Östergötland, öst till väst
path9	246	100000	Östergötland, norr till syd
path12	200	70000	Slumpat
path13	300	100000	Slumpat
path14	500	100000	Slumpat
path15	700	200000	Slumpat
path16	1000	300000	Slumpat

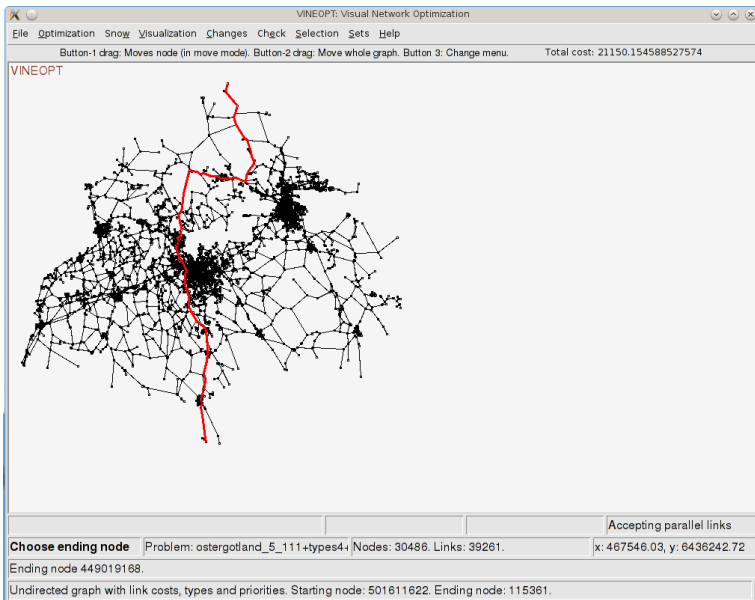
TAOP61 Projekt 2: path6



TAOP61 Projekt 2: path8



TAOP61 Projekt 2: path9



TAOP61 Projekt 2: Kod

På filen *initeldynp.m* finns ett förslag på början av matlab-koden. Den läser in indatafilerna. Programmet kontrollerar om batteriladdningen räcker till att köra på bara el hela vägen. (Dvs. om L_0 är tillräckligt stort.) Då minskas L_0 till precis vad som behövs för eldrift.

TAOP61 Projekt 2: Kod

På filen *initeldynp.m* finns ett förslag på början av matlab-koden. Den läser in indatafilerna. Programmet kontrollerar om batteriladdningen räcker till att köra på bara el hela vägen. (Dvs. om L_0 är tillräckligt stort.) Då minskas L_0 till precis vad som behövs för eldrift.

Efter detta ska ni skriva in er optimeringsalgoritm.

TAOP61 Projekt 2: Kod

På filen *initeldynp.m* finns ett förslag på början av matlab-koden. Den läser in indatafilerna. Programmet kontrollerar om batteriladdningen räcker till att köra på bara el hela vägen. (Dvs. om L_0 är tillräckligt stort.) Då minskas L_0 till precis vad som behövs för eldrift.

Efter detta ska ni skriva in er optimeringsalgoritm.

Några tips (matlab-kommandon som kan vara bra att ha):

```
ix=ix-a;
```

Minskar indexvektorn *ix* med *a* enheter.

```
nix=1-ix>0;
```

nix blir 0 i de positioner där *ix* är större än noll och 1 i de positioner där *ix* mindre eller lika med noll.

```
ft0=circshift(ft1,[0 a]);
```

Vektorn *ft0* är vektorn *ft1* skiftad (flyttad) *a* steg åt höger.

```
ft3(nix)=inf;
```

Elementen med index där *nix*=1 i vektorn *ft3* sätts till ett stort tal. (De kommer därför aldrig att ge minimum.)

TAOP61 Projekt 2

- 1 Gör nödvändiga definitioner och formuleringar för att lösa problemet med dynamisk programmering.
- 2 Gör ett program som löser problemet med dynamisk programmering.
- 3 Lös det lilla testproblemet för att se att allt fungerar korrekt.
- 4 Lös de större problemen. Notera målfunktionsvärde och lösning.
- 5 Lös problem *path7* för olika startladdning.