

# TAOP61 Projekt 4

# TAOP61 Projekt 4

Lösning av snöröjningsproblemet.

# TAOP61 Projekt 4

Lösning av snöröjningsproblemet.

Programvara: Vineopt. Snowplan.

# TAOP61 Projekt 4

Lösning av snöröjningsproblemet.

Programvara: Vineopt. Snowplan.

Material: Problembeskrivning: Snöröjning.

# TAOP61 Projekt 4

Lösning av snöröjningsproblemet.

Programvara: Vineopt. Snowplan.

Material: Problembeskrivning: Snöröjning.

Uppgifter:

# TAOP61 Projekt 4

Lösning av snöröjningsproblemet.

Programvara: Vineopt. Snowplan.

Material: Problembeskrivning: Snöröjning.

Uppgifter:

Ni ska använda Vineopt för att lösa kinesiska brevbärarproblem och lantbrevbärarproblem. Uppdelningen av uppgifter mellan fordon ska dels göras för hand, och dels med heuristik i Snowplan.

# TAOP61 Projekt 4

Lösning av snöröjningsproblemet.

Programvara: Vineopt. Snowplan.

Material: Problembeskrivning: Snöröjning.

Uppgifter:

Ni ska använda Vineopt för att lösa kinesiska brevbärarproblem och lantbrevbärarproblem. Uppdelningen av uppgifter mellan fordon ska dels göras för hand, och dels med heuristik i Snowplan.

Projektet redovisas med en skriftlig rapport.

## TAOP61 Projekt 4: Snöröjning i städer

Ni är inhyrda av en svensk kommun för att planera snöröjningen. En konsultfirma har levererat ett program, Snowplan, som påstås göra en bra planering. Er uppgift är att utvärdera snowplan, samt att ta fram bra planer för snöröjningen. Snowplan och Vineopt kan konkurrera och samarbeta, genom att låta ett program förbättra lösningen från det andra, i båda riktningarna.



## TAOP61 Projekt 4: Snöröjning i städer

Ni är inhyrda av en svensk kommun för att planera snöröjningen. En konsultfirma har levererat ett program, Snowplan, som påstås göra en bra planering. Er uppgift är att utvärdera snowplan, samt att ta fram bra planer för snöröjningen. Snowplan och Vineopt kan konkurrera och samarbeta, genom att låta ett program förbättra lösningen från det andra, i båda riktningarna.

Vad kommunen är intresserad av är följande:

## TAOP61 Projekt 4: Snöröjning i städer

Ni är inhyrda av en svensk kommun för att planera snöröjningen. En konsultfirma har levererat ett program, Snowplan, som påstås göra en bra planering. Er uppgift är att utvärdera snowplan, samt att ta fram bra planer för snöröjningen. Snowplan och Vineopt kan konkurrera och samarbeta, genom att låta ett program förbättra lösningen från det andra, i båda riktningarna.

Vad kommunen är intresserad av är följande:

- Bra planer för snöröjningen samt jämförelser för olika vikter på tid.

## TAOP61 Projekt 4: Snöröjning i städer

Ni är inhyrda av en svensk kommun för att planera snöröjningen. En konsultfirma har levererat ett program, Snowplan, som påstås göra en bra planering. Er uppgift är att utvärdera snowplan, samt att ta fram bra planer för snöröjningen. Snowplan och Vineopt kan konkurrera och samarbeta, genom att låta ett program förbättra lösningen från det andra, i båda riktningarna.

Vad kommunen är intresserad av är följande:

- Bra planer för snöröjningen samt jämförelser för olika vikter på tid.
- En jämförelse mellan snowplan och vad man kan få fram med Vineopt.

## TAOP61 Projekt 4: Snöröjning i städer

Ni är inhyrda av en svensk kommun för att planera snöröjningen. En konsultfirma har levererat ett program, Snowplan, som påstås göra en bra planering. Er uppgift är att utvärdera snowplan, samt att ta fram bra planer för snöröjningen. Snowplan och Vineopt kan konkurrera och samarbeta, genom att låta ett program förbättra lösningen från det andra, i båda riktningarna.

Vad kommunen är intresserad av är följande:

- Bra planer för snöröjningen samt jämförelser för olika vikter på tid.
- En jämförelse mellan snowplan och vad man kan få fram med Vineopt.
- Diskussion om brister/potential i snowplan.

## TAOP61 Projekt 4: Snöröjning i städer

Ni är inhyrda av en svensk kommun för att planera snöröjningen. En konsultfirma har levererat ett program, Snowplan, som påstås göra en bra planering. Er uppgift är att utvärdera snowplan, samt att ta fram bra planer för snöröjningen. Snowplan och Vineopt kan konkurrera och samarbeta, genom att låta ett program förbättra lösningen från det andra, i båda riktningarna.

Vad kommunen är intresserad av är följande:

- Bra planer för snöröjningen samt jämförelser för olika vikter på tid.
- En jämförelse mellan snowplan och vad man kan få fram med Vineopt.
- Diskussion om brister/potential i snowplan.
- Diskussion om möjligheterna att kombinera Vineopt och snowplan.

## TAOP61 Projekt 4: Snöröjning i städer

Ni är inhyrda av en svensk kommun för att planera snöröjningen. En konsultfirma har levererat ett program, Snowplan, som påstås göra en bra planering. Er uppgift är att utvärdera snowplan, samt att ta fram bra planer för snöröjningen. Snowplan och Vineopt kan konkurrera och samarbeta, genom att låta ett program förbättra lösningen från det andra, i båda riktningarna.

Vad kommunen är intresserad av är följande:

- Bra planer för snöröjningen samt jämförelser för olika vikter på tid.
- En jämförelse mellan snowplan och vad man kan få fram med Vineopt.
- Diskussion om brister/potential i snowplan.
- Diskussion om möjligheterna att kombinera Vineopt och snowplan.
- Ett anbud på snöröjningen i Vadstena.

## TAOP61 Projekt 4: Snöröjning i städer

I detta projekt behandlas ett förenklat problem, där man inte beaktar att det tar längre tid att vända än att köra rakt fram. Dessutom beaktas bara normala gator, ej gångvägar eller cykelvägar. Till sist bortser vi från detaljer avseende röjning av vändplatser och korsningar.

## TAOP61 Projekt 4: Snöröjning i städer

I detta projekt behandlas ett förenklat problem, där man inte beaktar att det tar längre tid att vända än att köra rakt fram. Dessutom beaktas bara normala gator, ej gångvägar eller cykelvägar. Till sist bortser vi från detaljer avseende röjning av vändplatser och korsningar.

Vi räknar med en medelhastighet av 7.2 km/h. Tiden det tar att röja en gata uppskattas till 2 gånger längden dividerat med hastigheten. Denna tid inkluderar två eller tre drag (mitten, höger sida, vänster sida) samt avslutande röjning vid vändplats och/eller korsning. Allt detta ses alltså som en operation, som startar i ena ändan av gatan och slutar i andra.



## TAOP61 Projekt 4: Snöröjning i städer

I detta projekt behandlas ett förenklat problem, där man inte beaktar att det tar längre tid att vända än att köra rakt fram. Dessutom beaktas bara normala gator, ej gångvägar eller cykelvägar. Till sist bortser vi från detaljer avseende röjning av vändplatser och korsningar.

Vi räknar med en medelhastighet av 7.2 km/h. Tiden det tar att röja en gata uppskattas till 2 gånger längden dividerat med hastigheten. Denna tid inkluderar två eller tre drag (mitten, höger sida, vänster sida) samt avslutande röjning vid vändplats och/eller korsning. Allt detta ses alltså som en operation, som startar i ena ändan av gatan och slutar i andra.

Detta leder till det tar en sekund per meter, och att problemet är ett kinesiskt brevbärarproblem, om man bara har ett fordon som ska göra allt.

## TAOP61 Projekt 4: Snöröjning i städer

I detta projekt behandlas ett förenklat problem, där man inte beaktar att det tar längre tid att vända än att köra rakt fram. Dessutom beaktas bara normala gator, ej gångvägar eller cykelvägar. Till sist bortser vi från detaljer avseende röjning av vändplatser och korsningar.

Vi räknar med en medelhastighet av 7.2 km/h. Tiden det tar att röja en gata uppskattas till 2 gånger längden dividerat med hastigheten. Denna tid inkluderar två eller tre drag (mitten, höger sida, vänster sida) samt avslutande röjning vid vändplats och/eller korsning. Allt detta ses alltså som en operation, som startar i ena ändan av gatan och slutar i andra.

Detta leder till det tar en sekund per meter, och att problemet är ett kinesiskt brevbärarproblem, om man bara har ett fordon som ska göra allt. Körning utan röjning går snabbare i Snowplan. I Vineopt måste en parameter sättas, eller en uträkning göras.

## TAOP61 Projekt 4: Snöröjning i städer

I detta projekt behandlas ett förenklat problem, där man inte beaktar att det tar längre tid att vända än att köra rakt fram. Dessutom beaktas bara normala gator, ej gångvägar eller cykelvägar. Till sist bortser vi från detaljer avseende röjning av vändplatser och korsningar.

Vi räknar med en medelhastighet av 7.2 km/h. Tiden det tar att röja en gata uppskattas till 2 gånger längden dividerat med hastigheten. Denna tid inkluderar två eller tre drag (mitten, höger sida, vänster sida) samt avslutande röjning vid vändplats och/eller korsning. Allt detta ses alltså som en operation, som startar i ena ändan av gatan och slutar i andra.

Detta leder till det tar en sekund per meter, och att problemet är ett kinesiskt brevbärarproblem, om man bara har ett fordon som ska göra allt.

Körning utan röjning går snabbare i Snowplan. I Vineopt måste en parameter sättas, eller en uträkning göras.

Uppgiften handlar om att jämföra resultatet av att använda olika antal fordon. Självklart går det snabbare att låta flera fordon arbeta parallellt, men varje fordon ger en fast kostnad.

## TAOP61 Projekt 4: Snöröjning i städer

I Vineopt kan man definiera en mängd av länkar för varje fordon ("linksets") (genom att välja länkar direkt eller noder och sedan länkarna mellan valda noder), med de olika verktygen i Vineopt. Sedan gör man en länkmängd i taget nödvändig ("required") för ett lantbrevbärrproblem och löser det.

## TAOP61 Projekt 4: Snöröjning i städer

I Vineopt kan man definiera en mängd av länkar för varje fordon ("linksets") (genom att välja länkar direkt eller noder och sedan länkarna mellan valda noder), med de olika verktygen i Vineopt. Sedan gör man en länkmängd i taget nödvändig ("required") för ett lantbrevbärrproblem och löser det.

Totalkostnaden för detta blir summan av kostnaderna för turerna för varje fordon, plus en fast kostnad för varje fordon. Vi antar att den fasta kostnaden för varje fordon är 200, och att varje sekund kostar 1.

Totalkostnaden är alltså summan av kostnaderna från Vineopt för varje fordon plus antal fordon gånger 200.

## TAOP61 Projekt 4: Snöröjning i städer

I Vineopt kan man definiera en mängd av länkar för varje fordon ("linksets") (genom att välja länkar direkt eller noder och sedan länkarna mellan valda noder), med de olika verktygen i Vineopt. Sedan gör man en länkmängd i taget nödvändig ("required") för ett lantbrevbärrarproblem och löser det.

Totalkostnaden för detta blir summan av kostnaderna för turerna för varje fordon, plus en fast kostnad för varje fordon. Vi antar att den fasta kostnaden för varje fordon är 200, och att varje sekund kostar 1.

Totalkostnaden är alltså summan av kostnaderna från Vineopt för varje fordon plus antal fordon gånger 200.

Tiden för lösningen är lika med den maximala tiden (kostnaden) för något fordon. Vilken lösning som är bäst beror då på hur viktigt man tycker att tiden är jämfört med kostnaden. Många fordon ger hög kostnad men kort tid, medan få fordon ger lägre kostnad men längre tid.

## TAOP61 Projekt 4: Snöröjning i städer

I Vineopt kan man definiera en mängd av länkar för varje fordon ("linksets") (genom att välja länkar direkt eller noder och sedan länkarna mellan valda noder), med de olika verktygen i Vineopt. Sedan gör man en länkmängd i taget nödvändig ("required") för ett lantbrevbärrarproblem och löser det.

Totalkostnaden för detta blir summan av kostnaderna för turerna för varje fordon, plus en fast kostnad för varje fordon. Vi antar att den fasta kostnaden för varje fordon är 200, och att varje sekund kostar 1.

Totalkostnaden är alltså summan av kostnaderna från Vineopt för varje fordon plus antal fordon gånger 200.

Tiden för lösningen är lika med den maximala tiden (kostnaden) för något fordon. Vilken lösning som är bäst beror då på hur viktigt man tycker att tiden är jämfört med kostnaden. Många fordon ger hög kostnad men kort tid, medan få fordon ger lägre kostnad men längre tid.

Vi beaktar två scenarior, ett där kostnad och tid är värda lika mycket, dvs. vårt mått är totalkostnaden plus tiden, och ett där tiden är dubbelt så viktig som kostnaden, dvs. måttet är totalkostnaden plus två gånger tiden.

## TAOP61 Projekt 4: Indata

Gatunätverket finns tillgängligt i Vineopt-format (typ lantbrevbärrproblem, som oriktad graf med bågkostnader och "required links"). Bågstörnaderna är lika med vägsträckornas längd i meter.



## TAOP61 Projekt 4: Indata

Gatunätverket finns tillgängligt i Vineopt-format (typ lantbrevbärrarproblem, som oriktad graf med bågstnader och “required links”). Bågstnaderna är lika med vägsträckornas längd i meter.

Vineopt kan läsa och skriva bågstmängder på fil, samt visa dem i grafen.

## TAOP61 Projekt 4: Indata

Gatunätverket finns tillgängligt i Vineopt-format (typ lantbrevbärrarproblem, som oriktad graf med bågkostnader och “required links”). Bågkostnaderna är lika med vägsträckornas längd i meter.

Vineopt kan läsa och skriva bågmängder på fil, samt visa dem i grafen.

Det finns ett litet nätverk, *colonia*, (studentbostäder vid märkesbacken) med 24 noder och 36 länkar,

## TAOP61 Projekt 4: Indata

Gatunätverket finns tillgängligt i Vineopt-format (typ lantbrevbärrproblem, som oriktad graf med bågkostnader och "required links"). Bågstörnaderna är lika med vägsträckornas längd i meter.

Vineopt kan läsa och skriva bågmängder på fil, samt visa dem i grafen.

Det finns ett litet nätverk, *colonia*, (studentbostäder vid märkesbacken) med 24 noder och 36 länkar,

två lite större, *skanninge-n* (Skänninge norra), med 62 noder och 80 länkar, och *atvid-s* (Åtvidaberg södra), med 68 noder och 95 länkar,

## TAOP61 Projekt 4: Indata

Gatunätverket finns tillgängligt i Vineopt-format (typ lantbrevbärrarproblem, som oriktad graf med bågkostnader och “required links”). Bågstörnaderna är lika med vägsträckornas längd i meter.

Vineopt kan läsa och skriva bågmängder på fil, samt visa dem i grafen.

Det finns ett litet nätverk, *colonia*, (studentbostäder vid märkesbacken) med 24 noder och 36 länkar,

två lite större, *skanninge-n* (Skänninge norra), med 62 noder och 80 länkar, och *atvid-s* (Åtvidaberg södra), med 68 noder och 95 länkar,

samt två större, *studentryd*, med 387 noder och 519 länkar, samt *vadstena* (liten stad med stort slott vid stor sjö), med 581 noder och 778 länkar.

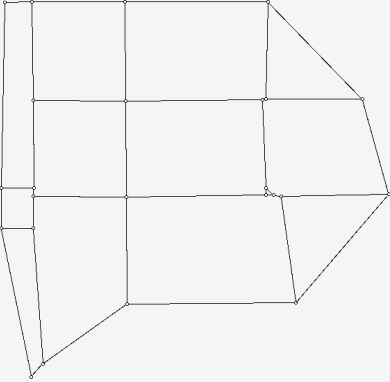
# TAOP61 Projekt 4: colonia

VINEOPT: Visual Network Optimization

File Optimization GPS Visualization Changes Check Selection Sets Help

Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu.

VINEOPT



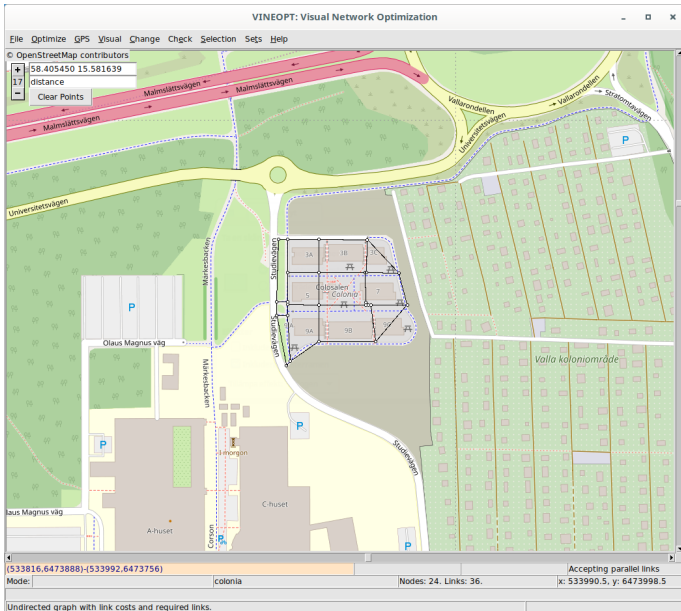
Accepting parallel links

<b>Move node</b>	Problem: colonia.	Nodes: 24. Links: 36.	x: 533930.25, y: 6473756.66
------------------	-------------------	-----------------------	-----------------------------

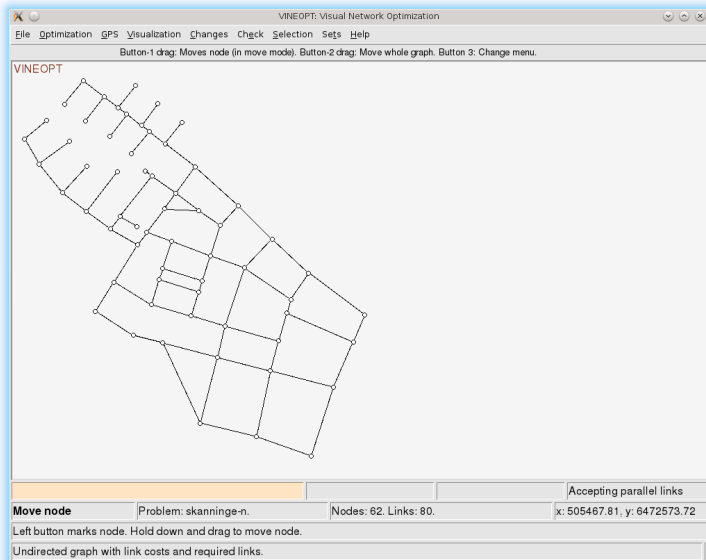
Left button marks node. Hold down and drag to move node.

Undirected graph with link costs and required links.

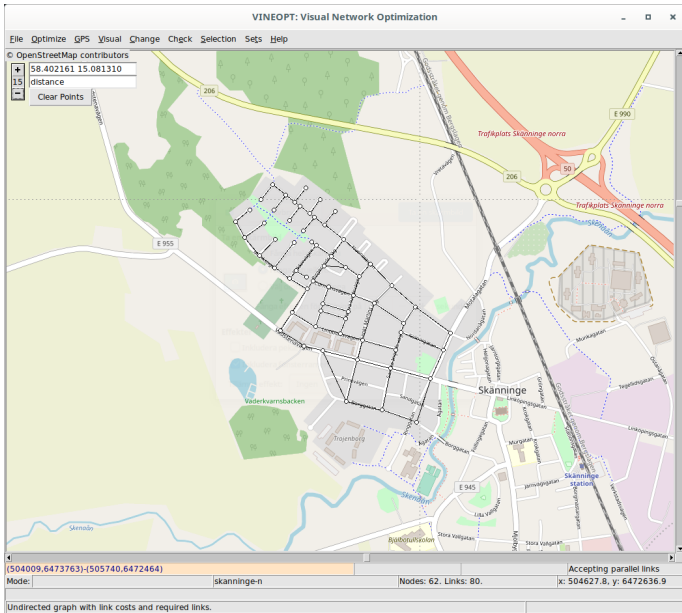
# TAOP61 Projekt 4: colonia



# TAOP61 Projekt 4: skanninge-n (Skänninge Norra)

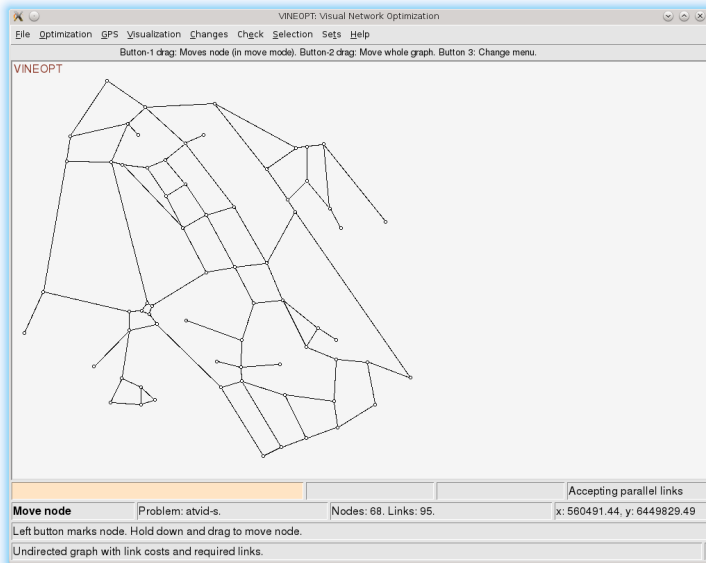


# TAOP61 Projekt 4: skanninge-n (Skänninge Norra)

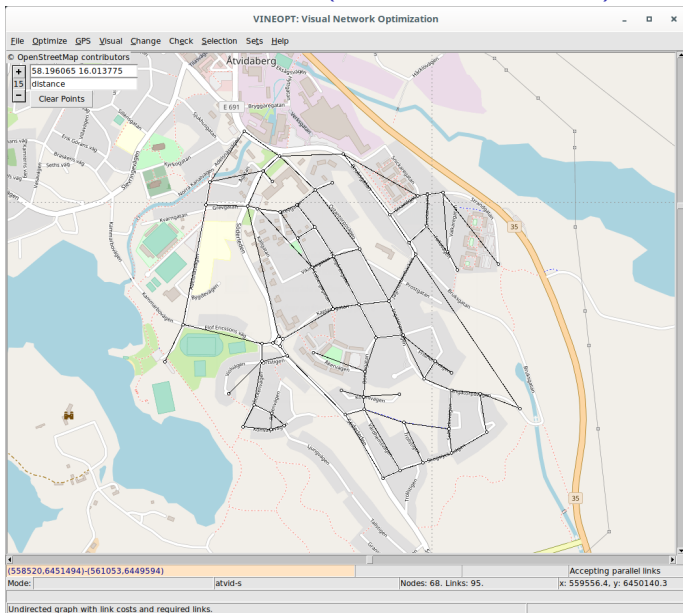




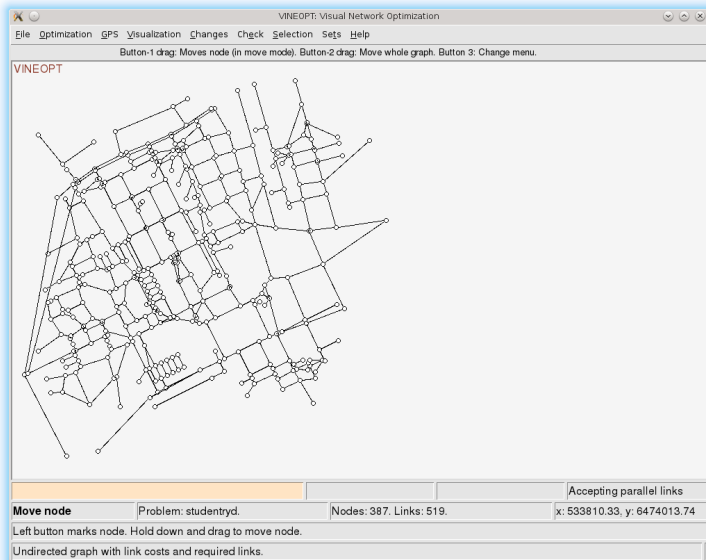
# TAOP61 Projekt 4: atvid-s (Åtvidaberg Södra)



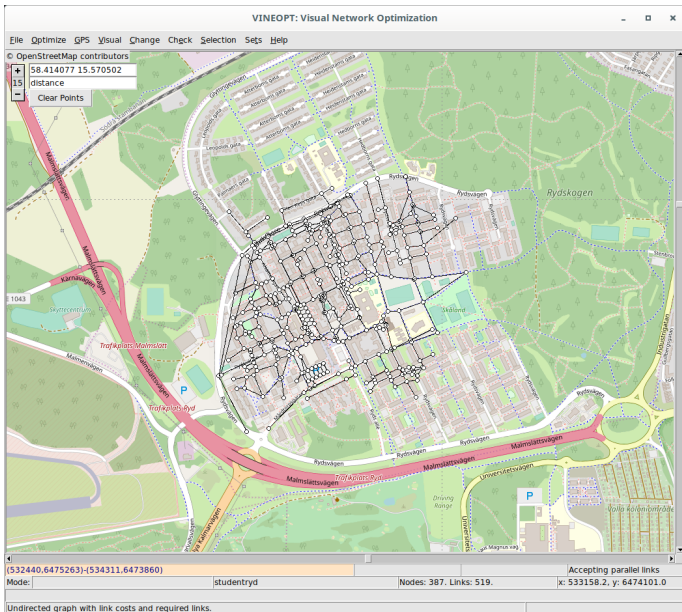
# TAOP61 Projekt 4: atvid-s (Åtvidaberg Södra)



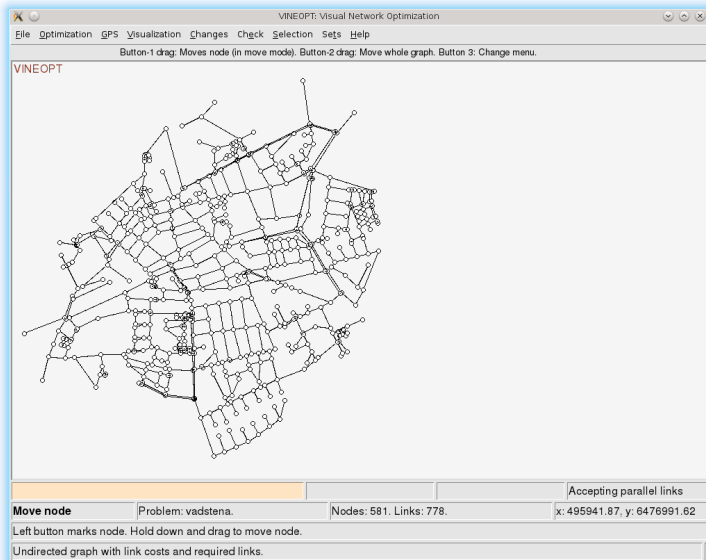
# TAOP61 Projekt 4: studentryd



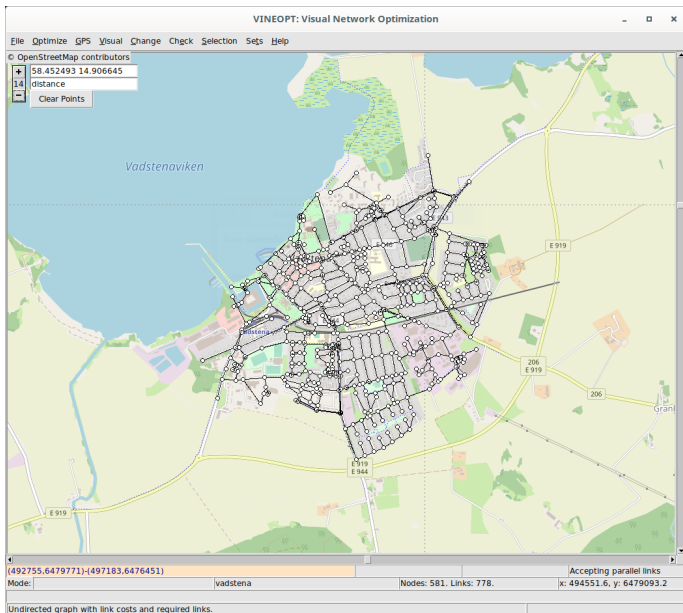
# TAOP61 Projekt 4: studentryd



# TAOP61 Projekt 4: vadstena



# TAOP61 Projekt 4: vadstena



## TAOP61 Projekt 4: Snowplan

Snowplan 3 är ett Pythonprogram som hjälper till vid lösning av ett “ $k$ -brevbärarproblem”, ett brevbärarproblem där alla bågar ska täckas (minst en gång), men inte av *en* brevbärare utan av flera. En “brevbärare” motsvarar här ett snöröjningsfordon.

## TAOP61 Projekt 4: Snowplan

Snowplan 3 är ett Pythonprogram som hjälper till vid lösning av ett “ $k$ -brevbärarproblem”, ett brevbärarproblem där alla bågar ska täckas (minst en gång), men inte av *en* brevbärare utan av flera. En “brevbärare” motsvarar här ett snöröjningsfordon.

Snowplan arbetar med ett subproblem, som består av ett lantbrevbärarproblem (rural postman problem) för varje fordon. Detta problem löses med en kod från Vineopt, *rupov*. Huvudfunktionen hos Snowplan är att finna en bra uppdelning av bågarna mellan de olika fordonen.



## TAOP61 Projekt 4: Snowplan

Snowplan 3 är ett Pythonprogram som hjälper till vid lösning av ett “ $k$ -brevbärarproblem”, ett brevbärarproblem där alla bågar ska täckas (minst en gång), men inte av *en* brevbärare utan av flera. En “brevbärare” motsvarar här ett snöröjningsfordon.

Snowplan arbetar med ett subproblem, som består av ett lantbrevbärarproblem (rural postman problem) för varje fordon. Detta problem löses med en kod från Vineopt, *rupov*. Huvudfunktionen hos Snowplan är att finna en bra uppdelning av bågarna mellan de olika fordonen.

Vektorer: `order`, som är ordningen i vilken fordonen behandlas,

## TAOP61 Projekt 4: Snowplan

Snowplan 3 är ett Pythonprogram som hjälper till vid lösning av ett “ $k$ -brevbärarproblem”, ett brevbärarproblem där alla bågar ska täckas (minst en gång), men inte av *en* brevbärare utan av flera. En “brevbärare” motsvarar här ett snöröjningsfordon.

Snowplan arbetar med ett subproblem, som består av ett lantbrevbärarproblem (rural postman problem) för varje fordon. Detta problem löses med en kod från Vineopt, *rupov*. Huvudfunktionen hos Snowplan är att finna en bra uppdelning av bågarna mellan de olika fordonen.

Vektorer: `order`, som är ordningen i vilken fordonen behandlas, och `vehdo`, som för varje båge anger vilket fordon som ska röja bågen.

## TAOP61 Projekt 4: Snowplan

Snowplan 3 är ett Pythonprogram som hjälper till vid lösning av ett “ $k$ -brevbärarproblem”, ett brevbärarproblem där alla bågar ska täckas (minst en gång), men inte av *en* brevbärare utan av flera. En “brevbärare” motsvarar här ett snöröjningsfordon.

Snowplan arbetar med ett subproblem, som består av ett lantbrevbärarproblem (rural postman problem) för varje fordon. Detta problem löses med en kod från Vineopt, *rupov*. Huvudfunktionen hos Snowplan är att finna en bra uppdelning av bågarna mellan de olika fordonen.

Vektorer: `order`, som är ordningen i vilken fordonen behandlas, och `vehdo`, som för varje båge anger vilket fordon som ska röja bågen. För att acceptera förflyttning till en ny punkt används simulerad kylning, dvs. bättre punkter accepteras alltid och sämre punkter accepteras med en viss sannolikhet.

## TAOP61 Projekt 4: Snowplan

Snowplan 3 är ett Pythonprogram som hjälper till vid lösning av ett “ $k$ -brevbärarproblem”, ett brevbärarproblem där alla bågar ska täckas (minst en gång), men inte av *en* brevbärare utan av flera. En “brevbärare” motsvarar här ett snöröjningsfordon.

Snowplan arbetar med ett subproblem, som består av ett lantbrevbärarproblem (rural postman problem) för varje fordon. Detta problem löses med en kod från Vineopt, *rupov*. Huvudfunktionen hos Snowplan är att finna en bra uppdelning av bågarna mellan de olika fordonen.

Vektorer: `order`, som är ordningen i vilken fordonen behandlas, och `vehdo`, som för varje båge anger vilket fordon som ska röja bågen. För att acceptera förflyttning till en ny punkt används simulerad kylning, dvs. bättre punkter accepteras alltid och sämre punkter accepteras med en viss sannolikhet.

Startallokering och ändring av `order` och `vehdo` kan göras på flera olika sätt.

## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat.  
Viktiga frågor är bl.a.

## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat.  
Viktiga frågor är bl.a.

1. Hur ska startlösningen konstrueras?

## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat. Viktiga frågor är bl.a.

1. Hur ska startlösningen konstrueras?
2. Hur ska ordning och allokering ändras mellan iterationerna?

## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat. Viktiga frågor är bl.a.

1. Hur ska startlösningen konstrueras?
2. Hur ska ordning och allokering ändras mellan iterationerna?
3. Kylningsparametrarna: Hur snabbt ska temperaturen sänkas?



## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat. Viktiga frågor är bl.a.

1. Hur ska startlösningen konstrueras?
2. Hur ska ordning och allokering ändras mellan iterationerna?
3. Kylningsparametrarna: Hur snabbt ska temperaturen sänkas?
4. Vilken typ av förbättringar av lösningen ska man göra?

## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat. Viktiga frågor är bl.a.

1. Hur ska startlösningen konstrueras?
2. Hur ska ordning och allokering ändras mellan iterationerna?
3. Kylningsparametrarna: Hur snabbt ska temperaturen sänkas?
4. Vilken typ av förbättringar av lösningen ska man göra?

Testa med en dålig startlösning för se hur snabbt lösningarna förbättras, och testa med en bra startlösning för att se om den överhuvudtaget kan förbättras.

## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat. Viktiga frågor är bl.a.

1. Hur ska startlösningen konstrueras?
2. Hur ska ordning och allokering ändras mellan iterationerna?
3. Kylningsparametrarna: Hur snabbt ska temperaturen sänkas?
4. Vilken typ av förbättringar av lösningen ska man göra?

Testa med en dålig startlösning för se hur snabbt lösningarna förbättras, och testa med en bra startlösning för att se om den överhuvudtaget kan förbättras.

Gör inte för få iterationer.

## TAOP61 Projekt 4: Hjälpmedel

Snowplan styrs av parametrar som kan justeras för att ge bra resultat. Viktiga frågor är bl.a.

1. Hur ska startlösningen konstrueras?
2. Hur ska ordning och allokering ändras mellan iterationerna?
3. Kylningsparametrarna: Hur snabbt ska temperaturen sänkas?
4. Vilken typ av förbättringar av lösningen ska man göra?

Testa med en dålig startlösning för se hur snabbt lösningarna förbättras, och testa med en bra startlösning för att se om den överhuvudtaget kan förbättras.

Gör inte för få iterationer.

Obs: En metod med slump ger olika resultat vid upprepade körningar.

## TAOP61 Projekt 4: Snowplan

- 1 Nätverket läses in.
- 2 En startordning konstrueras.
- 3 En startallokering av bågar till fordon görs.
- 4 Lantbrevbärrarproblemet löses för varje fordon.
- 5 Valfritt: De oröjda bågar ett fordon passerar röjs av fordonet, oavsett om bågen var allokerad till det fordonet.
- 6 Valfritt: Lösningen förbättras genom att flytta cykler mellan turer.
- 7 Ordningen för fordonen ändras.
- 8 Allokering av bågar till fordon ändras.
- 9 Upprepa steg 4 - 9 ett visst antal gånger.
- 10 Allokeringen av fordon till bågar skrivs ut (som bågmängder), i ett format som Vineopt kan läsa in.

## TAOP61 Projekt 4: Snowplan: Parametrar

nov	Antal fordon, $\geq 1$ (4)
fixvehcost	Fast kostnad för ett fordon, $\geq 0$ (200)
wcost	Vikt på kostnad i målfunktionen, $\geq 0$ (1.0)
wtime	Vikt på tid i målfunktionen, $\geq 0$ (1.0)
maxiter	Antal huvuditerationer, $\geq 1$ (30)
markdone	Markera passerade bågar som gjorda, 0,1 (1)
startvar	Variant för startallokering, 0 – 16, (15)
movevar	Variant för ändring av allokering, 0 – 7 (3)
impvar	Variant för förbättringsheuristiken, 0 – 3 (3)
dorevsol	Ändra allokeringen efter lösningen, 0,1 (1)
chordvar	Variant för ändring av ordning, 1 – 5 (5)
cymvar	Variant för flyttning av cykler, 1 – 4 (4)
cymimp	Flytta cykler bara om det ger förbättring, 0,1 (0)
starttemp	Starttemperatur i simulerad kylning, $\geq 0$ (1000)
intiter	Antal iter med samma temp i simulerad kylning, $\geq 1$ (7)
rcold	Kylningsfaktor i simulerad kylning, $> 0, < 1$ (0.3)

## TAOP61 Projekt 4: Snowplan

Programmet startas genom att skriva: `dine 6`  
vilket ger den interaktiva sidan med möjligheter att läsa in nätverk och parametrar, samt ändra parametrar och lösa problemet.

## TAOP61 Projekt 4: Snowplan

Programmet startas genom att skriva: `dine 6` vilket ger den interaktiva sidan med möjligheter att läsa in nätverk och parametrar, samt ändra parametrar och lösa problemet.

```
dine 6 network1 param1.par
```

läser in nätverket *network1* och parameterfilen *param1.par*.



## TAOP61 Projekt 4: Snowplan

Programmet startas genom att skriva: `dine 6` vilket ger den interaktiva sidan med möjligheter att läsa in nätverk och parametrar, samt ändra parametrar och lösa problemet.

```
dine 6 network1 param1.par
```

läser in nätverket *network1* och parameterfilen *param1.par*.

```
dine 6 network1 param1.par 1
```

löser problemet direkt. Blå linje: ursprungligt målfunktionsvärde, gröna kryss: efter förbättringsfasen, röd streckad: bästa värdet efter varje iteration. Ljusblå linje: undre gränsen.

## TAOP61 Projekt 4: Snowplan

Programmet startas genom att skriva: `dine 6` vilket ger den interaktiva sidan med möjligheter att läsa in nätverk och parametrar, samt ändra parametrar och lösa problemet.

```
dine 6 network1 param1.par
```

läser in nätverket *network1* och parameterfilen *param1.par*.

```
dine 6 network1 param1.par 1
```

löser problemet direkt. Blå linje: ursprungligt målfunktionsvärde, gröna kryss: efter förbättringsfasen, röd streckad: bästa värdet efter varje iteration. Ljusblå linje: undre gränsen.

```
dine 6 network1 param1.par 2
```

löser problemet och avslutar programmet. (Se terminalen.)

## TAOP61 Projekt 4: Snowplan

Filen *spar-def.par* innehåller de föreslagna värdena på parametrarna.

## TAOP61 Projekt 4: Snowplan

Filen *spar-def.par* innehåller de föreslagna värdena på parametrarna.

När ett problem löses adderas en rad till filen *fkres.txt*, med nätverksnamn, inparameterfil, datum, problemstorlek samt målfunktionsvärde och lösningstid.

## TAOP61 Projekt 4: Snowplan

Filen *spar-def.par* innehåller de föreslagna värdena på parametrarna.

När ett problem löses adderas en rad till filen *fkres.txt*, med nätverksnamn, inparameterfil, datum, problemstorlek samt målfunktionsvärde och lösningstid.

En liknande fil, *fpkres.txt*, skapas på samma sätt, men innehåller (kompakt) information om alla parametrar.

## TAOP61 Projekt 4: Snowplan

Filen *spar-def.par* innehåller de föreslagna värdena på parametrarna.

När ett problem löses adderas en rad till filen *fkres.txt*, med nätverksnamn, inparameterfil, datum, problemstorlek samt målfunktionsvärde och lösningstid.

En liknande fil, *fpkres.txt*, skapas på samma sätt, men innehåller (kompakt) information om alla parametrar.

Dessutom skapas flera resultatfiler i datakatalogen ( $\#$  = nummer):

*network1-param1\_#.png*: Bild av bästa allokeringen i png-format.

*network1-param1\_#.ps*: Bild av bästa allokeringen i postscript-format.

*network1-param1\_#.val*: Undre gräns(er) samt för varje iteration målfunktionsvärde före och efter förbättring, samt det bästa värdet dittills.

## TAOP61 Projekt 4: Snowplan

Filen *spar-def.par* innehåller de föreslagna värdena på parametrarna.

När ett problem löses adderas en rad till filen *fkres.txt*, med nätverksnamn, inparameterfil, datum, problemstorlek samt målfunktionsvärde och lösningstid.

En liknande fil, *fpkres.txt*, skapas på samma sätt, men innehåller (kompakt) information om alla parametrar.

Dessutom skapas flera resultatfiler i datakatalogen ( $\#$  = nummer):

*network1-param1\_#.png*: Bild av bästa allokeringen i png-format.

*network1-param1\_#.ps*: Bild av bästa allokeringen i postscript-format.

*network1-param1\_#.val*: Undre gräns(er) samt för varje iteration målfunktionsvärde före och efter förbättring, samt det bästa värdet dittills.

Efter varje körning sparas bågmängderna på filen *linksets#.set*, där  $\#$  står för ett nummer som räknas upp varje gång.

## TAOP61 Projekt 4: Snowplan

Filen *spar-def.par* innehåller de föreslagna värdena på parametrarna.

När ett problem löses adderas en rad till filen *fkres.txt*, med nätverksnamn, inparameterfil, datum, problemstorlek samt målfunktionsvärde och lösningstid.

En liknande fil, *fpkres.txt*, skapas på samma sätt, men innehåller (kompakt) information om alla parametrar.

Dessutom skapas flera resultatfiler i datakatalogen ( $\#$  = nummer):

*network1-param1\_#.png*: Bild av bästa allokeringen i png-format.

*network1-param1\_#.ps*: Bild av bästa allokeringen i postscript-format.

*network1-param1\_#.val*: Undre gräns(er) samt för varje iteration målfunktionsvärde före och efter förbättring, samt det bästa värdet dittills.

Efter varje körning sparas bågmängderna på filen *linksets#.set*, där  $\#$  står för ett nummer som räknas upp varje gång.

(Rensa gärna bland dessa filer ibland.)



# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan										
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit	
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset
Nonexisting indata file										
Parameter file: spar-def.par										
No data read yet	Number of vehicles: 4		Maxiter: 30		Weight of cost: 1.0					
Time for vehicle 1:	Start variant: 15		Fixed cost for vehicle: 200.0		Weight of time: 1.0					
Time for vehicle 2:	Move variant: 3		Target cost: 0.0		<input checked="" type="checkbox"/> Mark done? <input checked="" type="checkbox"/> Revise sol? <input checked="" type="checkbox"/> Move cycles only if improve? <input type="checkbox"/> Try all cycle swaps? <input type="checkbox"/> Print more?					
Time for vehicle 3:	Change order variant: 5		Target time: 0.0							
Time for vehicle 4:	Improvement variant: 3		SA: Start temp: 1000.0							
		Cycle moving variant: 4		SA: Internal iter: 7						
				SA: Cooling factor: 0.3						

# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan										
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit	
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset
Nonexisting indata file										
Parameter file: spar-def.par										
No data read yet	Number of vehicles: 4		Maxiter: 30		Weight of cost: 1.0					
Time for vehicle 1:	Start variant: 15		Fixed cost for vehicle: 200.0		Weight of time: 1.0					
Time for vehicle 2:	Move variant: 3		Target cost: 0.0		<input checked="" type="checkbox"/> Mark done? <input checked="" type="checkbox"/> Revise sol? <input checked="" type="checkbox"/> Move cycles only if improve? <input type="checkbox"/> Try all cycle swaps? <input type="checkbox"/> Print more?					
Time for vehicle 3:	Change order variant: 5		Target time: 0.0							
Time for vehicle 4:	Improvement variant: 3		SA: Start temp: 1000.0							
		Cycle moving variant: 4		SA: Internal iter: 7						
				SA: Cooling factor: 0.3						

“Update”: spara inställningarna.

# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan										
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit	
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset
Nonexisting indata file										
Parameter file: spar-def.par										
No data read yet										
Time for vehicle 1:	Number of vehicles:		4	Maxiter:		30	Weight of cost:		1.0	
Time for vehicle 2:	Start variant:		15	Fixed cost for vehicle:		200.0	Weight of time:		1.0	
Time for vehicle 3:	Move variant:		3	Target cost:		0.0	<input checked="" type="checkbox"/> Mark done? <input checked="" type="checkbox"/> Revise sol? <input checked="" type="checkbox"/> Move cycles only if improve? <input type="checkbox"/> Try all cycle swaps? <input type="checkbox"/> Print more?			
Time for vehicle 4:	Change order variant:		5	Target time:		0.0				
	Improvement variant:		3	SA: Start temp:		1000.0				
	Cycle moving variant:		4	SA: Internal iter:		7				
				SA: Cooling factor:		0.3				

“Update”: spara inställningarna. “Info”: Info.

# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan										
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit	
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset
Nonexisting indata file										
Parameter file: spar-def.par										
No data read yet										
Number of vehicles:	4		Maxiter:	30		Weight of cost:	1.0			
Time for vehicle 1:	Start variant:		15		Fixed cost for vehicle:	200.0		Weight of time:	1.0	
Time for vehicle 2:	Move variant:		3		Target cost:	0.0				
Time for vehicle 3:	Change order variant:		5		Target time:	0.0		<input checked="" type="checkbox"/>	Mark done?	
Time for vehicle 4:	Improvement variant:		3		SA: Start temp:	1000.0		<input checked="" type="checkbox"/>	Revise sol?	
	Cycle moving variant:		4		SA: Internal iter:	7		<input checked="" type="checkbox"/>	Move cycles only if improve?	
					SA: Cooling factor:	0.3		<input type="checkbox"/>	Try all cycle swaps?	
								<input type="checkbox"/>	Print more?	

“Update”: spara inställningarna. “Info”: Info. “Lowb”, “Chin lowb”: undre gräns.

# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan										
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit	
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset
Nonexisting indata file										
Parameter file: spar-def.par										
No data read yet										
Time for vehicle 1:		Number of vehicles: 4		Maxiter: 30		Weight of cost: 1.0				
Time for vehicle 2:		Start variant: 15		Fixed cost for vehicle: 200.0		Weight of time: 1.0				
Time for vehicle 3:		Move variant: 3		Target cost: 0.0		<input checked="" type="checkbox"/> Mark done?				
Time for vehicle 4:		Change order variant: 5		Target time: 0.0		<input checked="" type="checkbox"/> Revise sol?				
		Improvement variant: 3		SA: Start temp: 1000.0		<input checked="" type="checkbox"/> Move cycles only if improve?				
		Cycle moving variant: 4		SA: Internal iter: 7		<input type="checkbox"/> Try all cycle swaps?				
				SA: Cooling factor: 0.3		<input type="checkbox"/> Print more?				

“Update”: spara inställningarna. “Info”: Info. “Lowb”, “Chin lowb”: undre gräns. “Solve”: lös problemet.

# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan											
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit		
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset	
Nonexisting indata file											
Parameter file: spar-def.par											
No data read yet											
Time for vehicle 1:	Number of vehicles:		4		Maxiter:		30		Weight of cost:		1.0
Time for vehicle 2:	Start variant:		15		Fixed cost for vehicle:		200.0		Weight of time:		1.0
Time for vehicle 3:	Move variant:		3		Target cost:		0.0		<input checked="" type="checkbox"/> Mark done?		
Time for vehicle 4:	Change order variant:		5		Target time:		0.0		<input checked="" type="checkbox"/> Revise sol?		
	Improvement variant:		3		SA: Start temp:		1000.0		<input checked="" type="checkbox"/> Move cycles only if improve?		
	Cycle moving variant:		4		SA: Internal iter:		7		<input type="checkbox"/> Try all cycle swaps?		
					SA: Cooling factor:		0.3		<input type="checkbox"/> Print more?		

“Update”: spara inställningarna. “Info”: Info. “Lowb”, “Chin lowb”: undre gräns. “Solve”: lös problemet. “Step”: gör en iteration.

# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan										
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit	
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset
Nonexisting indata file										
Parameter file: spar-def.par										
No data read yet										
Time for vehicle 1:	Number of vehicles:		4	Maxiter:		30	Weight of cost:		1.0	
Time for vehicle 2:	Start variant:		15	Fixed cost for vehicle:		200.0	Weight of time:		1.0	
Time for vehicle 3:	Move variant:		3	Target cost:		0.0	<input checked="" type="checkbox"/> Mark done? <input checked="" type="checkbox"/> Revise sol? <input checked="" type="checkbox"/> Move cycles only if improve? <input type="checkbox"/> Try all cycle swaps? <input type="checkbox"/> Print more?			
Time for vehicle 4:	Change order variant:		5	Target time:		0.0				
	Improvement variant:		3	SA: Start temp:		1000.0				
	Cycle moving variant:		4	SA: Internal iter:		7				
				SA: Cooling factor:		0.3				

“Update”: spara inställningarna. “Info”: Info. “Lowb”, “Chin lowb”: undre gräns. “Solve”: lös problemet. “Step”: gör en iteration. “Run Vineopt”: starta Vineopt med nuvarande allokering.

# TAOP61 Projekt 4: Snowplan: Parametrar

Snowplan										
Update	Solve	Step	Reset step	Stop	Lowb	Chin lowb	See alloc	Info	Quit	
Read param		Save param		Read network		Run Vineopt		Read linkset from Vineopt		Read linkset
Nonexisting indata file										
Parameter file: spar-def.par										
No data read yet										
Number of vehicles: 4		Maxiter: 30		Weight of cost: 1.0						
Start variant: 15		Fixed cost for vehicle: 200.0		Weight of time: 1.0						
Move variant: 3		Target cost: 0.0		<input checked="" type="checkbox"/> Mark done?						
Change order variant: 5		Target time: 0.0		<input checked="" type="checkbox"/> Revise sol?						
Improvement variant: 3		SA: Start temp: 1000.0		<input checked="" type="checkbox"/> Move cycles only if improve?						
Cycle moving variant: 4		SA: Internal iter: 7		<input type="checkbox"/> Try all cycle swaps?						
		SA: Cooling factor: 0.3		<input type="checkbox"/> Print more?						

“Update”: spara inställningarna. “Info”: Info. “Lowb”, “Chin lowb”: undre gräns. “Solve”: lös problemet. “Step”: gör en iteration. “Run Vineopt”: starta Vineopt med nuvarande allokering. “Read linkset from Vineopt”: läs allokering från Vineopt.



## TAOP61 Projekt 4: Snowplan: Funktion *startalloc*

Skapar en fordonsallokering (*vehdo*), styrs av parametern *startvar*:

*startvar* = 1: Skapa en slumpmässig allokering.

*startvar* = 2: Läs en bågmängd från fil som startallokering.

*startvar* = 3: Låt fordon 1 göra allt.

*startvar* = 4: Sortera i  $x$  för startnod, fyll efterhand.

*startvar* = 5: Som 4, men sortera i  $x$  för slutnod.

*startvar* = 6: Som 4, men sortera i  $x$  för mittpunkt.

*startvar* = 7-9: Som 4-6, men sortera i  $y$ .

*startvar* = 10: Som 4, men sortera i avstånd till origo för startnod.

*startvar* = 11-12: Som 10, men sortera slutnod/mittpunkt.

*startvar* = 13: *kmeans* skapar ett kluster för varje fordon. En båge allokeras till klustret med högst nummer av ändnoderna.

*startvar* = 14: Som 13, men med lägst nummer.

*startvar* = 15: Som 13, men slumpvis till ett av klustren.

*startvar* = 16: Som 13, men min distans.

Ändrar allokeringen av bågar till fordon (vektorn *vehdo*).

Styrs av parametrarna *movevar* och *moalloc*:

*movevar* = 1: Byt två slumpmässiga länkar.

*movevar* = 2: Ändra *moalloc* slumpmässiga länkar.

*movevar* = 3: Flytta en från ett fordon som har mest till ett som har minst.

*movevar* = 4: Flytta en båge mellan två slumpade fordon där båda noderna är gemensamma.

*movevar* = 5: Byt mellan två slumpade fordon där båda noderna är gemensamma.

*movevar* = 6: Flytta en båge, där en nod är gemensam.

*movevar* = 7: Byt en båge, där en nod är gemensam.

Ändrar ordningen. Styrts av parametrar *chordvar* och *moord*:

*chordvar* = 1: Byt två slumpmässiga platser i vektorn *order*.

*chordvar* = 2: Skapa helt ny slumpmässig ordning.

*chordvar* = 3: Skifta vektorn *order* *moord* steg åt höger.

*chordvar* = 4: Skifta vektorn *order* *moord* steg åt vänster.

*chordvar* = 5: Byt den som har flest mot den som har minst.

## TAOP61 Projekt 4: Snowplan: Funktion *improvesol*

Förbättrar lösningen: Finn en cykel i en tur, kolla om cykeln innehåller en nod i en annan tur, flytta cykeln till den andra turen.

Styrs av parametern *impvar*:

*impvar* = 1: Flytta cykler, ett slumpförsök.

*impvar* = 2: Flytta cykler, sluta när ingen förbättring sker.

*impvar* = 3: Flytta cykler, kolla alla kombinationer.

## TAOP61 Projekt 4: Snowplan: Funktion *improvesol*

Förbättrar lösningen: Finn en cykel i en tur, kolla om cykeln innehåller en nod i en annan tur, flytta cykeln till den andra turen.

Styrs av parametern *impvar*:

*impvar* = 1: Flytta cykler, ett slumpförsök.

*impvar* = 2: Flytta cykler, sluta när ingen förbättring sker.

*impvar* = 3: Flytta cykler, kolla alla kombinationer.

Funktion *cyclemove*:

Anropas från *improvesol*.

Styrs av parametern *cymvar*:

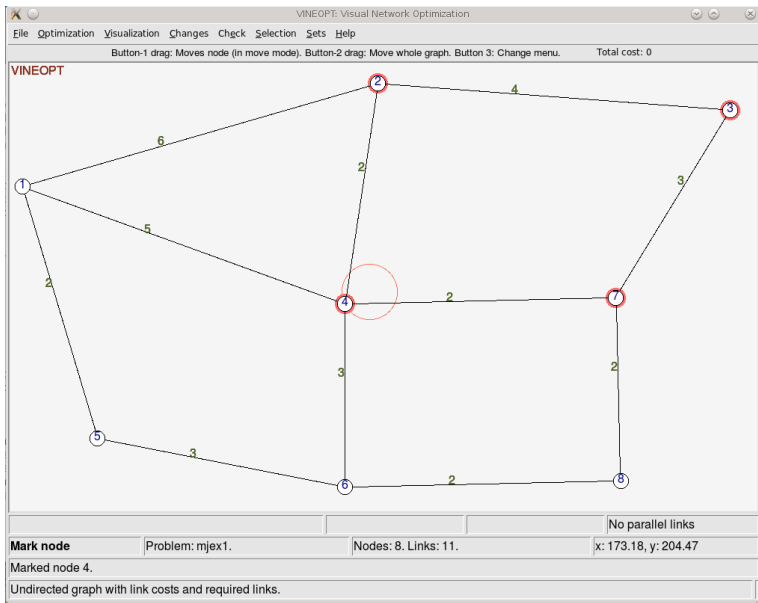
*cymvar* = 1: Försök flytta en slumpad cykel mellan två slumpade fordon.

*cymvar* = 2: Försök flytta en cykel från en lång tur till en kort.

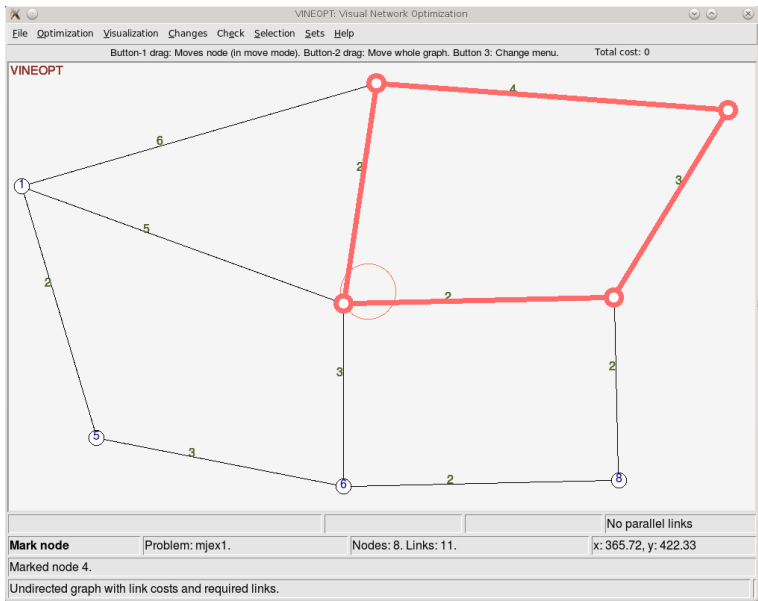
*cymvar* = 3: Försök flytta en cykel mellan givna fordon.

*cymvar* = 4: Försök flytta en cykel från en dyr tur till en billig.

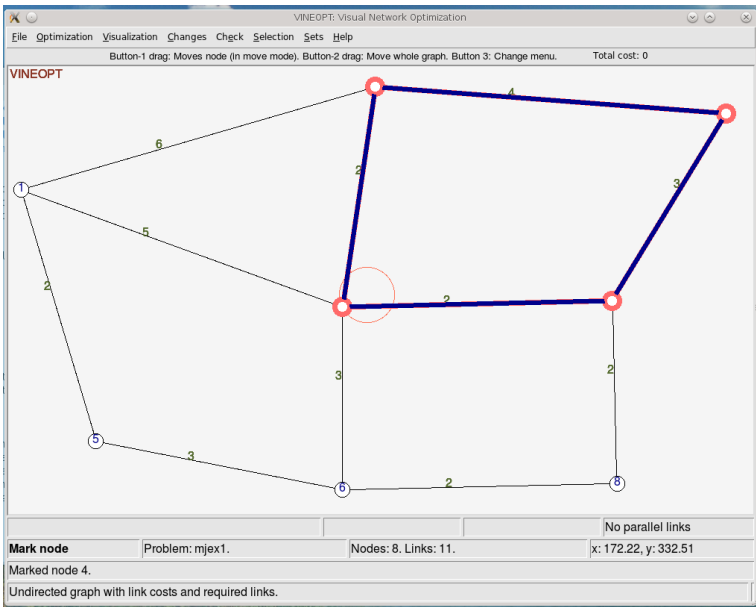
# TAOP61 Projekt 4: Vineopt



# TAOP61 Projekt 4: Vineopt



# TAOP61 Projekt 4: Vineopt





# TAOP61 Projekt 4: Vineopt

VINEOPT: Visual Network Optimization

File Optimization Visualization Changes Check Selection Sets Help

Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu. Total cost: 11

VINEOPT

Solution found. Total cost: 11.

OK

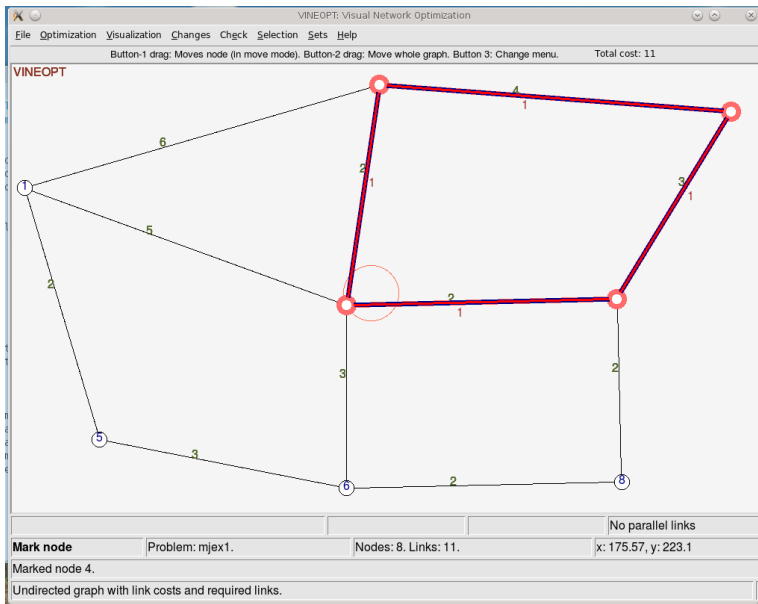
No parallel links

Mark node Problem: mjex1. Nodes: 8. Links: 11.

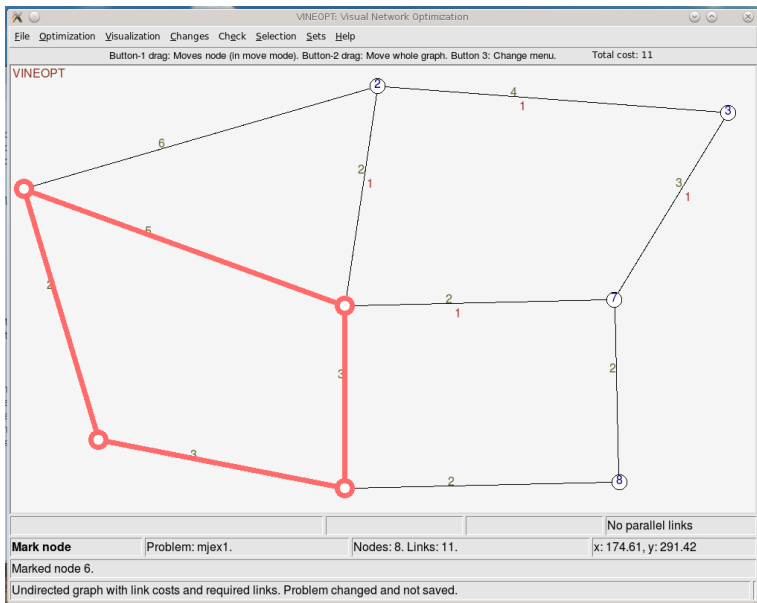
Marked node 4.

Undirected graph with link costs and required links.

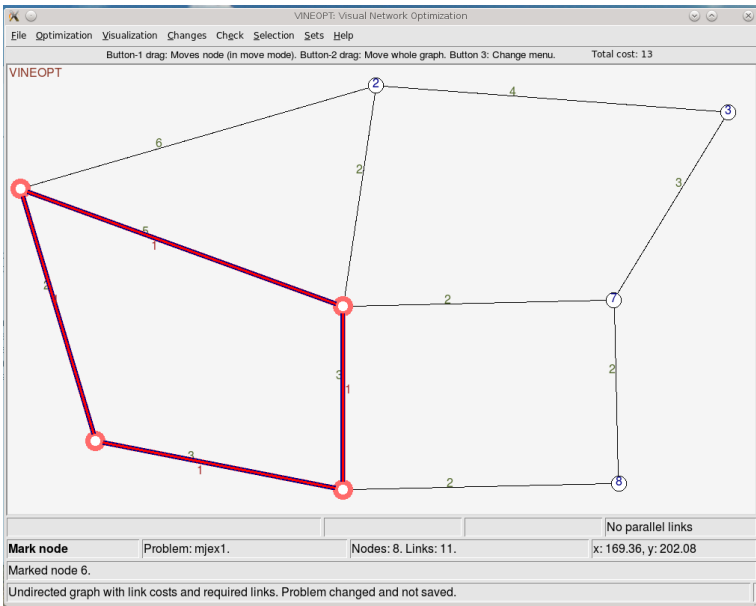
# TAOP61 Projekt 4: Vineopt



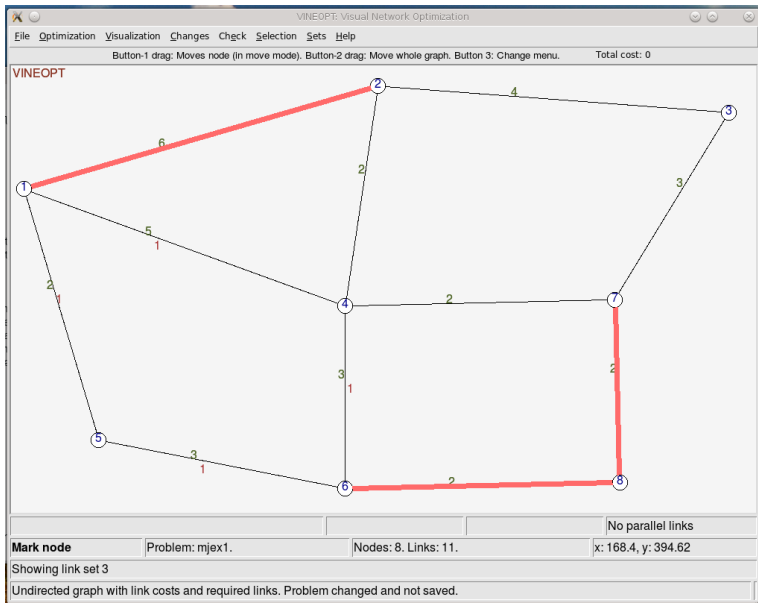
# TAOP61 Projekt 4: Vineopt



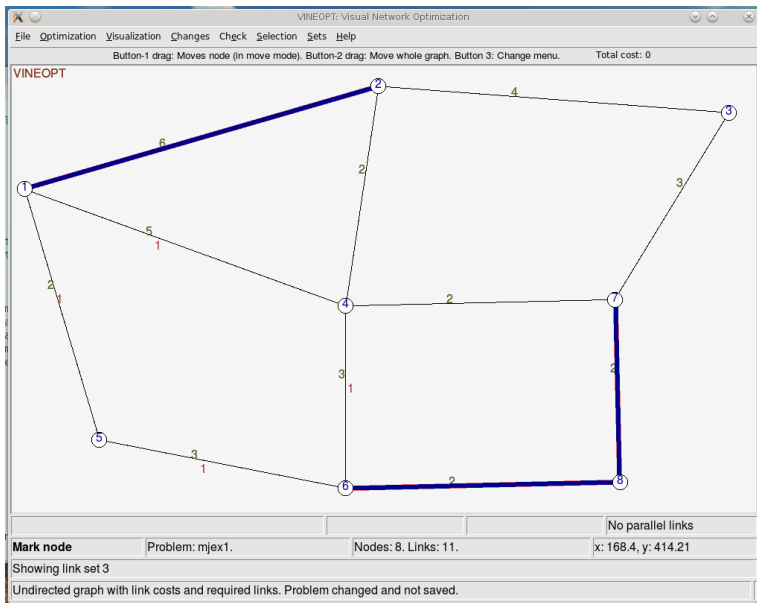
# TAOP61 Projekt 4: Vineopt



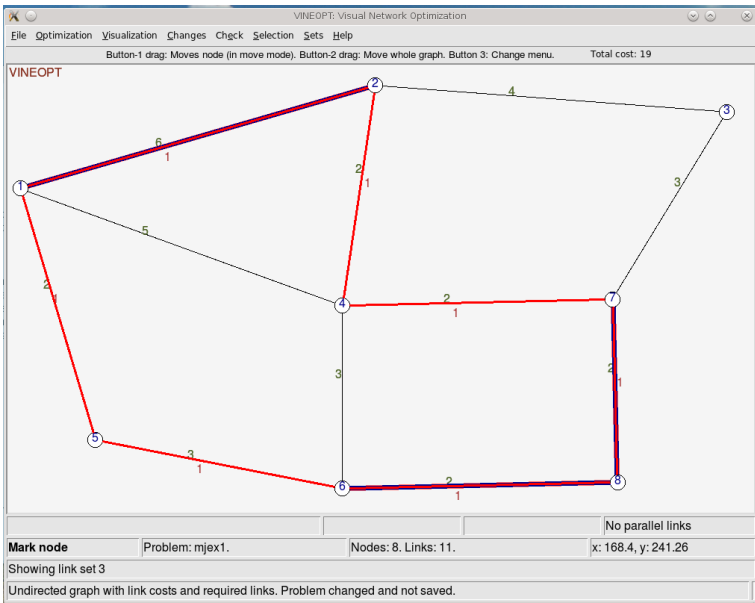
# TAOP61 Projekt 4: Vineopt



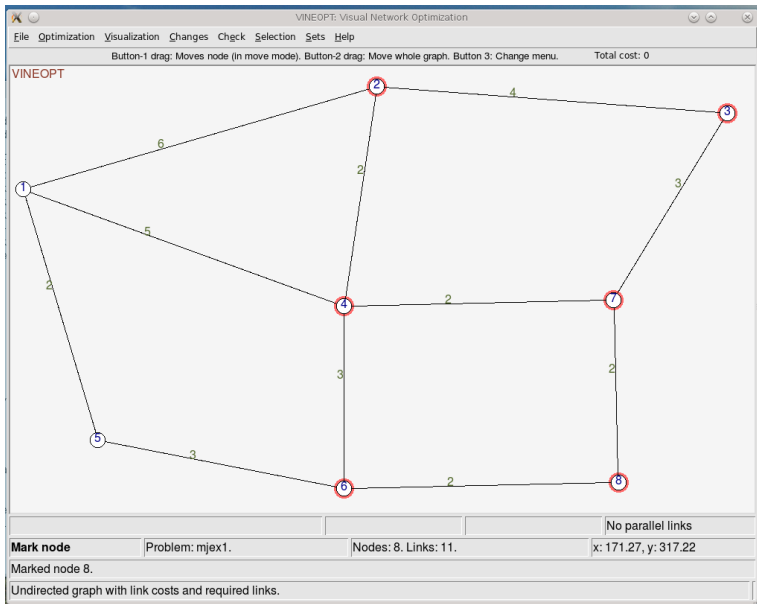
# TAOP61 Projekt 4: Vineopt



# TAOP61 Projekt 4: Vineopt

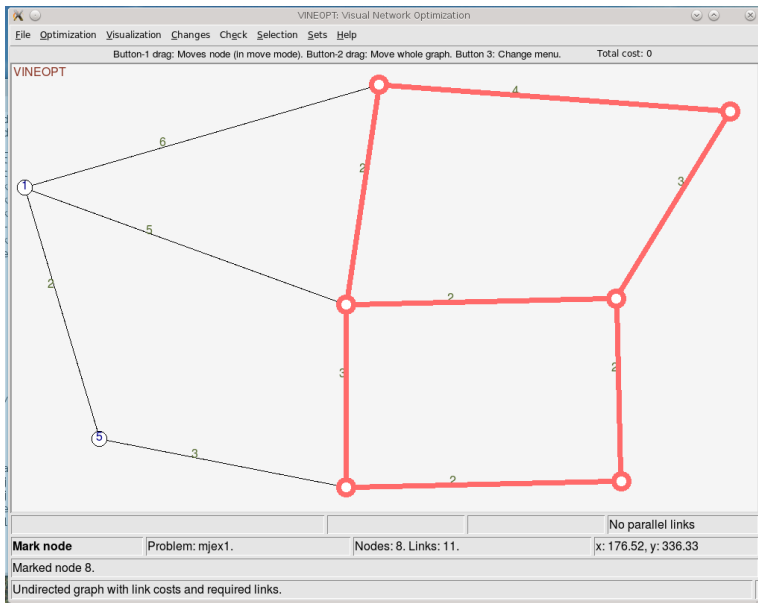


# TAOP61 Projekt 4: Vineopt

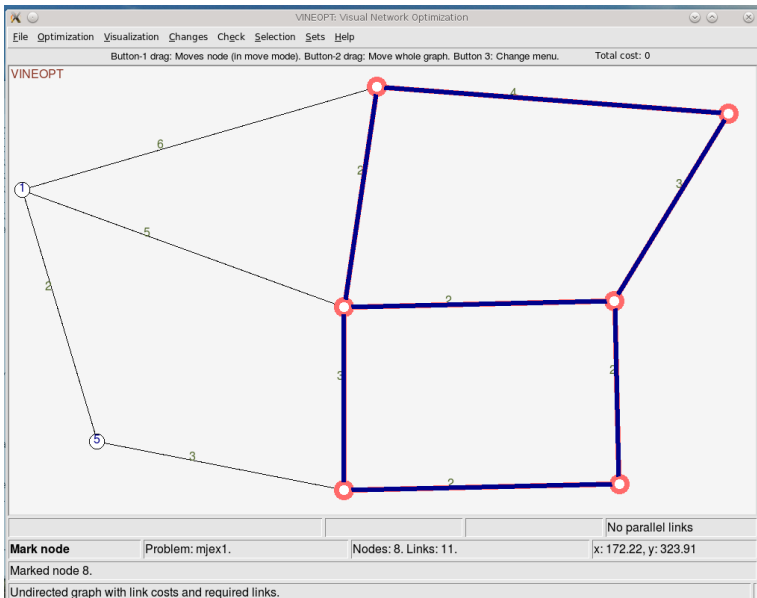




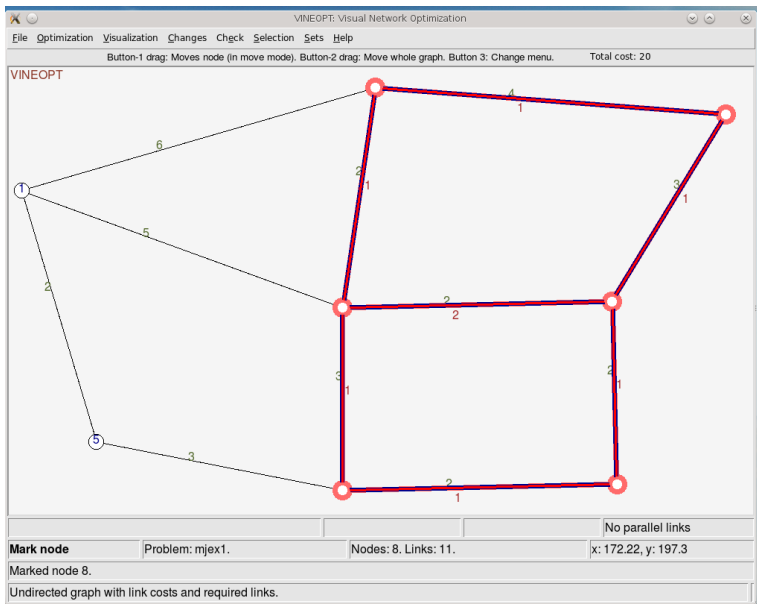
# TAOP61 Projekt 4: Vineopt



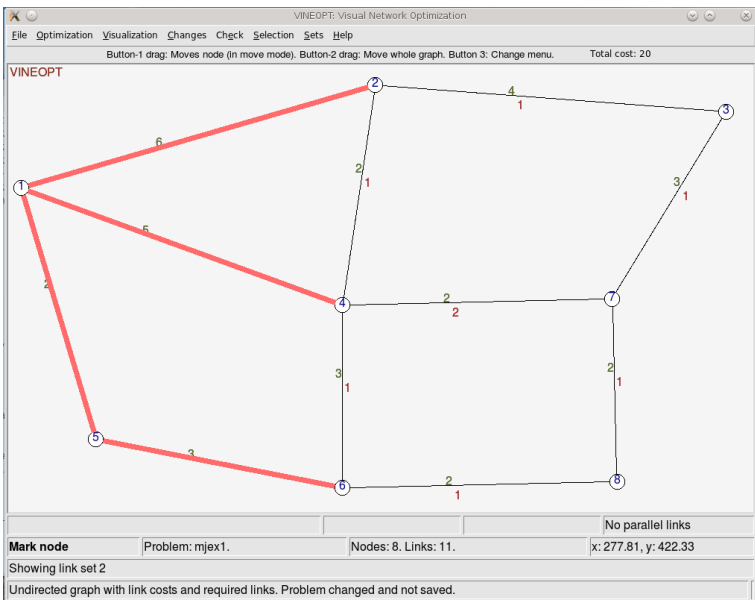
# TAOP61 Projekt 4: Vineopt



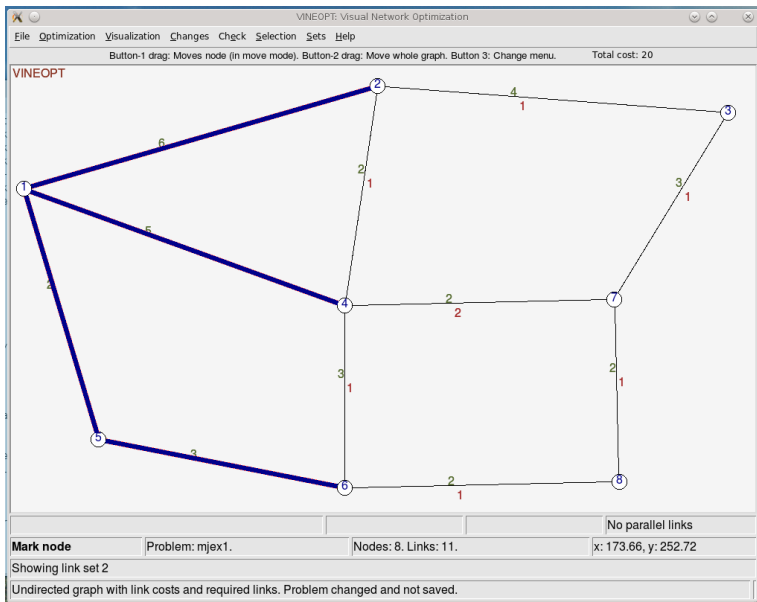
# TAOP61 Projekt 4: Vineopt



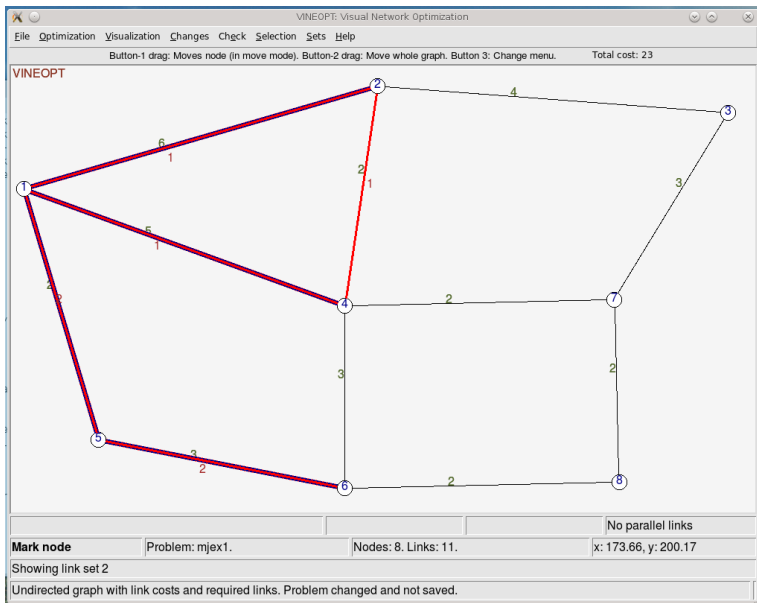
# TAOP61 Projekt 4: Vineopt



# TAOP61 Projekt 4: Vineopt



# TAOP61 Projekt 4: Vineopt



# TAOP61 Projekt 4: Vineopt

Med tre fordon:

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11.



## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13.

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19.

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3*7 = 64$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20.

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23.



## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2 \cdot 7 = 57$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2 \cdot 7 = 57$ .

Maxtid:  $\max(20, 23) = 23$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3*7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2*7 = 57$ .

Maxtid:  $\max(20, 23) = 23$ .

Mått (1, 1): Tre:  $64 + 19 = 83$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2 \cdot 7 = 57$ .

Maxtid:  $\max(20, 23) = 23$ .

Mått (1, 1): Tre:  $64 + 19 = 83$ . Två:  $57 + 23 = 80$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2 \cdot 7 = 57$ .

Maxtid:  $\max(20, 23) = 23$ .

Mått (1, 1): Tre:  $64 + 19 = 83$ . Två:  $57 + 23 = 80$ . Två bäst.

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2 \cdot 7 = 57$ .

Maxtid:  $\max(20, 23) = 23$ .

Mått (1, 1): Tre:  $64 + 19 = 83$ . Två:  $57 + 23 = 80$ . Två bäst.

Mått (1, 2): Tre:  $64 + 2 \cdot 19 = 102$ .

## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2 \cdot 7 = 57$ .

Maxtid:  $\max(20, 23) = 23$ .

Mått (1, 1): Tre:  $64 + 19 = 83$ . Två:  $57 + 23 = 80$ . Två bäst.

Mått (1, 2): Tre:  $64 + 2 \cdot 19 = 102$ . Två:  $57 + 2 \cdot 23 = 103$ .



## TAOP61 Projekt 4: Vineopt

Med tre fordon:

Turen för fordon 1 kostade 11. Turen för fordon 2 kostade 13. Turen för fordon 3 kostade 19. Summa rörliga kostnader:  $11 + 13 + 19 = 43$ .

Fast kostnad 7 per fordon:  $43 + 3 \cdot 7 = 64$ .

Maxtid:  $\max(11, 13, 19) = 19$ .

Med två fordon:

Turen för fordon 1 kostade 20. Turen för fordon 2 kostade 23. Summa rörliga kostnader:  $20 + 23 = 43$ .

Fast kostnad 7 per fordon:  $43 + 2 \cdot 7 = 57$ .

Maxtid:  $\max(20, 23) = 23$ .

Mått (1, 1): Tre:  $64 + 19 = 83$ . Två:  $57 + 23 = 80$ . Två bäst.

Mått (1, 2): Tre:  $64 + 2 \cdot 19 = 102$ . Två:  $57 + 2 \cdot 23 = 103$ . Tre bäst.

# TAOP61 Projekt 4: Uppgifter

1. Betrakta det lilla testproblemet *colonia*.

- a) Lös problemet för ett fordon. Notera lösningen, totalkostnaden samt vilken tid röjningen tar.
- b) Finn en lösning för två fordon genom att för hand i Vineopt ange vilka länkar som ska åtgärdas av varje fordon. Försök att hitta en bra uppdelning. Notera bästa lösningen, totalkostnaden samt vilken tid röjningen tar.
- c) Finn en lösning för tre fordon på samma sätt som i uppgift 1b.
- d) Antag att man ger lika vikt vid tid och kostnad. Vilken lösning är bäst?
- e) Antag att tiden är dubbelt så viktig som kostnaden, dvs. tiden får vikten två och kostnaden vikten ett. Vilken lösning är då bäst?

2. Betrakta problemet *colonia*.

- a) Lös problemet för två fordon med heuristiken snowplan. Jämför resultatet med uppgift 1b.
- b) Lös problemet för tre fordon med snowplan. Jämför med uppgift 1c.
- c) Besvara frågorna 1d och 1e för dessa lösningar.

## TAOP61 Projekt 4: Uppgifter

2. Betrakta problemet *colonia*.

- a) Lös problemet för två fordon med heuristiken snowplan. Jämför resultatet med uppgift 1b.
- b) Lös problemet för tre fordon med snowplan. Jämför med uppgift 1c.
- c) Besvara frågorna 1d och 1e för dessa lösningar.

3. Betrakta det lite större problemet *skanninge-n* (Skänninge norra). Gör samma sak som i uppgift 1 och 2.

## TAOP61 Projekt 4: Uppgifter

2. Betrakta problemet *colonia*.

- a) Lös problemet för två fordon med heuristiken snowplan. Jämför resultatet med uppgift 1b.
- b) Lös problemet för tre fordon med snowplan. Jämför med uppgift 1c.
- c) Besvara frågorna 1d och 1e för dessa lösningar.

3. Betrakta det lite större problemet *skanninge-n* (Skänninge norra). Gör samma sak som i uppgift 1 och 2. (Tips: Genom att läsa in Vineopt-lösningen i snowplan och bara göra en iteration, får man målfunktionsvärdet.)

4. Betrakta problemet *atvid-s* (Åtvidaberg södra). Finn lösningar för 1, 2, 3 och 4 fordon med snowplan. Notera målfunktionsvärde, totalkostnad och tid, samt lösningstid för Snowplan. Finn bästa antal fordon med lika vikt vid tid och kostnad, samt med tiden dubbelt så viktig som kostnaden. Studera den bästa lösningen/uppdelningen i Vineopt.

## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

Kör först snowplan. Notera hur lång tid de olika fordonen tar. Läs sedan in lösningen i Vineopt, och försök förbättra den genom att flytta bågar från en mängd till en annan.



## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

Kör först snowplan. Notera hur lång tid de olika fordonen tar. Läs sedan in lösningen i Vineopt, och försök förbättra den genom att flytta bågar från en mängd till en annan.

(Man kan se alla bågmängder samtidigt i olika färger.)

## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

Kör först snowplan. Notera hur lång tid de olika fordonen tar. Läs sedan in lösningen i Vineopt, och försök förbättra den genom att flytta bågar från en mängd till en annan.

(Man kan se alla bågmängder samtidigt i olika färger.)

Ta fram några troliga förbättringar och spara dessa bågmängder på fil.

## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

Kör först snowplan. Notera hur lång tid de olika fordonen tar. Läs sedan in lösningen i Vineopt, och försök förbättra den genom att flytta bågar från en mängd till en annan.

(Man kan se alla bågmängder samtidigt i olika färger.)

Ta fram några troliga förbättringar och spara dessa bågmängder på fil.

Läs därefter in dem i snowplan. Gör en iteration för att få målfunktionsvärdet (blev det bättre?), och/eller flera iterationer för att se om lösningen kan förbättras ytterligare.

## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

Kör först snowplan. Notera hur lång tid de olika fordonen tar. Läs sedan in lösningen i Vineopt, och försök förbättra den genom att flytta bågar från en mängd till en annan.

(Man kan se alla bågmängder samtidigt i olika färger.)

Ta fram några troliga förbättringar och spara dessa bågmängder på fil.

Läs därefter in dem i snowplan. Gör en iteration för att få målfunktionsvärdet (blev det bättre?), och/eller flera iterationer för att se om lösningen kan förbättras ytterligare.

Gör ytterligare försök att förbättra lösningen i Vineopt + snowplan.

## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

Kör först snowplan. Notera hur lång tid de olika fordonen tar. Läs sedan in lösningen i Vineopt, och försök förbättra den genom att flytta bågar från en mängd till en annan.

(Man kan se alla bågmängder samtidigt i olika färger.)

Ta fram några troliga förbättringar och spara dessa bågmängder på fil.

Läs därefter in dem i snowplan. Gör en iteration för att få målfunktionsvärdet (blev det bättre?), och/eller flera iterationer för att se om lösningen kan förbättras ytterligare.

Gör ytterligare försök att förbättra lösningen i Vineopt + snowplan.

Lämna därefter in ert anbud: målfunktionsvärde, tid för varje fordon, samt en fil med fordonsallokeringar (bågmängder). Lägst bud vinner.

## TAOP61 Projekt 4: Uppgifter

5. Finn en mycket bra lösning för det största problemet *vadstena*, för 6 fordon och lika vikt på kostnad och tid på följande sätt.

Kör först snowplan. Notera hur lång tid de olika fordonen tar. Läs sedan in lösningen i Vineopt, och försök förbättra den genom att flytta bågar från en mängd till en annan.

(Man kan se alla bågmängder samtidigt i olika färger.)

Ta fram några troliga förbättringar och spara dessa bågmängder på fil.

Läs därefter in dem i snowplan. Gör en iteration för att få målfunktionsvärdet (blev det bättre?), och/eller flera iterationer för att se om lösningen kan förbättras ytterligare.

Gör ytterligare försök att förbättra lösningen i Vineopt + snowplan.

Lämna därefter in ert anbud: målfunktionsvärde, tid för varje fordon, samt en fil med fordonsallokeringar (bågmängder). Lägst bud vinner.

6. Frivillig uppgift: Gör något kul med problemet *studentryd*.