DEPARTMENT OF ECONOMICS
AND BUSINESS ECONOMICS
AARHUS UNIVERSITY

EKSAMENSOPGAVE

<table>
<tr><td>Uddannelse og niveau</td><td colspan="7">HA, 5. semester, valgfag</td></tr>
<tr><td>Termin</td><td colspan="7">V21-22o</td></tr>
<tr><td>Kursusnavn og eksamenskode(r)</td><td colspan="6">Modellering inden for prescriptive analytics</td><td>460181E018</td></tr>
<tr><td>Eksamensform og varighed</td><td colspan="6">WOAI</td><td>3 timer</td></tr>
<tr><td>Dato og tidspunkt</td><td colspan="6">3. januar 2022</td><td>Kl. 14:00 – 17:00</td></tr>
<tr><td>Hjælpemidler</td><td>Alle</td><td>X</td><td>Anviste</td><td></td><td>Ingen</td><td></td><td></td></tr>
<tr><td>Anden relevant information</td><td colspan="7">Undgå mistanke om eksamenssnyd!

Husk kildehenvisninger og citationstegn, hvis du kopierer andres tekster eller hvis du genbruger dele af en tidligere afleveret opgave (plagiering og selvplagiering). Eksamensbesvarelsen skal udarbejdes individuelt. Der udføres plagieringskontrol på alle eksamensopgaver, så snyd og samarbejde mellem studerende vil kunne spores.</td></tr>
<tr><td>Det er tilladt at aflev. håndskrevet mat.</td><td>Ja</td><td></td><td>Nej</td><td>X</td><td colspan="3"></td></tr>
<tr><td>Anonym eksamen?</td><td>Ja</td><td>x</td><td>Nej</td><td></td><td colspan="3">Undlad venligst at anføre navn og studienummer i besvarelsen. Brug i stedet flow-løbenr., som du finder på dit omslag.</td></tr>
<tr><td>Antal sider
(inkl. forside)</td><td colspan="7">4</td></tr>
</table>

Preamble

- Indeværende eksamensopgave indeholder 9 opgaver. Ud for hver opgave er der anført en *vejledende vægtning* af opgaven, som udelukkende giver information om den relative vægtning af opgaven baseret på opgavens omfang og kompleksitet. Den endelige bedømmelse af besvarelsen vil blive baseret på besvarelsens helhed.
- Såfremt du mener, at der i en opgave mangler informationer for, at du kan løse den, kan du gøre dig relevante antagelser og klart og tydeligt specificere disse.
- Denne eksamen er med alle hjælpemidler (WOAI). Dog er den *individuel* hvilket vil sige, at samarbejde med andre ikke er tilladt.
- Hvis du reproducerer modeller, beregninger og/eller forklaringer fra litteraturen, opgaver eller internettet bedes du klart og tydeligt angive kilder på disse for at undgå beskyldninger om plagiat.

Eksamensopgave

I en kommune skal det besluttes, hvor der skal oprettes COVID-19 testcentre. Der er udpeget 6 potentielle steder, hvor der kan etableres centre. I kommunen er der 10 landsbyer, som man ønsker, at de oprettede testcentre skal servicere. Borgerne i kommunen er generelt magelige og betragter et testcenter som ligger længere væk end 3 kilometer som værende irrelevant. Vi vil derfor i den resterende del af denne opgave beskrive en landsby som ”dækket”, hvis der etableres et testcenter inden for en afstand af 3 kilometer fra landsbyen.

De 6 potentielle placeringer af testcentre er nummereret fra 1 til 6, og de 10 landsbyer er nummereret fra 1 til 10. I den vedlagte Excel-fil er data som beskriver:

- Koordinaterne for de potentielle testcentre og for de 10 landsbyer.
- De estimerede månedlige driftsomkostninger (f_i) ved hvert potentielt testcenter $i = 1, \dots, 6$.
- Indbyggertallet (b_j) i landsby $j = 1, \dots, 10$.
- En afstandsmatrix ($d_{ij} = 1$), som angiver afstanden fra hver potentiel placering af et testcenter i og til hver landsby j.
- En matrix (a_{ij}), som angiver om afstanden mellem en potentiel placering i og en landsby j er mindre end eller lig med 3 kilometer ($a_{ij} = 1$) eller ej ($a_{ij} = 0$).

Opgave 1 (Ca. 20 %)

På baggrund af ovenstående, formuler en lineær heltalsmodel, som minimerer antallet af etablerede testcentre, givet at alle landsbyer skal have et testcenter inden for en afstand på 3 kilometer (hver landsby skal dækkes af mindst et testcenter). Relater din model til forelæsningsnoterne og modeller præsenteret heri.

Opgave 1 – svar

Definer binære variabler y_i for hvert potentielt testcenter, som antager værdien 1 såfremt et testcenter etableres på lokation i og 0 ellers. Vi ønsker at minimere antallet af etablerede centre, hvor objektfunktionen naturligt bliver

$$\min \sum_{i=1}^{6} y_i$$

Der skal være et testcenter inden for en radius af 3 kilometer fra hver landsby. Dette klares ved at kræve at mindst ét testcenter skal etableres inden for en radius af 3 kilometer fra hver landsby:

$$\sum_{i=1}^{6} a_{ij} y_i \geq 1, \quad \forall j = 1, \ldots, 10$$

Dette leder til den samlede model

$$\min \sum_{i=1}^{6} y_i$$

$$s.t.: \sum_{i=1}^{6} a_{ij} y_i \geq 1, \quad \forall j = 1, \ldots, 10$$

$$y_i \in \{0,1\}, \quad \forall i = 1, \ldots, 6$$

Denne model er beskrevet i kursets forelæsningsnoter side 26 som et covering location problem. Her er antallet af faciliteter der skal dække en ”kunde” lig med 1, og der er ingen øvre grænse for hvor mange faciliteter der skal placeres (da vi minimerer dette antal under alle omstændigheder).

Opgave 2 (Ca. 15%)

Implementer din model fra Opgave 1 i OPL og løs den ved hjælp af CPLEX. Beskriv din løsning og lav en grafisk illustration, som viser hvilke potentielle sites, der skal benyttes. Udregn de estimerede driftsomkostninger ved den beregnede løsning

Opgave 2 – svar

En implementering i OPL kan ses her:

```
int n = ...; // # of test sites
int m = ...; // # of villages
range testSites = 1..n; // Range for sites
range villages = 1..m; // Range for villages
int f[testSites] = ...; // Operating costs
int b[villages] = ...; // Inhabitants in villages
//a[i][j]=1 iff site i can cover village j
int a[testSites][villages] = ...;

dvar boolean y[testSites]; // y[i]=1 iff center at site i
dvar float operatingCosts; // Just a helper variable
minimize sum ( i in testSites ) y[i];// Minimize # of centers
subject to
{
  // Store total cost in a variable
  operatingCosts == sum ( i in testSites ) f[i] * y[i];
  // Covering constraints for each village
  forall ( j in villages )
  {
    sum ( i in testSites ) a[i][j] * y[i] >= 1;
  }
}
```

Med følgende datafil

```
n = 6;
m = 10;
f = [2000 3000 2200 2400 3200 4000];
b =[3757 3984 3312 1714 3911 4655 4411 3852 2939 3182];
a = [[  0   0   0   0   0   0   0   0   0   1   ],
     [  1   0   0   0   0   0   0   0   0   0   ],
     [  0   1   1   0   1   0   1   0   0   0   ],
     [  0   0   1   0   0   1   0   1   1   0   ],
     [  0   1   1   0   0   0   0   0   0   1   ],
     [  0   1   0   1   0   0   0   0   0   0   ]];
```

Følgende løsning opnås ved løsning med cplex: Testcentre skal åbnes ved sites 1, 2, 3, 4 og 6 og de samlede driftsomkostninger beløber sig til 13600. Landsbyerne er dækket som følger

Landsby	1	2	3	4	5	6	7	8	9	10
Dækket af site(s)	2	3, 6	3, 4	6	3	4	3	4	4	1

Læg gerne mærke til, at der er mere end en optimal løsning til dette problem. Ikke alle giver de samme driftsomkostninger, men alle skal åbne 5 testcentre.

Kommunen, som er drevet af et økonomisk ansvarligt flertal i byrådet, vil gerne vide hvor mange borgere man samlet kan dække med testcentre inden for en afstand af 3 kilometer, hvis man maksimalt vil acceptere en omkostning på 75% af omkostningen ved løsningen i Opgave 2. Hvis du ikke har beregnet en løsning i Opgave 2, kan du antage, at omkostningen var 13600.

Opgave 3 (Ca. 20%)

Formuler en lineær heltalsmodel, som maksimerer antallet af borgere, der kan nå et testcenter indenfor for 3 kilometer og under bi-betingelse af, at de samlede driftsomkostninger ikke må overstige 75% af omkostningen beregnet i Opgave 2.

Opgave 3 – svar

For at maksimerer antallet af borgere, der kan nå et testcenter inden for 3 kilometer, indføres nu nye binære variabler for hver landsby, defineret ved

$$z_j = \begin{cases} 1, hvis\ der\ er\ et\ testcenter\ inden\ for\ en\ radius\ af\ 3\ kilometer\ fra\ landsby\ j \\ 0, ellers \end{cases}$$

Objektfunktionen ændres nu til

$$\max \sum_{j=1}^{10} b_j z_j$$

Samtidig skal der indføres begrænsninger, som sørger for, at en landsby kun kan være "dækket" hvis der findes et testcenter inden for 3 kilometer. Dette gøres med begrænsningerne

$$\sum_{i=1}^{6} a_{ij} y_i \geq z_j, \quad \forall j = 1, \dots, 10$$

Slutteligt skal vi sørge for, at udgifterne ikke overstiger 75% af omkostningerne fra Opgave 2. Derfor lader vi $z_2^* = 13600$ være de optimale omkostninger fra Opgave 2. Budgetbegrænsningen bliver således

$$\sum_{i=1}^{6} f_i y_i \leq 0.75 * z_2^*$$

Opsummeres ovenstpende, fås følgende maximum covering location problem

$$\max \sum_{j=1}^{10} b_j z_j$$

$$s.t.: \sum_{i=1}^{6} a_{ij} y_i \geq z_j, \quad \forall j = 1, \dots, 10$$

$$\sum_{i=1}^{6} f_i y_i \leq 0.75 * z_2^*$$

$$y_i \in \{0,1\}, \quad \forall i = 1, \dots, 6$$

$$z_j \in \{0,1\}, \quad \forall j = 1, \dots, 10$$

Dette er som nævn en maximum set covering location model med en budgetbegrænsning (se Opgave 3.9 i forelæsningsnoterne side 29).

Opgave 4 (Ca. 15%)

Implementer din model fra Opgave 3 i OPL og løs den ved hjælp af CPLEX. Beskriv din løsning og lav en grafisk illustration, som viser hvilke potentielle sites, der skal benyttes og hvilke landsbyer, der kan nå et testcenter inden for en afstand af 3 kilometer. Sammenlign din løsning med løsningen fra Opgave 2.

Opgave 4 – svar

Implementering i OPL:

```
int n = ...; // # of test sites
int m = ...; // # of villages
range testSites = 1..n; // Range for sites
range villages = 1..m; // Range for villages
int f[testSites] = ...; // Operating costs
int b[villages] = ...; // Inhabitants in villages
//a[i][j]=1 iff site i can cover village j
int a[testSites][villages] = ...;

dvar boolean y[testSites]; // y[i]=1 iff center at site i
dvar boolean z[villages]; // z[j]=1 iff village j is covered

// maximize number of covered citizens
maximize sum ( j in villages ) b[j] * z[j];
subject to
{
  // Only count as covered, if a center is close enough
  forall ( j in villages )
  {
    sum ( i in testSites ) a[i][j] * y[i] >= z[j];
  }
  // Budget constraint
  sum ( i in testSites ) f[i] * y[i] <= 13600*0.75;
}
```

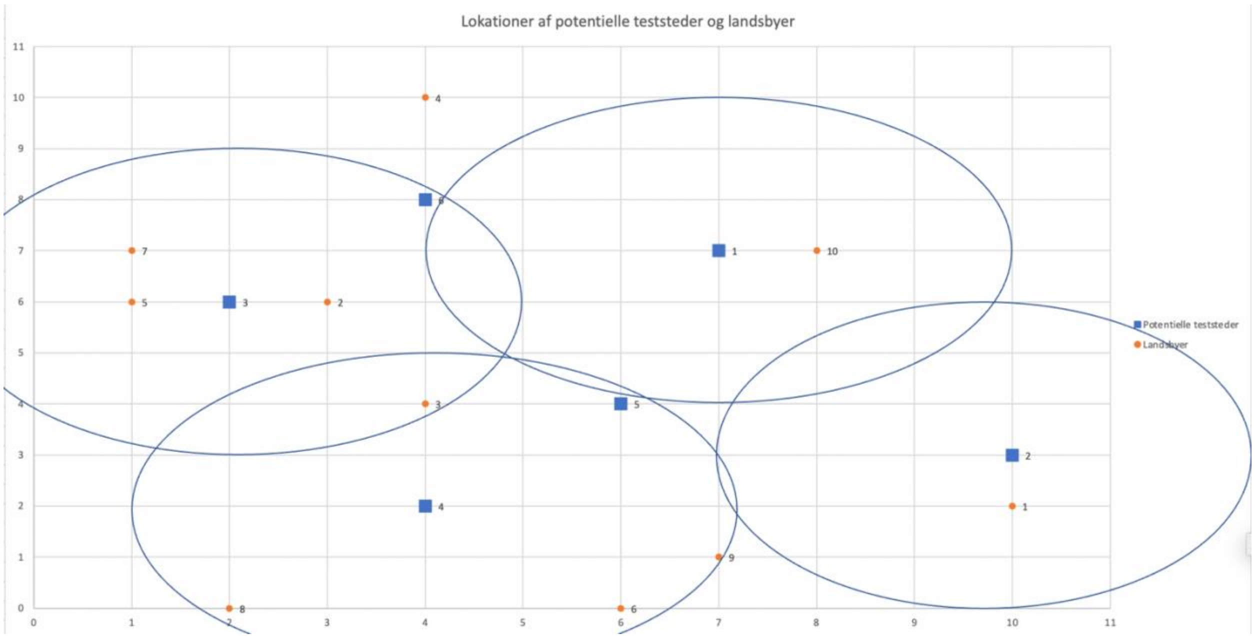
Der er benytte samme datafil som i Opgave 2.

Løsningen giver, at følgende testcentre skal åbnes: 1, 2, 3 og 4. De samlede omkostninger beløber sig til 9600.

Løsningen giver, at følgende landsbyer er dækket:

Landsbyer	1	2	3	4	5	6	7	8	9	10
Dækket (ja/nej)	Ja	Ja	Ja	Nej	Ja	Ja	Ja	Ja	Ja	Ja
Dækket af site(s)	2	3	3, 4	---	3	4	3	4	4	1

Grafisk illustration. Der er lagt cirkler på 3 kilometers radius omkring hvert åbent testcenter. De landsbyer, der ligger inden for cirklerne, er "dækket".



Opgave 5 (Ca. 5%)

Beskriv, med egne ord, hvad man forstår ved en Pareto optimal/efficient løsning. Beskriv, ligeledes med egne ord, hvad man forstår ved et ikke-domineret punkt.

Opgave 5 – svar

En Pareto optimal løsning, er en brugbar løsning, $\tilde{x}$, til et multi-kriterie optimeringsproblem. Desuden skal der gælder om $\tilde{x}$, at der ikke er muligt at forbedre én objektfunktion unden samtidig at forværre mindst en anden objektfunktion. Det vil sige, at $\tilde{x}$ er et rationelt kompromis mellem modsatrettede objektfunktioner.

Et ikke-domineret punkt, er afbildningen af en Pareto optimal løsning $\tilde{x}$ i *kriterierummet.* Det vil sige, at hvis objektfunktionerne til et multikriterie optimeringsproblem er givet ved $f_1(x), f_2(x), \dots, f_p(x)$, da er punktet $\tilde{y} \in \mathbb{R}^p$ givet ved $\tilde{y} = (f_1(\tilde{x}), f_2(\tilde{x}), \dots, f_p(\tilde{x}))$ et ikke-domineret punkt.

Kommunen ønsker en analyse af trade-off mellem antallet af borgere, der kan nå et testcenter inden for 3 kilometer og de forventede driftsomkostninger forbundet med de valgte testcentre.

Opgave 6 (Ca. 5%)

Med udgangspunkt i tidligere opgaver, formuler et bi-kriterieproblem, hvor man ønsker at maksimere antallet af borgere, der kan nå et testcenter inden for 3 kilometer og samtidigt ønsker at minimere de forventede driftsomkostninger.

Opgave 6 – svar

Ved at flytte budgetbegrænsningen fra Opgave 3 op som en objektfunktion, kan vi nu opstille en bi-kriterie model som fokuserer på at maksimere antallet af dækkede borger *og* minimering af de samlede driftsomkostninger:

$$\max \sum_{j=1}^{10} b_j z_j$$

$$\min \sum_{i=1}^{6} f_i y_i$$

$$s.t.: \sum_{i=1}^{6} a_{ij} y_i \geq z_j, \quad \forall j = 1, \dots, 10$$

$$y_i \in \{0,1\}, \quad \forall i = 1, \dots, 6$$

$$z_j \in \{0,1\}, \quad \forall j = 1, \dots, 10$$

Den første objektfunktion (fra opgave 3) maksimerer antallet af dækkede borgere, mens den anden objektfunktion mininmerer de samlede driftsomkostninger. Begrænsningerne $\sum_{i=1}^{6} a_{ij} y_i \geq z_j$ sørger for, at en landsby kun er dækket, såfremt der etableres et testcenter inden for en radius af 3 kilometer.

Opgave 7 (Ca. 10%)

Opskriv en ε-skalarisering af bi-kriterieproblemet fra Opgave 6. Forklar, med egne ord, hvordan ε-metoden fungerer. Beskriv mindst én styrke og én svaghed ved ε-metoden.

Opgave 7 – svar

En mulig ε-skalarisering af bi-kriterieproblemet fra Opgave 6 opnås ved at flytte de samlede driftsomkostninger (tilbage) til begrænsningerne således at budgettet er opadtil begrænset af en konstant ε:

$$\max \sum_{j=1}^{10} b_j z_j$$

$$s.t.: \sum_{i=1}^{6} a_{ij} y_i \geq z_j, \quad \forall j = 1, \dots, 10$$

$$\sum_{i=1}^{6} f_i y_i \leq \varepsilon$$

$$y_i \in \{0,1\}, \quad \forall i = 1, \dots, 6$$

$$z_j \in \{0,1\}, \quad \forall j = 1, \dots, 10$$

Opgave 8 (Ca. 5%)

Generer, ved hjælp af ε-metode beskrevet i Opgave 7, alle ikke-dominerede punkter for bi-kriterieproblemet fra Opgave 6. Beskriv din fremgangsmåde.

Opgave 8 – svar

Jeg vil ved gentagen løsning af ε-skalariseringen fra Opgave 7 generere alle ikke-dominerede punkter til bikriterieproblemet fra Opgave 6. Først sættes $\varepsilon = 13600$ således at budgettet ikke er begrænset opad til (vi ved fra Opgave 2, at vi kan dække alle landsbyer til en omkostning på 13600). Løses ε-skaliseringen for denne værdi af epsilon, fås en løsning bestående af følgende etablerede testcentre: 1, 2, 3, 4 og 6 til med driftsomkostninger på 13.600 og i alt 35.717 dækkede borgere. Programmet som blev løst er vist her:

```
int n = ...; // # of test sites
int m = ...; // # of villages
range testSites = 1..n; // Range for sites
range villages = 1..m; // Range for villages
int f[testSites] = ...; // Operating costs
int b[villages] = ...; // Inhabitants in villages
//a[i][j]=1 iff site i can cover village j
int a[testSites][villages] = ...;

dvar boolean y[testSites]; // y[i]=1 iff center at site i
dvar boolean z[villages]; // z[j]=1 iff village j is covered
dvar float operatingCosts; // Just a helper variable

float epsilon = 13600; // Epsilon

maximize sum ( j in villages ) b[j] * z[j];
subject to
{
  forall ( j in villages )
  {
    sum ( i in testSites ) a[i][j] * y[i] >= z[j];
  }
  operatingCosts == sum ( i in testSites ) f[i] * y[i];
  operatingCosts <= epsilon;
}
```

Driftsomkostningerne til denne løsning var 13.600 hvorfor ε opdateres: $\varepsilon = 13.600 - 1 = 13599$. Programmet løses igen og vi får at testcentrene 2, 3, 4 og 5 skal åbnes. Denne løsning dækker 34.003 borgere og koster 10800 kroner at drive. Programmet, som blev løst, er givet ved

```
int n = ...; // # of test sites
int m = ...; // # of villages
range testSites = 1..n; // Range for sites
range villages = 1..m; // Range for villages
int f[testSites] = ...; // Operating costs
int b[villages] = ...; // Inhabitants in villages
//a[i][j]=1 iff site i can cover village j
int a[testSites][villages] = ...;

dvar boolean y[testSites]; // y[i]=1 iff center at site i
dvar boolean z[villages]; // z[j]=1 iff village j is covered
dvar float operatingCosts; // Just a helper variable

float epsilon = 13599; // Epsilon

maximize sum ( j in villages ) b[j] * z[j];
subject to
{
  forall ( j in villages )
  {
    sum ( i in testSites ) a[i][j] * y[i] >= z[j];
  }
  operatingCosts == sum ( i in testSites ) f[i] * y[i];
  operatingCosts <= epsilon;
}
```

Driftsomkostningerne til denne løsning var 10.800 hvorfor ε opdateres: $\varepsilon = 10.800 - 1 = 10.799$.

Den næste løsning dækker også 34.003 borgere men koster kun 9.600 at drive. Metode fortsætter indtil løsningen hvor alle testcentre er lukkede og omkostningen derfor er 0. Den efterfølgende iteration af epsilon metode ($\varepsilon = 0 - 1 = -1$) vil derfor være *infeasible.* Proceduren stopper med følgende ikke-dominerede punkter

Løsninger	1	2	3	4	5	6	7	8	9	10
Dækkede borgere	35.717	34.003	34.003	30.821	30.246	27.064	18.800	15.618	3.182	0
Omkostninger	13.600	10.800	9.600	7.600	6.600	4.600	4.200	2.200	2.000	0

Ovenstående metode kan automatiseres med følgende main-script

```
main
{
  thisOplModel.generate();
  while ( cplex.solve () )
  {
    var costs = thisOplModel.operatingCosts;
    writeln ( cplex.getObjValue(),"\t", costs);
    thisOplModel.operatingCosts.UB = costs - 1;
  }
}
```

Opgave 9 (Ca. 5%)

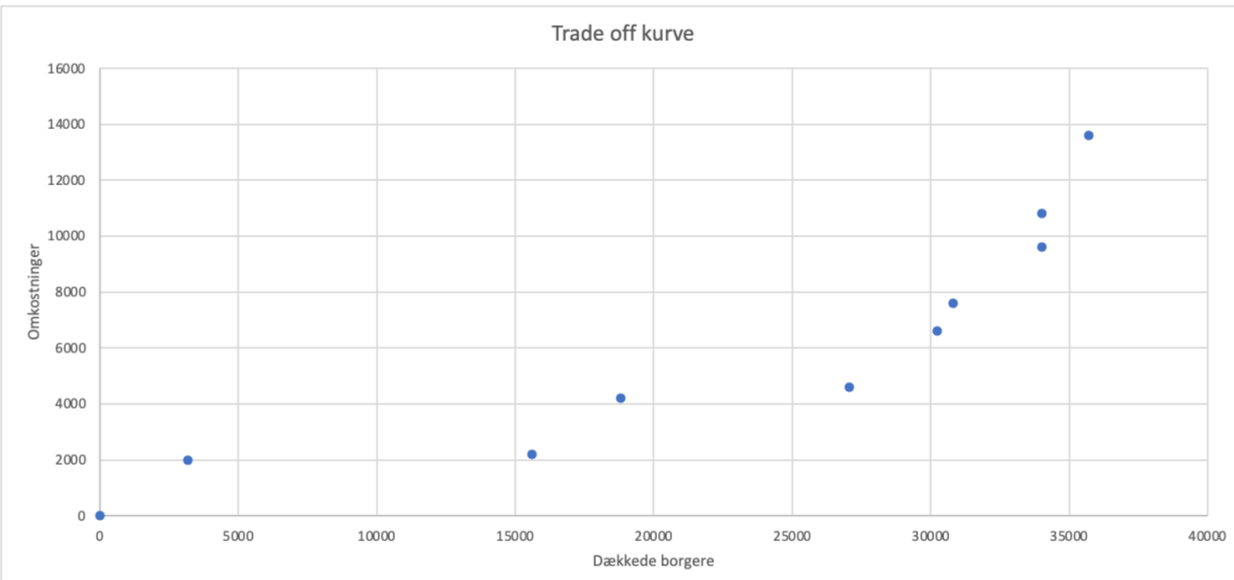
Er alle løsninger fundet i Opgave 8 ikke-dominerede? Plot løsningerne i et koordinatsystem og udvælg, baseret på dit plot, fire løsninger, som kunne være særligt interessante for kommunen. Begrund dit valg af løsninger.

Såfremt du ikke har besvaret Opgave 8, kan du antage at de fundne løsninger er givet som i tabellen nedenfor:

Løsninger	1	2	3	4	5	6	7	8	9	10
Dækkede borgere	35717	34003	34003	30821	30246	27064	18800	15618	3182	0
Omkostninger	13600	9600	10800	7600	6600	4600	4200	2200	2000	0

Opgave 9 – svar

Et plot af de ovenstående punkter giver følgende graf:



Det ses, at alle disse punkter *ikke* er ikke-dominerede. Punkt 2 dominerer punkt 3, idet de to korresponderende løsninger dækker det samme antal borgere, men punkt 2 har en lavere omkostning en punkt3. Derfor kan vi, hvis i står i punkt 3, forbedre den anden objektfunktion unden at forværre den første objektfunktion. Dermed er punkt 3 ikke et ikke-domineret punkt.

En række løsninger kunne her udvælges grundet deres forskellige karakteristika. Startende nederst til venstre, giver løsning 8 god mening at inkludere, da denne giver tæt på laveste omkostning mens test facilitet installeres. Samtidig giver løsning dækker denne løsning markant flere borgere end løsning 9 (som har de laveste omkostning) men giver kun en relativ lille stigning i omkostningerne.

På samme måde ser løsningerne 6 og 2 også interessante ud, da disse danner gode jævne kompromisser mellem de to objektfunktioner

Slutteligt giver det god mening at inddrage løsning 1, da denne giver fuld dækning, til lavest mulig omkostning (under bibetingelse af, at alle borgere skal dækkes).

Disse 4 løsninger er ikke de eneste der kan udvælges!