Laborationsinformation

Laboration 5: Lösning av heltalsproblem med heuristik

Laborationen syftar till att påvisa möjligheterna att utnyttja ett problems specifika struktur, dels genom att utveckla en alldeles egen heuristik för problemet.

Heuristiken implementeras troligtvis enklast i MATLAB, Python, Julia eller i ett valfritt programmeringsspråk. Det finns kort information om MATLAB och Python i mappen "Lab 5" på Lisam.

Problemställning

Det optimeringsproblem som ska behandlas i denna laboration är det kapaciterade lokaliseringsproblemet. Låt m vara antalet möjliga platser för lokalisering av anläggningar (t.ex. fabriker), n antalet kunder, d_j efterfrågan hos kund j, s_i kapaciteten hos en anläggning på plats i, f_i den fasta kostnaden för att öppna en anläggning på plats i, c_{ij} transportkostnaden per enhet till kund j från anläggningsplats i, och e en diskonteringsfaktor. Samtliga koefficienter kan antas vara positiva.

Variabeldefinition: $y_i = 1$ om en anläggning placeras på plats i, och 0 om inte.
x_{ij} = transporterad mängd från anläggning på plats i till kund j.

Problemet att finna lösningen med lägst totalkostnad kan modelleras på följande sätt.

$$
\textbf{(P1)} \quad
\begin{array}{rlrcll}
v^* = & \min & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} e f_i y_i & & & \\
& \text{s.t.} & \displaystyle\sum_{j=1}^{n} x_{ij} & \leq & s_i y_i & i = 1,\dots,m \quad (1) \\
& & \displaystyle\sum_{i=1}^{m} x_{ij} & = & d_j & j = 1,\dots,n \quad (2) \\
& & x_{ij} & \geq & 0 & i = 1,\dots,m, j = 1,\dots,n \quad (3) \\
& & y_i & \in & \{0,1\} & i = 1,\dots,m \quad (4)
\end{array}
$$

Ofta avser egentligen kostnaderna olika tidsperioder. De fasta anläggningskostnaderna kan avse nybyggnation idag, medan transportkostnaderna kan avse samlade transportkostnader över en längre period (kanske flera år). Man måste då göra dessa kostnader jämförbara, och diskonteringsfaktorn e kan användas för detta. Man kan även prova olika scenarior (för t.ex. olika nivå på räntan). De första körningarna ska göras med $e = 1$.

Ibland finns en konflikt mellan fasta och rörliga kostnader. En plats med låga transportkostnader ligger troligen centralt, vilket leder till höga priser för mark och byggnation, dvs. höga fasta kostnader. Låga fasta kostnader uppträder troligen långt från kunderna, vilket medför höga trasnportkostnader.

Ni ska lösa följande instanser av lokaliseringsproblemet.

Name	m	n
floc1	3	5
floc2	12	18
floc3	10	25
floc6	20	100
floc7	30	150
floc8	30	200

Det finns MATLAB-filer, floc1.m, floc2.m, osv, där indata tilldelas på ett sätt som kan användas i MATLAB, samt datafiler i MATLABs interna format, floc1.mat, etc, som kan läsas in med kommandot `load` i MATLAB. Det finns även floc1.npz, floc2.npz, osv, som kan läsas in i Python och floc1.jl, floc2.jl, osv, som kan läsas in i Julia samt datafiler floc1.txt, floc2.txt, osv, som kan användas i ett valfritt programmeringsspråk. Filerna finns i de separata mapparna i mappen "Lab 5" på Lisam.

Laborationsuppgifter

Resultat och svar på frågor skrivs ner på separat resultatblad. Laborationen redovisas under Inlämningar i Lisam. Man ska skicka in resultaten samt koden.

1. Hitta på och beskriv en egen girighetsheuristik för detta problem. Man ska försöka minimera den totala kostnaden, så det är inte tillåtet att strunta i anläggningskostnaderna eller transportkostnaderna. Tänk igenom implementeringen. Ett skal för koden i MATLAB och ett skal i python 2 finns i mappen "Lab 5" på Lisam som ni kan använda om ni vill.

2. Applicera er heuristik på dessa instanser genom att skriva ett litet datorprogram (MATLAB/Python/Julia rekommenderas). Skriv in kursens namn och era namn som en kommentar först i filen. Beskrivningen av metoden kan göras som utförliga kommentarer i koden eller separat. Hur bra lösningar hittar er heuristik jämfört med heltalsoptimum? Notera tiden programmet tar.

   Man kan göra ett generellt program som förutsätter att data redan finns, och sedan börja med att anropa t.ex. floc1.m. Det går dock snabbare (för stora problem) att läsa en mat-fil.

3. Lös floc3 med heuristiken för $e = 0.01$, 0.1, 1 (redan gjort), 10 och 100. Notera målfunktionsvärden, samt hur många anläggningar som öppnas. Jämför resultatet med de optimala lösningarna.

4. För vilka värden på e fungerar heuristiken bäst? Kommer värdet på diskonteringsfaktorn e att påverka hur bra heuristiken är jämfört med den optimala lösningen?

5. Ge en kort beskrivning av er heuristik. Gör även en bedömning av den. Vad är bra? Vad är dåligt? Vad kan förbättras? Vilka problemegenskaper passar den bättre/sämre för? (Stora/små fasta kostnader, stora/små kapaciteter, etc.)