Linköpings Tekniska Högskola | Kursmål
Matematiska institutionen/Optimeringslära | 101221
Kaj Holmberg | TAOP86

1 Kursmål för TAOP86: Kombinatorisk optimering med miljötillämpningar

1.1 Övergripande

Optimeringslära är ett ämne inom den tillämpade matematiken som behandlar olika former av beslutsproblem, traditionellt ofta av teknisk eller ekonomisk natur, men även angående planering av olika aktiviteter så att t.ex. negativ miljöpåverkan minimeras. Praktiska optimeringsproblem beskrivs av *matematiska modeller*, där de möjliga besluten beskrivs med *variabler*, målsättningen med en *målfunktion* och de möjliga besluten begränsas av givna restriktioner eller *bivillkor*. En *optimallösning* till ett problem är en lösning som uppfyller givna bivillkor och som minimerar/maximerar målfunktionens värde. Kombinatorisk optimering är ett delområde inom optimeringslära som behandlar problem med en underliggande kombinatorisk och/eller grafstruktur. Det är starkt relaterat till discipliner såsom diskret matematik och datalogi. Viktiga aspekter är hur man *formulerar* ett optimeringsproblem matematiskt och hur man *löser* problemet med en effektiv metod.

Ett övergripande mål i kursen är att du ska kunna identifiera, modellera och klassificera beslutsproblem av optimeringskaraktär. Du ska ha viss förmåga att bedöma huruvida ett verkligt problem passar för att angripa med optimeringsteknik eller ej. Du ska ha en färdighet i att bedöma ett kombinatoriskt optimeringsproblems svårighetsgrad med hjälp av komplexitetsteori. Vidare ska du ha kunskaper om några välkända typer av kombinatoriska optimeringsproblem, om algoritmer med vilka de effektivt kan lösas samt om uppbyggnaden av dessa algoritmer.

1.2 Grundläggande linjärprogramming

Många kombinatoriska optimeringsproblem har egenskapen att de kan formuleras och lösas som *linjära optimeringsproblem*. För att lösa ett sådant problem används *simplexmetoden*, som systematiskt avsöker potentiella optimallösningar (*tillåtna baslösningar*) i strävan efter att uppfylla *optimalitetsvillkor*. Till varje linjärt optimeringsproblem associeras ett så kallat *dualt problem* som är definierat med samma indata som det ursprungliga, *primala* problemet. *Dualitetsteorin* är ett viktigt redskap vid utveckling av nya lösningsmetoder för olika typer av svåra optimeringsproblem.

Efter kursen ska du kunna formulera linjära optimeringsproblem matematiskt. Du ska kunna konstruera en baslösning till ett givet linjärt problem och avgöra huruvida den motsvarar en tillåten lösning och huruvida den är optimal. Vidare ska du kunna tolka stegen i simplexmetoden geometriskt.

Du ska förstå hur simplexmetoden är konstruerad, speciellt kriterierna för inkommande och utgående variabler samt för optimalitet, och kunna tillämpa algoritmen. Slutligen ska du för ett givet linjärt optimeringsproblem kunna formulera motsvarande duala problem, och känna till relationerna till det primala problemet.

1.3 Generella optimalitetsvillkor

För ett optimeringsproblem i kontinuerliga variabler kan man formulera generella och praktiskt användbara *optimalitetsvillkor*, som kan användas för att avgöra huruvida en given lösning är optimal eller ej. (För ett problem i diskreta variabler är det ofta mycket svårare att avgöra optimalitet.) *Konvexitet* är en viktig egenskap i detta sammanhang.

Efter kursen ska du kunna formulera optimalitetsvillkor för kontinuerliga optimeringsproblem, kunna använda dem för att avgöra optimalitet hos en given punkt samt känna till vilka egenskaper som krävs (främst konvexitet) för att optimalitetsvillkoren ska vara nödvändiga respektiva tillräckliga.

1.4 Problemklassificering

Kombinatoriska optimeringsproblem kan klassificeras som teoretiskt "lätta" (lösbara på *polynomisk* tid) eller "svåra" (troligtvis *inte* lösbara på polynomisk tid). Denna klassificering bygger på komplexiteten hos de algoritmer som kan användas för att lösa problemet. För att kunna visa att ett problem är teoretiskt lätt är det tillräckligt att presentera en polynomisk algoritm som löser problemet. Om man istället vill visa att ett problem är teoretiskt svårt används s k *polynomisk transformation/reduktion* från ett känt $\mathcal{NP}$-fullständigt problem.

Efter kursen ska du känna till definitionerna av problemklasserna $\mathcal{P}$, $\mathcal{NP}$ och $\mathcal{NP}$-fullständiga/$\mathcal{NP}$-hårda problem, samt förstå innebörden av klasstillhörigheten. Du ska känna till klasstillhörigheten för vissa välkända kombinatoriska optimeringsproblem. Vidare ska du för vissa problemtyper kunna visa att de är teoretiskt svåra, genom att använda polynomisk transformation.

1.5 Problem i grafer och flöde i nätverk

Graf- och nätverksproblem kan kort karaktäriseras som optimeringsproblem med underliggande grafstruktur. *Billigaste uppspännande träd-problemet*, *handelsresandeproblemet* och *det kinesiska brevbärarproblemet* är *grafproblem* med speciella strukturer, och som kan lösas med olika specialutvecklade algoritmer. *Billigaste uppspännande träd-problemet* kan lösas med metoder baserade på *girighetsprincipen*, medan *handelsresandeproblemet* är ett betydligt svårare problem, och har attackerats med flera olika lösningsmetodiker. *Minkostnadsflödesproblemet* är ett *nätverksproblem* som kan formuleras som ett linjärt optimeringsproblem med speciell bivillkorsstruktur och kan lösas med *simplexmetoden för nätverksproblem.* Vi studerar också två specialfall av minkostnadsflödesproblemet, nämligen *maxflödesproblemet* och *billigaste väg-problemet.* Dessa problem löses med effektiva algoritmer baserade på *grafsökning* och *nodmärkning. Dynamisk programmering* är en teknik som möjliggör lösning av stegindelade optimeringsproblem, ofta med tiden som steg, med hjälp av billigaste vägmetodik. Det finns även många andra grafproblem, bl.a. avseende färgning, övertäckning mm. av grafer, som tas upp begreppsmässigt.

Efter kursen ska du kunna formulera allmänna nätverksproblem som linjära optimeringsproblem och vissa grafproblem som linjära heltalsproblem. Du ska även kunna formulera vissa optimeringsproblem i graftermer.

- *Minkostnadsflödesproblemet: Du ska förstå hur simplexmetoden för nätverksproblem är konstruerad, speciellt kriterierna för inkommande och utgående variabler samt för optimalitet, och kunna tillämpa algoritmen.*
- *Maxflödesproblem: Du ska känna till och förstå innebörden av maxflödesminsnittssatsen. Vidare ska du känna till principerna för Ford-Fulkersons algoritm och kunna tillämpa den samt känna till algoritmens komplexitet.*
- *Billigaste väg-problem: Du ska känna till principerna för nodmärkningsalgoritmer (såsom Dijkstras och Fords metoder) samt kunna välja lämplig algoritm vid olika fall, exempelvis vid förekomst av negativa bågkostnader. Du ska kunna tillämpa dessa algoritmer samt känna till deras komplexitet.*
- *Billigaste uppspännande träd-problem: Du ska känna till Prims och Kruskals algoritmer och kunna tillämpa dessa. Vidare ska du känna till algoritmernas komplexitet.*
- *Handelsresandeproblem: Du skall ha kunskap om olika formuleringar av problemet samt känna till heuristiska och exakta lösningsmetoder, och kunna tillämpa dessa. Vidare ska du känna till problemets komplexitet.*
- *Kinesiska brevbärarproblemet: Du skall kunna formulera problemet samt känna till och kunna tillämpa en exakt lösningsmetod.*
- *Andra grafproblem: Du skall känna till ett antal olika grafbegrepp, såsom matchning, nod-/bågövertäckning, oberoende mängd, nod-/bågfärgning, känna till de associerade optimeringsproblemens komplexitet samt känna till heuristiker för vissa av dessa problem.*

1.6 Trädsökning och plansnittning

Trädsökning och *plansnittning* är generella angreppssätt för algoritmdesign och de används för att praktiskt lösa svåra optimeringsproblem till optimalitet, speciellt heltalsproblem. Trädsökning bygger på grafsökning, där den graf som avsöks genereras under tiden problemet löses. Noderna i grafen representerar olika delproblem, som i sig är optimeringsproblem. Båda angreppssätten arbetar iterativt med att lösa en relaxation av problemet, och därefter förbjuda oönskade lösningar.

Efter kursen ska du förstå principerna för hur trädsökning och plansnittning kan användas för att lösa optimeringsproblem. Du ska kunna specificera en trädsökningsalgoritm för ett linjärt heltalsproblem samt för ett oriktat handelsresandeproblem. Vidare ska du kunna tillämpa trädsökningsmetodik på dessa problem. Du ska även känna till hur plansnittning kan appliceras på vissa problem.

1.7 Heuristiker

Heuristiker (metoder som ej säkert ger optimum) används för att snabbt kunna generera tillåtna och förhoppningsvis bra lösningar till svåra optimeringsproblem. Inom denna klass finns bl.a. *giriga metoder* samt *lokalsökningsalgoritmer* med utvidgningar såsom *tabusökning*, *simulerad kylning* samt *genetiska algoritmer*.

Efter kursen ska du känna till vanliga principer för heuristiska metoder. Vidare ska du känna till och kunna tillämpa heuristiska algoritmer på exempelvis oriktade handelsresandeproblem. Dessutom ska du förstå de grundläggande principerna för lokalsöknings- och girighetsalgoritmer samt hur de kan tillämpas på kombinatoriska optimeringsproblem, samt hur de kan utvidgas för att inte fastna i lokala optima.

1.8 Hållbar utveckling

Optimering kan används för många olika syften, t.ex. för att maximera vinsten av en verksamhet eller minimera kostnaden för en viss produktion. Man kan även använda optimering för att minimera negativ miljöpåverkan, och därmed främja *hållbar utveckling*. I kursen försöker vi framhålla tillämpningar med det syftet. Ofta handlar det om att reflektera över vilken målfunktion man ska välja.

Efter kursen ska du känna till ett antal exempel på hur optimering kan användas för att främja hållbar utveckling och minimera skadlig miljöpåverkan.