

**Produktionsplanering av el och värme
- Matematiska modeller och metoder**

Erik Dotzauer

**Produktionsplanering av el och värme
- Matematiska modeller och metoder**

Erik Dotzauer

Forskningsrapport MDH ISt 2002:2

Institutionen för Samhällsteknik

Mälardalens Högskola

721 23 Västerås

Förord

Boken ger en introduktion till matematiska modeller och metoder för produktionsplanering i elkraft- och fjärrvärmesystem. Målgruppen är personer med erfarenhet av tekniska beräkningar men utan större kunskap i matematisk optimering.

Detta är den första utgåvan. Synpunkter på innehållet mottages tacksamt via e-post: erik.dotzauer@mdh.se.

Stockholm, November 2002

Erik Dotzauer

Innehåll

1	Inledning	1
2	Operationsanalys	3
2.1	Litteratur	5
3	Modellering av energisystem	7
3.1	Basmodeller	7
3.2	Utökade modeller	11
3.3	Litteratur	16
4	Linjärprogrammering	17
4.1	Exempel - Simplexmetoden	17
4.2	Vattenkraftplanering	19
4.3	Långtidsplanering för kraftvärmeverk	20
4.4	Litteratur	21
5	Mixed Integer Programming (MIP)	23
5.1	Exempel - Branch-and-bound	24
5.2	Korttidsplanering för kraftvärmeverk	27
5.3	Litteratur	28
6	Dynamisk programmering	29
6.1	Elproduktionsplanering	29
6.2	Värmeproduktionsplanering	32
6.3	Litteratur	33
7	Lagrangerrelaxering	35
7.1	Elproduktionsplanering	37
7.2	Värmeproduktionsplanering	38
7.3	Litteratur	40
8	Genetiska algoritmer	41
8.1	Elproduktionsplanering	41

8.2	Litteratur	43
9	Stokastisk optimering	45
9.1	Värmeproduktionsplanering	45
9.2	Litteratur	47
10	Linjära tidsserier	49
10.1	Ellastprognoser	49
10.2	Litteratur	51
11	Artificiella neurala nätverk	53
11.1	Värmelastprognoser	53
11.2	Litteratur	55
12	Modelleringspråk	57
12.1	Modellering i GAMS	57
12.2	Litteratur	58
13	Litteratur	59
13.1	Litteraturlista	59

Kapitel 1

Inledning

Dagens samhälle ställer höga krav på effektivitet. Detta gäller i högsta grad i energibranschen där ett effektivt utnyttjande av produktionsanläggningarna är önskvärt. Dagens samhälle, eller snarare dagens moderna teknik, skapar även förutsättningar för att uppnå detta. Snabbare, billigare och mer lättanvända datorer ger idag ovärderlig hjälp vid beslutsfattande och i planeringsarbeten. En förutsättning är dock att erforderlig programvara finns tillgänglig, dvs programvara i form av matematiska modeller som beskriver problematiken och algoritmer som utför beräkningsarbetet. Denna typ av planeringsarbete där man tar hjälp från matematiken benämns operationsanalys (eng. Operations Research). Området operationsanalys beskrivs i Kapitel 2.

I ett elkraftsystem finns elproducerande anläggningar och i ett fjärrvärmesystem finns anläggningar för värmeproduktion. Med kraftvärmesystem avses system som innefattar anläggningar för både el- och värmeproduktion. Produktionsplaneringen kan göras över olika tidshorisonter. Med långtidsplanering (eng. long-term planning) menas normalt planering för ett år eller längre och med mellanlång planering (eng. mid-term planning) avses planering över en tidshorisont på runt en månad. Korttidsplanering (eng. short-term planning) är planering för den närmaste veckan.

Beroende på vilken tidshorisont man beaktar är olika frågeställningar mer eller mindre relevanta. När det gäller långtidsplanering är huvudsyftet att uppskatta hur stor mängd av varje bränsle som kommer att förbrukas. Den mellanlånga planeringen motiveras bl.a. av att vissa skatter avräknas på månadsbasis. Även planering av bränslelager kräver en planeringshorisont på åtminstone några veckor. Huvudsyftet med korttidsplaneringen är att bestämma när de olika produktionsenheterna ska startas och stoppas.

Om man med datorns hjälp vill planera produktionen krävs att man kan beskriva processen med matematiska formler. Storheter och samband som ofta används presenteras i Kapitel 3. Man skiljer mellan fysikaliska modeller och black-box-modeller. I en fysikalisk modell försöker man beskriva alla fysikaliska samband som de ser ut i verk-

ligheten, dvs varje parameter i modellen har en fysikalisk motsvarighet i den verkliga processen. I en black-box-modell behandlar man den fysikaliska processen som en "svart låda". Man försöker här hitta ett matematiskt uttryck som beskriver hur systemets "insignaler" påverkar systemets "utsignaler". Parametrarna i en sådan modell har inga motsvarigheter i den verkliga processen.

Produktionsplanering av el och värme går normalt till på så sätt att man först konstruerar en lastprognos, och sedan, givet lastprognosen, konstruerar en produktionsplan. För detta behövs två typer av modeller: en lastprognosmodell och en planeringsmodell. Lastprognosmodellen är oftast en black-box-modell medan planeringsmodellen normalt är byggd som en fysikalisk modell.

Två engelska benämningar som ofta förekommer i dessa sammanhang är "the Unit Commitment Problem" och "the Economic Dispatch Problem". "Unit Commitment"-problemet avser problemet att bestämma när de olika produktionsenheterna ska startas och stoppas och "Economic Dispatch"-problemet avser problemet att bestämma enheternas produktionsnivåer givet vilka enheter som är i och ur drift. Modeller och metoder för de två problemtyperna diskuteras i Kapitel 4 - 9. Lastprognosmodeller presenteras i Kapitel 10 och 11.

Modellerna och beräkningsalgoritmerna programmeras som datorprogram. Några olika programmeringsspråk presenteras i Kapitel 12. I Kapitel 13 listas slutligen ett antal referenser.

Kapitel 2

Operationsanalys

Operationsanalysen har sitt ursprung i den militära verksamhet som bedrevs under andra världskriget där ett effektivt utnyttjande av knappa resurser var av livsavgörande betydelse. Idag har operationsanalys kommit att förknippas med ekonomisk planering trots att samma metoder kan användas för i stort sett alla former av planeringsarbete.

Det centrala inom operationsanalysen är att någon form av optimeringsberäkning utförs, dvs man beräknar fram den handlingsplan som minimerar eller maximerar en storhet, t.ex. kostnad eller tidsåtgång, givet att de villkor som beskriver processen är uppfyllda. En annan benämning som används är matematisk programmering (eng. Mathematical Programming). Gränsdragningen mellan begreppen är oskarp men en av många använd skiljelinje är att matematisk programmering innefattar optimering rent allmänt medan operationsanalys innefattar just olika typer av planeringsarbeten.

Innan beräkningsarbetet kan utföras måste frågeställningen formuleras matematiskt, dvs man måste bygga en matematisk modell av den verkliga processen. Vilka förenklingar av verkligheten som bör göras i modellen måste bedömas genom att väga önskad noggrannhet i det slutliga resultatet mot beräkningstider för den algoritm som ska användas. Många gånger är det omöjligt att göra en exakt modell, processen är helt enkelt så komplex att den inte går att beskriva i detalj på ett praktiskt användbart sätt.

Ett optimeringsproblem skrivs generellt som:

$$\begin{aligned} \min_x [f(x)] \\ \text{s.t. } x \in \Omega. \end{aligned} \tag{2.1}$$

Funktionen $f(x)$ är problemets objektfunktion. Vektorn x består av problemets beslutsvariabler, dvs de storheter som efterfrågas. Ω är mängden av tillåtna lösningar. Förkortningen *s.t.* står för "subject to" vilket är engelska och betyder "med hänsyn till". Problemet är formulerat som ett minimeringsproblem (min). Optimeringsproblemet ska utläsas som: hitta den vektor x som finns i mängden Ω och som

ger lägst värde på $f(x)$. Ett maximeringsproblem formuleras analogt men med förkortningen "max" istället för "min".

Normalt beskrivs det tillåtna området Ω med ett antal funktioner, så kallade bivillkor. Dessa kan vara både likhetsbivillkor, $g_j(x) = 0$, $j = 1, \dots, m_e$, och olikhetsbivillkor, $h_j(x) \leq 0$, $j = 1, \dots, m_i$. En alternativ formulering av det generella optimeringsproblemet blir därmed:

$$\begin{aligned} \min_x [f(x)] \\ \text{s.t. } \quad g_j(x) = 0, \quad j = 1, \dots, m_e \\ \quad \quad h_j(x) \leq 0, \quad j = 1, \dots, m_i. \end{aligned} \tag{2.2}$$

Beslutsvariablerna kan vara av typen kontinuerliga variabler eller heltalsvariabler. Kontinuerliga variabler kan anta decimalvärden medan heltalsvariabler enbart kan anta heltalsvärden. Heltalsvariabler som enbart tillåts anta värdena noll eller ett kallas binära variabler.

Om objektfunktionen $f(x)$ och alla funktioner som beskriver det tillåtna området Ω är linjära säger vi att problemet är linjärt. Om någon av dessa funktioner är icke-linjär säger vi att problemet är icke-linjärt.

Problem som enbart innefattar heltalsvariabler, dvs problem utan kontinuerliga variabler, kallas heltalsoptimeringsproblem. Om problemet innefattar både kontinuerliga variabler och heltalsvariabler talar vi om blandad heltalsoptimering.

Man skiljer även mellan deterministiska och stokastiska problem. I ett deterministiskt problem antas alla problemparametrar vara kända exakt medan man i ett stokastiskt problem låter parametrarna tillhöra en given sannolikhetsfördelning.

Om processen som ska modelleras är icke-linjär men man av någon anledning vill eller är tvungen att formulera optimeringsproblemet som linjärt finns möjligheten att beskriva de icke-linjära funktionerna som styckvis linjära. Detta kan göras enligt följande. Låt J vara antalet linjära segment på den styckvis linjära funktionen $y = f(x)$. Låt de binära variablerna u_j , $j = 1, \dots, J$, indikera vilket segment som är aktuellt. Definiera även de kontinuerliga variablerna x_j , $j = 1, \dots, J$. Segment j beskrivs av parametrarna α_j och β_j . Segmentindelningen ges av parametrarna \tilde{x}_j , $j = 0, \dots, J$. Parametrarna är valda så att

funktionen $f(x)$ är sammanhängande. Bivillkoren

$$\sum_{j=1}^J u_j = 1, \quad (2.3)$$

och

$$\tilde{x}_{j-1}u_j \leq x_j \leq \tilde{x}_j u_j, \quad j = 1, \dots, J, \quad (2.4)$$

styr vilket segment som är aktuellt. Denna konstruktion med (2.3) och (2.4) medför att endast en av variablerna x_j blir skilld från noll. Resultterande x och y ges från

$$x = \sum_{j=1}^J x_j \quad (2.5)$$

respektive

$$y = \sum_{j=1}^J (\alpha_j x_j + \beta_j u_j). \quad (2.6)$$

2.1 Litteratur

Det finns åtskilligt med litteratur inom områdena operationsanalys och matematisk programmering. En bok som ofta används i introducerande kurser är "Introduction to Mathematical Programming - Applications and Algorithms" av Winston [Win95]. Mer ingående presentationer av respektive delområden görs i: linjär optimering [BeT97], heltalsoptimering [Wol98], ickelinjär optimering [Fle87] och stokastisk optimering [BiL97].

Kapitel 3

Modellering av energisystem

I detta kapitel presenteras matematiska storheter och samband som används i modeller för produktionsplanering. Om inte annat anges så gäller definitionerna genomgående i dokumentet.

3.1 Basmodeller

Att bygga en matematisk produktionsplaneringsmodell innebär att man beskriver det aktuella systemet med ett optimeringsproblem. Objektfunktionen representerar normalt systemets totala driftskostnad. Det man här vill åstadkomma är att hitta den mest ekonomiska produktionsplanen givet att man uppfyller åtagandet att producera en viss mängd el och/eller värme. Man kan även tänka sig att låta objektfunktionen beskriva systemets påverkan på miljön. Produktionsplaneringen syftar då till att minimera mängden förorenande utsläpp.

Vi inleder här med att illustrera tekniken att modellera med två enkla produktionsplaneringsmodeller: ett "Unit Commitment"-problem för elproduktion:

$$\begin{aligned} \min_{p,u,v} & \left[\sum_{i=1}^I \sum_{k=1}^K (\alpha_{i,k} p_{i,k} + \beta_{i,k} u_{i,k}) + \sum_{i=1}^I \sum_{k=1}^K v_{i,k} \gamma_{i,k} \right] \\ \text{s.t.} & \sum_{k=1}^K p_{i,k} = p_{i,D} \\ & \underline{p}_{i,k} u_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} u_{i,k} \\ & u_{i-1,k} - u_{i,k} + v_{i,k} \geq 0 \\ & u_{0,k} = u_k^{init} \\ & u_{i,k} \in \{0, 1\} \\ & v_{i,k} \in \{0, 1\}, \end{aligned} \tag{3.1}$$

och ett ”Economic Dispatch”-problem för värmeproduktion:

$$\begin{aligned}
\min_{q,e} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0 \right) + \sum_{i=1}^I \alpha_{i,S} (q_{i,S})^2 \right] \\
s.t. & \sum_{k=1}^K q_{i,k} + q_{i,S} = q_{i,D} \\
& e_{i,S} = e_{i-1,S} - q_{i,S} \\
& \underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k} \\
& \underline{q}_{i,S} \leq q_{i,S} \leq \bar{q}_{i,S} \\
& \underline{e}_{i,S} \leq e_{i,S} \leq \bar{e}_{i,S} \\
& e_{0,S} = \underline{e}_{0,S} \\
& e_{I,S} = \underline{e}_{I,S}.
\end{aligned} \tag{3.2}$$

Den tidshorisont som beaktas delas upp i ett antal tidsintervall. Antalet intervall betecknas med bokstaven I . Tidsintervallen kan vara av varierande längd men precis som i problem (3.1) och (3.2) är det vanligt att man låter samtliga intervall vara en timme långa. Antalet produktionsenheter i respektive system betecknas med bokstaven K . Värmesystemet innefattar även en ackumulatortank, här betecknad med index S .

Problemen innefattar följande variabler:

- $p_{i,k}$: producerad el i enhet k under tidsintervall i (kontinuerlig variabel).
- $q_{i,k}$: producerad värme i enhet k under tidsintervall i (kontinuerlig variabel).
- $q_{i,S}$: uttagen effekt från ackumulatortanken under tidsintervall i (kontinuerlig variabel). Vid urladdning är $q_{i,S}$ positiv och vid laddning negativ.
- $e_{i,S}$: energimängd i ackumulatortanken vid slutet av tidsintervall i (kontinuerlig variabel).
- $u_{i,k}$: binär variabel som indikerar om enhet k under tidsintervall i är i drift ($u_{i,k} = 1$) eller ur drift ($u_{i,k} = 0$).
- $v_{i,k}$: binär variabel som indikerar om enhet k under tidsintervall i startas ($v_{i,k} = 1$) eller inte startas ($v_{i,k} = 0$).

De enskilda variablerna är samlade i vektorerna $p_k = (p_{1,k}, \dots, p_{I,k})$, $p = (p_1, \dots, p_K)$, $q_k = (q_{1,k}, \dots, q_{I,k})$, $q_S = (q_{1,S}, \dots, q_{I,S})$, $q = (q_1, \dots, q_K, q_S)$, $e = (e_{0,S}, \dots, e_{I,S})$, $u_k = (u_{0,k}, \dots, u_{I,k})$, $u = (u_1, \dots, u_K)$, $v_k = (v_{1,k}, \dots, v_{I,k})$ och $v = (v_1, \dots, v_K)$.

Problemen innefattar en mängd olika bivillkor. I båda problemen finns behovsvillkor som måste uppfyllas. I elproblemet skrivs detta som

$$\sum_{k=1}^K p_{i,k} = p_{i,D}, \quad (3.3)$$

och i värmeproblemet som

$$\sum_{k=1}^K q_{i,k} + q_{i,S} = q_{i,D}, \quad (3.4)$$

där $p_{i,D}$ är elbehovet i tidsintervall i och $q_{i,D}$ är värmebehovet i tidsintervall i .

Det finns krav på produktionsnivåerna, här beskrivet som enkla gränser för motsvarande variabel:

$$\underline{p}_{i,k} u_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} u_{i,k} \quad (3.5)$$

och

$$\underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k}. \quad (3.6)$$

Parametrarna $\underline{p}_{i,k}$ ($\underline{q}_{i,k}$) och $\bar{p}_{i,k}$ ($\bar{q}_{i,k}$) beskriver undre respektive övre gräns för produktionen. Observera att om $u_{i,k} = 0$ i (3.5) så blir $p_{i,k} = 0$. Med $u_{i,k} = 1$ erhålls motsvarande relation som i (3.6).

Akkumulatortanken i värmeproblemet beskrivs med en energibalanskvation:

$$e_{i,S} = e_{i-1,S} - q_{i,S}. \quad (3.7)$$

Laddnings och energinivån begränsas av

$$\underline{q}_{i,S} \leq q_{i,S} \leq \bar{q}_{i,S} \quad (3.8)$$

och

$$\underline{e}_{i,S} \leq e_{i,S} \leq \bar{e}_{i,S}. \quad (3.9)$$

Man antar även att initialt och slutligt energiinnehåll är känt:

$$e_{0,S} = \underline{e}_{0,S} \quad (3.10)$$

och

$$e_{I,S} = \underline{e}_{I,S}. \quad (3.11)$$

Även initial enhetsuppsättning i problem (3.1) antas vara känd:

$$u_{0,k} = u_k^{init}, \quad (3.12)$$

där parametern u_k^{init} är lika med noll om enhet k är ur drift vid tidshorisontens början och lika med ett om den är i drift. Normalt skrivs inte (3.12) ut explicit i problemet utan det är underförstått att villkoret gäller.

För att modellera startkostnader i problem (3.1) krävs logiska villkor:

$$u_{i-1,k} - u_{i,k} + v_{i,k} \geq 0. \quad (3.13)$$

Startkostnaden ges av

$$c_{i,k}^{start} = v_{i,k} \gamma_{i,k}, \quad (3.14)$$

där $\gamma_{i,k} > 0$ är kostnaden för att starta enhet k i tidsintervall i . Studeras (3.13) och (3.14) närmare inses att den binära variabeln $v_{i,k}$ antar värdet ett endast när $u_{i-1,k} = 0$ och $u_{i,k} = 1$, dvs (3.14) ger ett positivt tillskott till objektfunktionen bara om enhet k startas i tidsintervall i .

Produktionskostnaden för en elproducerande enhet k i tidsintervall i är

$$c_{i,k}^{prod} = \alpha_{i,k} p_{i,k} + \beta_{i,k} u_{i,k}, \quad (3.15)$$

och produktionskostnaden för en värmeproducerande enhet är

$$c_{i,k}^{prod} = \alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0. \quad (3.16)$$

Parametrarna $\beta_{i,k}$ och $\alpha_{i,k}^0$ beskriver de fasta kostnader som uppstår när enheterna är i drift och $\alpha_{i,k}$, $\alpha_{i,k}^1$ och $\alpha_{i,k}^2$ beskriver de kostnader som beror av produktionsnivån. Kostnaden för ackumulatortanken beskrivs som

$$c_{i,S} = \alpha_{i,S} (q_{i,S})^2. \quad (3.17)$$

Vi sammanfattar här definitionerna av de två modellerna med att konstatera att problem (3.1) är ett linjärt problem med både heltalsvariabler (binära variabler) och kontinuerliga variabler. Problem (3.2) är ett icke-linjärt problem med enbart kontinuerliga variabler.

3.2 Utökade modeller

Problem (3.1) och (3.2) är två enkla modeller. Dessa kan nu modifieras och byggas ut med fler variabler och bivillkor beroende på utseendet hos det system som ska modelleras. Några vanliga modelleringskomponenter presenteras i det resterande av detta kapitel.

Tidsberoende startkostnader. En enhet som varit avställd en längre tid är avkyld och därmed dyrare att starta än om den fortfarande varit varm. Låt parametern $\tau_k^{cold} > 0$ beteckna avkylningstiden för enhet k . Om vi i modellen endast vill modellera kallstart och varmstart används två binära variabler, $v_{i,k}^{cold}$ och $v_{i,k}^{warm}$, som indikerar om enhet k i tidsintervall i gör en kallstart respektive varmstart. Motsvarande startkostnad är:

$$c_{i,k}^{start} = \gamma_{i,k}^{warm} v_{i,k}^{warm} + \gamma_{i,k}^{cold} v_{i,k}^{cold}, \quad (3.18)$$

där $\gamma_{i,k}^{warm} > 0$ betecknar varmstartkostnaden och $\gamma_{i,k}^{cold} > 0$ betecknar motsvarande extrakostnad för kallstart (dvs den totala kallstartkostnaden är $\gamma_{i,k}^{warm} + \gamma_{i,k}^{cold}$). Det krävs även att följande bivillkor läggs till i modellen:

$$\Omega_{i,k}^{start} = \begin{cases} u_{i-1,k} - u_{i,k} + v_{i,k}^{warm} \geq 0 \\ \sum_{j=i-\tau_k^{cold}}^{i-1} u_{j,k} - u_{i,k} + v_{i,k}^{cold} \geq 0. \end{cases} \quad (3.19)$$

Vi noterar att den tidsberoende startkostnaden definierad av (3.18) och (3.19) är linjär. Ickelinjära startkostnader beskrivs normalt som:

$$c_{i,k}^{start} = \gamma_{i,k}^1 + \gamma_{i,k}^2 (1 - e^{-\gamma_{i,k}^3 \tau_{i,k}}), \quad (3.20)$$

där parametrarna $\gamma_{i,k}^j$ är positiva, dvs $\gamma_{i,k}^j > 0$, $j = 1, 2, 3$, och $\tau_{i,k}$ betecknar den tid enheten varit avställd när den startas.

Stoppkostnader. På motsvarande sätt som startkostnaderna modelleras kan även de kostnader som uppstår när en enhet tas ur drift beskrivas. Vi illustrerar detta med en stoppkostnad som är konstant och given som $\gamma_{i,k}^{stop} > 0$. Låt $w_{i,k}$ vara en binär variabel som indikerar om enhet k under tidsintervall i stoppas ($w_{i,k} = 1$) eller inte stoppas ($w_{i,k} = 0$). Lägg till följande bivillkor:

$$u_{i-1,k} - u_{i,k} - w_{i,k} \leq 0, \quad (3.21)$$

samt addera följande kostnad till objektfunktionen:

$$C_{i,k}^{stop} = w_{i,k} \gamma_{i,k}^{stop}. \quad (3.22)$$

Minsta upptid och minsta nertid. Minsta upptid är den tid en produktionsenhet måste vara i drift när den startats och minsta nertid är den tid en enhet måste vara ur drift när den stoppats. Minsta upptid för enhet k betecknas $\underline{\tau}_k^{up} \geq 0$. Minsta nertid betecknas $\underline{\tau}_k^{down} \geq 0$. Tiderna uppfylles med hjälp av följande bivillkor:

$$\Omega_{i,k}^{time} = \begin{cases} u_{j,k} - u_{j-1,k} \leq u_{i,k}, & j = i - \underline{\tau}_k^{up} + 1, \dots, i - 1 \\ u_{j-1,k} - u_{j,k} \leq 1 - u_{i,k}, & j = i - \underline{\tau}_k^{down} + 1, \dots, i - 1. \end{cases} \quad (3.23)$$

Rampvillkor. På grund av ökat slitage vill man normalt inte göra snabba förändringar av produktionsnivån hos en enhet. Det finns även fysiska begränsningar för hur mycket man kan förändra produktionen. Detta modelleras med rampvillkor:

$$\underline{\delta}_k \leq q_{i,k} - q_{i-1,k} \leq \bar{\delta}_k. \quad (3.24)$$

Parametrarna $\underline{\delta}_k$ och $\bar{\delta}_k$ beskriver hur mycket enhet k tillåts minska respektive öka sin produktion från timme till timme. Villkoret gäller enbart när enheten är i drift under både tidsintervall $i - 1$ och i .

Akkumulatorförluster. Förluster i en akkumulator tank modelleras på så sätt att ekvation (3.7) ersätts med:

$$e_{i,S} = (1 - l_{i,S})e_{i-1,S} - q_{i,S} - l'_{i,S}, \quad (3.25)$$

där parametrarna $l_{i,S}$ och $l'_{i,S}$ beskriver förlusterna.

Reservbivillkor. Ett energibolag som åttagit sig att producera en viss mängd el vill ofta ha extra produktionskapacitet redo ifall någon produktionsenhet plötsligt skulle falla ur eller om lasten hastigt skulle öka. Detta modelleras med:

$$\sum_{k=1}^K \bar{p}_{i,k} u_{i,k} \geq p_{i,R}, \quad (3.26)$$

där $p_{i,R}$ beskriver reservbehovet i tidsintervall i .

Modellering av kraftvärmeenhet. En kraftvärmeenhet kan bestå av en eller flera pannor och turbiner. Dessa komponenter kan modelleras var och en för sig för att sedan kopplas samman med ett antal

ångbalanser. Tekniken illustreras på en enhet med en panna och en turbin. Antag att man kan elda både olja och biobränsle (t.ex. pellets) i pannan. Detta producerar ångeffekterna $s_{i,k}^{oil}$ och $s_{i,k}^{bio}$. Ångproduktionen i pannan begränsas av:

$$\underline{s}_{i,k} \leq s_{i,k}^{oil} + s_{i,k}^{bio} \leq \bar{s}_{i,k}. \quad (3.27)$$

Ångan leds till turbinen för mottrycksproduktion (MT) av el och värme alternativt direkt till en värmeväxlare för produktion av enbart värme (direktvärme, förkortat DV). Detta innebär att sex olika produkter produceras under varje tidsintervall i , vilka representeras med följande variabler:

- $p_{i,k}^{oil}$ och $p_{i,k}^{bio}$: el från olja respektive biobränsle.
- $q_{i,k}^{oil,MT}$ och $q_{i,k}^{bio,MT}$: mottrycksvärme från olja respektive biobränsle.
- $q_{i,k}^{oil,DV}$ och $q_{i,k}^{bio,DV}$: direktvärme från olja respektive biobränsle.

För var och en av de sex variablerna finns enkla gränser av typen (3.6).

Relationen mellan producerad mottrycksel och mottrycksvärme beskrivs med:

$$p_{i,k}^{oil} + p_{i,k}^{bio} = \rho_{i,k}(q_{i,k}^{oil,MT} + q_{i,k}^{bio,MT}), \quad (3.28)$$

där $\rho_{i,k}$ representerar det så kallade alfavärdet. Variablerna kopplas samman med en ångbalans för vardera olja och biobränsle:

$$s_{i,k}^{oil} = p_{i,k}^{oil} + q_{i,k}^{oil,MT} + q_{i,k}^{oil,DV} \quad (3.29)$$

och

$$s_{i,k}^{bio} = p_{i,k}^{bio} + q_{i,k}^{bio,MT} + q_{i,k}^{bio,DV}. \quad (3.30)$$

Kraftvärmeenheten genererar en kostnad:

$$\begin{aligned} c_{i,k} = & \alpha_{i,k}^{oil} s_{i,k}^{oil} + \alpha_{i,k}^{bio} s_{i,k}^{bio} + \alpha_{i,k}^{oil,MT,tax} q_{i,k}^{oil,MT} \\ & + \alpha_{i,k}^{bio,MT,tax} q_{i,k}^{bio,MT} + \alpha_{i,k}^{oil,DV,tax} q_{i,k}^{oil,DV} \\ & + \alpha_{i,k}^{bio,DV,tax} q_{i,k}^{bio,DV} - \beta_{i,k}^{oil} p_{i,k}^{oil} - \beta_{i,k}^{bio} p_{i,k}^{bio}, \end{aligned} \quad (3.31)$$

där $\alpha_{i,k}^{oil}$ och $\alpha_{i,k}^{bio}$ är bränslekostnaderna, $\alpha_{i,k}^{oil,MT,tax}$, $\alpha_{i,k}^{bio,MT,tax}$, $\alpha_{i,k}^{oil,DV,tax}$ och $\alpha_{i,k}^{bio,DV,tax}$ är gällande skattesatser, och slutligen, $\beta_{i,k}^{oil}$

och $\beta_{i,k}^{bio}$ är intäkten för såld el. Observera att intäkten för producerad värme inte modelleras i ekvation (3.31). Värmeproduktion framtvings istället med ett behovsvillkor av typ (3.4).

Modellen kan utökas ytterligare, t.ex. kan variabler för avtappning av hjälpånga läggas till i ångbalanserna (3.29) och (3.30).

Hjälpel. Anläggningar som producerar el och/eller värme konsumerar en hel del el i pumpar, fläktar etc. Hjälpsförbrukningen för enhet k under tidsintervall i betecknas $p_{i,k}^{help}$. Förbrukningen består dels av en konstant del $\alpha_{i,k}^{0,help}$ och en del $\alpha_{i,k}^{1,help}$ som beror av produktionsnivån. För ett värmeverk modelleras detta med:

$$p_{i,k}^{help} = \alpha_{i,k}^{1,help} q_{i,k} + \alpha_{i,k}^{0,help} u_{i,k}. \quad (3.32)$$

Förbrukningen belastar även objektfunktionen med en kostnad:

$$c_{i,k}^{help} = \beta_{i,k}^{help} p_{i,k}^{help}, \quad (3.33)$$

där $\beta_{i,k}^{help}$ är motsvarande kostnadsparameter.

Modellering av fjärrvärmenät. Så som problem (3.2) är formulerat förbrukas värmen under samma timme som den produceras. Detta kan antas gälla för små fjärrvärmesystem. För stora system med stor utbredning kan tidsfördröjningarna i nätet vara betydande. Tidsfördröjningar, eventuella begränsningar i överföringskapacitet i nätet samt det faktum att värmekunderna och produktionsanläggningarna är spridda i nätet kan hanteras genom att modellera distributionsnätet som ett nätverk. Anläggningarna och kunderna är lokaliserade i nätverkets noder och bågarna representerar möjliga distributionsvägar.

Låt N beteckna antalet noder och låt variabeln $e_i^{(n,m)}$ beskriva den energimängd som flyttas från nod n till nod m under tidsintervall i . Vi konstruerar nätverket så att det tar τ_i timmar att flytta energin. Energinivåerna i början och i slutet av tidshorisonten, $e_0^{(n,m)}$ och $e_I^{(n,m)}$, antas vara kända. Eventuella överföringsbegränsningar hos båge (n,m) modelleras med $\bar{e}_i^{(n,m)}$. Variabeln $q_{i,T}^n$ med tillhörande begränsningar får representera effektutbytet som nod n har med det resterande nätverket.

Låt $\eta_i^{(m,n)}$ beteckna förlusterna hos båge (m,n) under tidsintervall

i och definiera en energibalans ekvation för varje nod n :

$$q_{i,T}^n \tau_i + \sum_{m=1}^N e_i^{(n,m)} - \sum_{m=1}^N \eta_i^{(m,n)} e_{i-1}^{(m,n)} = 0. \quad (3.34)$$

Låt $K(n)$ vara mängden av produktionsenheter som är lokaliserade i nod n och låt parametern $q_{i,D}^n$ beteckna värmebehovet i nod n under tidsintervall i . Sammanfattningsvis erhålls följande optimeringsproblem:

$$\begin{aligned} \min_{q,e} & \left[\sum_{i=1}^I \sum_{n=1}^N \sum_{k \in K(n)} \left(\alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0 \right) \tau_i \right] \\ \text{s.t.} & \quad q_{i,T}^n \tau_i + \sum_{m=1}^N e_i^{(n,m)} - \sum_{m=1}^N \eta_i^{(m,n)} e_{i-1}^{(m,n)} = 0 \\ & \quad \sum_{k \in K(n)} q_{i,k} + q_{i,T}^n = q_{i,D}^n \\ & \quad \underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k} \\ & \quad \underline{q}_{i,T}^n \leq q_{i,T}^n \leq \bar{q}_{i,T}^n \\ & \quad 0 \leq e_i^{(n,m)} \leq \bar{e}_i^{(n,m)} \\ & \quad e_0^{(n,m)} = \underline{e}_0^{(n,m)} \\ & \quad e_I^{(n,m)} = \underline{e}_I^{(n,m)}. \end{aligned} \quad (3.35)$$

De ekvationer som beskriver nätverket i problem (3.35) är alla linjära. Vill man modellera ett fjärrvärmenät som icke linjärt används vattentemperatur och vattenflöde som variabler istället för effekt och energi (dvs istället för q och e). På motsvarande sätt som problem (3.35) beskriver ett fjärrvärmenät kan också kraftnät modelleras.

Bränslelager och begränsningar i tillgången på bränsle. Vill man modellera t.ex. ett oljelager görs detta enligt följande. Låt variabeln f_i^{oil} beteckna antalet kubikmeter olja som finns i lagret i slutet av tidsintervall i . Lagerinnehållet begränsas av:

$$\underline{f}_i^{oil} \leq f_i^{oil} \leq \bar{f}_i^{oil}. \quad (3.36)$$

Den mängd olja (i kubikmeter) som levereras till lagret under tidsintervall i representeras av parametern $\varphi_i^{oil,in}$. Låt K^{oil} vara mängden av produktionsenheter som använder olja som bränsle och låt ρ_k^{oil}

beteckna den relativa oljeförbrukningen för enhet k . Definiera slutligen lagervillkoret:

$$f_i^{oil} = f_{i-1}^{oil} + \varphi_i^{oil,in} - \sum_{k \in K^{oil}} \rho_k^{oil} q_{i,k}. \quad (3.37)$$

Ett alternativt sätt att hantera tillgången på bränsle är att sätta en begränsning för hur stor volym som får användas under en viss period. Till exempel, antag att man tillåts att förbruka maximalt \hat{f}^{oil} kubikmeter olja under de första J tidsintervallen. Detta ger bivillkoret:

$$\sum_{i=1}^J \sum_{k \in K^{oil}} \rho_k^{oil} q_{i,k} \leq \hat{f}^{oil}. \quad (3.38)$$

Logiska villkor. Med de binära variablerna kan logiska villkor formuleras. Till exempel, ett krav av typen ”om enhet k är i drift måste även enhet m vara i drift” skrivs:

$$u_{i,k} \leq u_{i,m}. \quad (3.39)$$

Kravet ”åtminstone en av enheterna k eller m måste vara i drift” skrivs:

$$u_{i,k} + u_{i,m} \geq 1. \quad (3.40)$$

Andra exempel är ”enhet k och m kan inte vara i drift samtidigt”:

$$u_{i,k} + u_{i,m} \leq 1, \quad (3.41)$$

och ”maximalt två starter tillåts under samma timme”:

$$\sum_{k=1}^K v_{i,k} \leq 2. \quad (3.42)$$

3.3 Litteratur

Modeller för produktionsplanering av el och värme finns väl beskrivna i litteraturen. Doktorsavhandlingar innehåller ofta många referenser: [Arv01], [Dot01], [Eri94], [Nil97] och [Zha95]. Även artikelsamlingar, t.ex. IEEE Transactions on Power Systems, beskriver denna typ av problem. ”Unit Commitment”-problemet diskuteras i [SeK98] och [ShF94], och ”Economic Dispatch”-problemet i [ChR90]. Fredriksen och Werner [FrW93] presenterar fjärrvärmetekniken.

Kapitel 4

Linjärprogrammering

Linjärprogrammering (eng. Linear Programming, förkortat LP) avser linjära optimeringsproblem med enbart kontinuerliga variabler. Ett LP-problem skrivs generellt:

$$\begin{aligned} \min_x & [c^T x] \\ \text{s.t.} & Ax = b \\ & x \geq 0. \end{aligned} \tag{4.1}$$

Vektorn c tillsammans med de kontinuerliga variablerna x definierar problemets objektfunktion. Matrisen A och vektorerna x och b beskriver problemets bivillkor. Problem (4.1) är skrivet på standardform vilket innebär att problemet är ett minimeringsproblem med enbart likhetsbivillkor och icke-negativa variabler. Alla LP-problem, även maximeringsproblem och de som innehåller olikhetsbivillkor, kan skrivas på standardform.

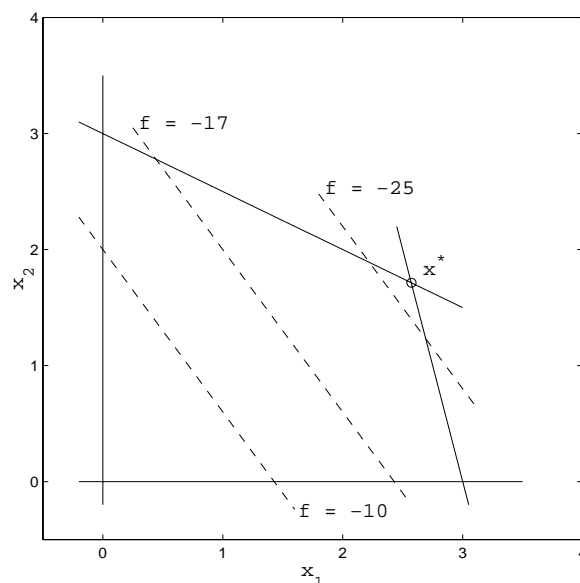
Simplexmetoden är den mest kända lösningsmetoden för LP-problem. Principen bakom metoden beskrivs enklast med ett exempel:

4.1 Exempel - Simplexmetoden

Betrakta följande LP-problem:

$$\begin{aligned} \min_x & [-7x_1 - 5x_2] \\ \text{s.t.} & x_1 + 2x_2 \leq 6 \\ & 4x_1 + x_2 \leq 12 \\ & x_1 \geq 0 \\ & x_2 \geq 0. \end{aligned} \tag{4.2}$$

Eftersom problemet endast har två variabler kan det illustreras grafiskt, se Figur 4.1. Bivillkoren är utritade som heldragna linjer. Det tillåtna området är den yta som innesluts av bivillkoren. Objektfunktionen tänker vi oss som ett lutande plan som skär igenom det tillåtna området (genom papperet). De streckade linjerna visar några nivåkurvor $f =$



Figur 4.1. LP-exempel. De heldragna linjerna visar bivillkoren och de streckade linjerna nivåkurvor för objektfunktionen. Det tillåtna området är den yta som innesluts av bivillkoren. Optimal lösning x^* är markerad med en ring.

$-7x_1 - 5x_2$. Beroende på objektfunktionens koefficienter (i detta fall $c_1 = -7$ och $c_2 = -5$), dvs nivåkurvornas lutning i x-y-planet, inses att optimal lösning till problemet måste ligga i ett ”hörn” av det tillåtna området. Den optimala lösningen $x^* = (\frac{18}{7}, \frac{12}{7})$ till vårt problem är markerad med en ring.

Det faktum att optimal lösning måste ligga i ett hörn utnyttjas i Simplexmetoden. Beräkningarna går till på så sätt att man startar i ett godtyckligt hörn och sedan hoppar från hörn till hörn tills man nått optimum. Hoppen görs endast till närliggande hörn som har lägre (högre vid maximering) objektfunktionsvärde än hörnet man står i, dvs man hoppar i en riktning där objektfunktionsplanet ”lutar neråt”. När man kommit till ett hörn där planet ”lutar uppåt” i alla riktningar har man hittat optimal lösning till problemet.

Tankesättet ovan kan generaliseras och gäller således även för problem med fler än två variabler. I praktiken kan en del svårigheter uppstå. Om man t.ex. i ett problem med n variabler kommer till ett hörn som begränsas av fler än n bivillkor finns risken att Simplexmetoden fastnar i det aktuella hörnet. Det finns dock metoder som hanterar

denna typ av problematik.

4.2 Vattenkraftplanering

Vattenkraftstationer som ligger längs samma älv påverkar varandra. Vatten som leds genom eller förbi en station når förr eller senare stationerna nedströms. Låt variabeln $x_{i,k}$ beskriva innehållet i vattenmagasinet (i kubikmeter) vid station k i slutet av timme i . Variablerna $a_{i,k}$ och $s_{i,k}$ beskriver tappningen genom respektive spillet förbi station k under timme i . Tillrinningen betecknas av parametern $\omega_{i,k}$.

Låt parametern $\tau(m,k)$ beteckna den tid (i hela timmar) det tar för vattnet att rinna från station m till närmast nedströms liggande station k . En vattenbalansekvation kan nu formuleras för varje station och timme:

$$x_{i,k} = x_{i-1,k} - a_{i,k} - s_{i,k} + a_{i-\tau(m,k),m} + s_{i-\tau(m,k),m} + \omega_{i,k}. \quad (4.3)$$

Låt variabeln $h_{i,k}$ beteckna magasinshöjden. Nivån tillåts variera mellan:

$$\underline{h}_{i,k} \leq h_{i,k} \leq \bar{h}_{i,k}. \quad (4.4)$$

Givet den undre gränsen $\underline{h}_{i,k}$ kan magasinshöjden beräknas med:

$$x_{i,k} = \beta_{i,k} (h_{i,k} - \underline{h}_{i,k}). \quad (4.5)$$

Den el $p_{i,k}$ som produceras i station k under timme i antas vara proportionell mot mängden vatten som leds genom stationen, dvs:

$$p_{i,k} = \rho_{i,k} a_{i,k}. \quad (4.6)$$

Om vi antar linjära produktionskostnader och kompletterar med

randvillkor och enkla gränser för variablerna erhålls följande LP-modell:

$$\begin{aligned}
\min_{p,x,a,s,h} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^1 p_{i,k} + \alpha_{i,k}^0 \right) \right] \\
s.t. & \sum_{k=1}^K p_{i,k} = p_{i,D} \\
& x_{i,k} = x_{i-1,k} - a_{i,k} - s_{i,k} + a_{i-\tau(m,k),m} + s_{i-\tau(m,k),m} + \omega_{i,k} \\
& x_{i,k} = \beta_{i,k} (h_{i,k} - \underline{h}_{i,k}) \\
& p_{i,k} = \rho_{i,k} a_{i,k} \\
& \underline{p}_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} \\
& \underline{x}_{i,k} \leq x_{i,k} \leq \bar{x}_{i,k} \\
& \underline{a}_{i,k} \leq a_{i,k} \leq \bar{a}_{i,k} \\
& \underline{s}_{i,k} \leq s_{i,k} \leq \bar{s}_{i,k} \\
& \underline{h}_{i,k} \leq h_{i,k} \leq \bar{h}_{i,k} \\
& x_{0,k} = \underline{x}_{0,k} \\
& x_{I,k} = \underline{x}_{I,k},
\end{aligned} \tag{4.7}$$

där $p_{i,D}$ är den mängd el som måste produceras under timme i .

Problem (4.7) ger en förenklad bild av verkligheten. Till exempel, parametern $\beta_{i,k}$ som beskriver förhållandet mellan vattennivå och magasininnehåll antas här vara konstant, dvs man antar att vattenytans area är konstant. I verkligheten ökar normalt arean när vattennivån ökar. En annan förenkling som gjorts är att producerad el $p_{i,k}$ i (4.6) enbart beror av vattenflödet $a_{i,k}$ som leds genom stationen. Normalt påverkas produktionen också av vattennivån $h_{i,k}$. För att bli användbar i praktiken bör modellen även modifieras till att hantera rinntider $\tau(m,k)$ som inte behöver anges i hela timmar.

4.3 Långtidsplanering för kraftvärmeverk

Ett syfte med långtidsplanering för kraftvärmeverk är att uppskatta hur stor mängd av varje bränsle som kommer att förbrukas det kommande året. Detta är viktig information för bl.a. budgetarbetet och bränslehandeln. Planering över tidshorisonter på ett år eller längre baseras på relativt osäkra indata. Till exempel vet man inte de exakta bränslepriserna och värmelasten beskrivs ofta med någon form av normalårskurva. Med så pass osäkra indata är det inte säkert att planeringen blir bättre bara för att man använder en modell med heltalsvari-

abler eller med olinjära samband. En LP-modell kan många gånger ge minst lika relevanta resultat.

4.4 Litteratur

I boken [BeT97] diskuterar Bertsimas och Tsitsiklis linjärprogrammering. Vattenkraftmodeller behandlas av Olof Nilsson [Nil97].

Kapitel 5

Mixed Integer Programming (MIP)

Ett linjärt optimeringsproblem med både kontinuerliga variabler och heltalsvariabler benämns på engelska Mixed Integer Linear Program (MILP). Något felaktigt brukar sådana problem även kallas Mixed Integer Program (MIP), dvs när man talar om MIP-problem är det underförstått att problemen är linjära.

Ett MIP-problem skrivs generellt:

$$\begin{aligned} \min_{x,y} [c^T x + e^T y] \\ \text{s.t. } Ax + Dy = b \\ x \geq 0 \\ y \geq 0 \text{ och heltal.} \end{aligned} \tag{5.1}$$

Vektorerna c och e tillsammans med de kontinuerliga variablerna x och heltalsvariablerna y beskriver objektfunktionen. Matriserna A och D samt vektorerna x , y och b beskriver bivillkoren. Precis som LP-problemet (4.1) är MIP-problemet (5.1) skrivet på standardform.

Branch-and-bound är en metod som ofta används för att lösa generella MIP-problem. Tekniken är att successivt tvinga fler och fler av heltalsvariablerna till nya värden. Detta genererar en trädstruktur där man i varje gren (branch) lägger till ytterligare restriktioner på någon av heltalsvariablerna. I varje gren kan man även beräkna en motsvarande undre gräns (bound) för den optimala lösningen. Ett sätt att hitta en undre gräns är att använda LP-relaxering, vilket innebär att gränsen definieras som optimalt objektfunktionsvärde till det LP-problem man erhåller då heltalskraven i MIP-problemet slopas.

Vi illustrerar Branch-and-bound-metoden på ett problem med kontinuerliga och binära variabler:

5.1 Exempel - Branch-and-bound

Betrakta följande MIP-problem:

$$\begin{aligned}
 \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\
 \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\
 & \quad 7x_3 - 3x_5 \leq 3 \\
 & \quad x_1 \in \{0, 1\}, x_2 \in \{0, 1\}, x_3 \in \{0, 1\} \\
 & \quad x_4 \geq 0, x_5 \geq 0.
 \end{aligned} \tag{5.2}$$

Problemet har fem variabler varav tre är binära (x_1 , x_2 och x_3) och två är kontinuerliga (x_4 och x_5). Vi löser först det LP-relaxerade problemet (med t.ex. Simplexmetoden, se Kapitel 4):

$$\begin{aligned}
 \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\
 \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\
 & \quad 7x_3 - 3x_5 \leq 3 \\
 & \quad 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, 0 \leq x_3 \leq 1 \\
 & \quad x_4 \geq 0, x_5 \geq 0.
 \end{aligned} \tag{5.3}$$

Den optimala lösningen till (5.3) är $x_{LP}^{(1)} = (0.23, 1, 0.43, 0, 0)$ och motsvarande optimalt objektfunktionsvärde är $f_{LP}^{(1)} = 6.73$. Objektfunktionsvärdet $f_{LP}^{(1)}$ är en undre gräns för lösningen till problem (5.2), dvs vi vet nu att optimalt objektfunktionsvärde i (5.2) inte är lägre än 6.73. Vi ser att x_2 antar ett heltalsvärde men att x_1 och x_3 antar decimalvärden. Vi måste alltså förgrena på antingen x_1 eller x_3 . Vi väljer x_3 . Detta ger två nya problem:

$$\begin{aligned}
 \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\
 \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\
 & \quad 7x_3 - 3x_5 \leq 3 \\
 & \quad 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, x_3 = 0 \\
 & \quad x_4 \geq 0, x_5 \geq 0,
 \end{aligned} \tag{5.4}$$

och

$$\begin{aligned}
 \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\
 \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\
 & \quad 7x_3 - 3x_5 \leq 3 \\
 & \quad 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, x_3 = 1 \\
 & \quad x_4 \geq 0, x_5 \geq 0.
 \end{aligned} \tag{5.5}$$

Vi väljer att lösa problem (5.4) först, vilket ger den optimala lösningen $x_{LP}^{(2)} = (0.87, 1, 0, 0, 0)$ och $f_{LP}^{(2)} = 7.37$. Variabel x_1 är fortfarande ett decimaltal så vi gör ytterligare en förgrening. Detta skapar således två nya problem:

$$\begin{aligned} \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\ \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\ & \quad 7x_3 - 3x_5 \leq 3 \\ & \quad x_1 = 0, \quad 0 \leq x_2 \leq 1, \quad x_3 = 0 \\ & \quad x_4 \geq 0, \quad x_5 \geq 0, \end{aligned} \tag{5.6}$$

och

$$\begin{aligned} \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\ \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\ & \quad 7x_3 - 3x_5 \leq 3 \\ & \quad x_1 = 1, \quad 0 \leq x_2 \leq 1, \quad x_3 = 0 \\ & \quad x_4 \geq 0, \quad x_5 \geq 0. \end{aligned} \tag{5.7}$$

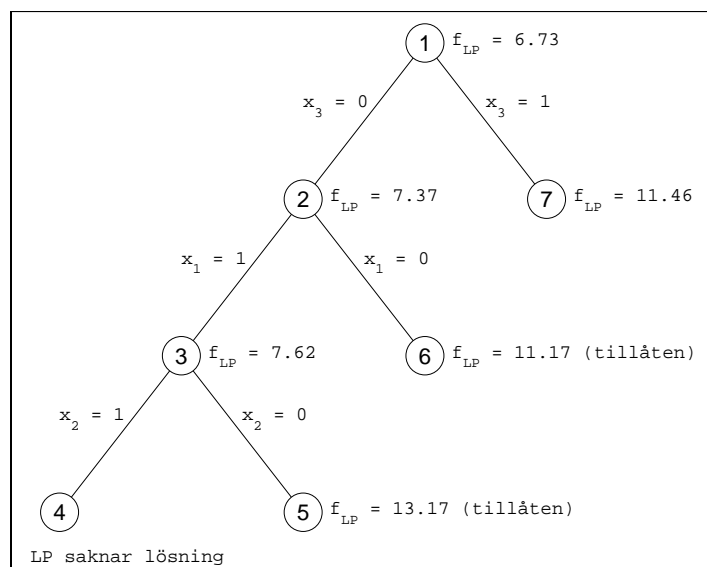
Vi väljer nu att lösa problem (5.7) och erhåller $x_{LP}^{(3)} = (1, 0.87, 0, 0, 0)$ och $f_{LP}^{(3)} = 7.62$. Eftersom x_2 fortfarande är ett decimaltal görs ytterligare en förgrening, vilket ger:

$$\begin{aligned} \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\ \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\ & \quad 7x_3 - 3x_5 \leq 3 \\ & \quad x_1 = 1, \quad x_2 = 0, \quad x_3 = 0 \\ & \quad x_4 \geq 0, \quad x_5 \geq 0, \end{aligned} \tag{5.8}$$

och

$$\begin{aligned} \min_x & [5x_1 + 3x_2 + 6x_3 + 7x_4 + 4x_5] \\ \text{s.t.} & \quad 4x_1 + 4x_2 + 6x_3 + 3x_4 + x_5 = 7.5 \\ & \quad 7x_3 - 3x_5 \leq 3 \\ & \quad x_1 = 1, \quad x_2 = 1, \quad x_3 = 0 \\ & \quad x_4 \geq 0, \quad x_5 \geq 0. \end{aligned} \tag{5.9}$$

Vi försöker nu lösa (5.9) men inser snart att problemet saknar lösning. Vi löser istället problem (5.8) och erhåller $x_{LP}^{(5)} = (1, 0, 0, 1.17, 0)$ och $f_{LP}^{(5)} = 13.17$. Eftersom x_1 , x_2 och x_3 antar heltalsvärden innebär det att vi hittat en tillåten lösning till problem (5.2). Frågan är om $x_{LP}^{(5)}$ är



Figur 5.1. Branch-and-bound-träd. Noderna är numrerade i den ordning som algoritmen har stegat igenom trädet. Ett LP-relaxerat problem löstes i varje nod. Första tillåtna lösning till ursprungsproblemet hittades i nod fem. Optimal lösning hittades i nod sex. I nod fyra saknar det relaxerade problemet tillåtna lösningar.

den optimala lösningen. För att ta reda på det låter vi algoritmen jobba vidare med att lösa problem (5.6), vilket ger $x_{LP}^{(6)} = (0, 1, 0, 1.17, 0)$ och $f_{LP}^{(6)} = 11.17$. Vi konstaterar här att x_1, x_2 och x_3 antar heltalsvärden och att $f_{LP}^{(6)} < f_{LP}^{(5)}$ gäller, dvs $x_{LP}^{(6)}$ är en bättre lösning till (5.2). Eftersom problem (5.6) gav heltalsvärden för samtliga relaxerade variabler finns det ingen anledning att förgrena vidare på x_2 . Vi löser istället problem (5.5) vilket ger $x_{LP}^{(7)} = (0, 0.04, 1, 0, 1.33)$ och $f_{LP}^{(7)} = 11.46$. Här är x_2 ett decimaltal men eftersom $f_{LP}^{(7)} > f_{LP}^{(6)}$ kommer ytterligare förgreningar från problem (5.5) inte att ge några lösningar med funktionsvärden lägre än $f_{LP}^{(6)}$. Det finns nu inga problem kvar att lösa vilket bevisar att $x_{LP}^{(6)} = (0, 1, 0, 1.17, 0)$ är optimal lösning till problem (5.2).

Beräkningsgången illustreras i Figur 5.1 med ett så kallat Branch-and-bound-träd. Noderna i trädet representerar de olika LP-problemen och bågarna representerar de variabler man förgrenar på. Notera att

optimalt objektfunktionsvärde till det relaxerade problemet ökar när man går djupare ner i trädet.

Det finns flera varianter av Branch-and-bound-metoden, t.ex. finns olika metoder för hur man ska välja förgreningsvariabler och för hur man ska söka igenom trädet. Exempel på det senare är djupsökning och breddsökning. Vid djupsökning söker man sig snabbt nedåt i trädet medan breddsökning innebär att man löser alla problem på en nivå i trädet innan man förflyttar sig ner till nästa nivå.

Metoden som används i exemplet kan generaliseras till att även hantera generella heltalsvariabler.

Ett problem med Branch-and-bound-metoden enligt ovan är att problemet som ska lösas behandlas generellt, dvs lösningsmetoden utnyttjar inte problemets struktur. Vid praktiska tillämpningar kan det resulterande MIP-problemet bli så stort att det tar oacceptabelt lång tid att lösa med denna metod. Detta motiverar användandet av optimeringsmetoder som är anpassade till det aktuella problemets struktur. En fördel med metoden är att man kan avbryta beräkningarna och nöja sig med den dittills bästa lösning man hittat, en lösning som många gånger kan vara tillräckligt bra.

5.2 Korttidsplanering för kraftvärmeverk

Ett kraftvärmeverk är en komplicerad process som är svår att överblicka. Användandet av datormodeller blir därmed i princip ett måste i planeringen. För att hålla nere beräkningstiderna bör modellerna vara relativt enkla, samtidigt som de måste vara så pass detaljerade att de ger relevanta produktionsplaner. För korttidsplanering av kraftvärmesystem har det visat sig att MIP ofta uppfyller båda dessa krav. Möjligheten att modellera pannor och turbiner, hjälpelsförbrukning, bränslelager osv. på ett relativt enkelt och överskådligt sätt (se Kapitel 3) talar för metodiken. Dessutom finns kommersiella optimeringslösare för problemtypen, dvs modelleringsarbetet kan koncentreras till själva modellbyggandet och inte på algoritmutveckling. Nackdelen är att de resulterande optimeringsproblemen ofta tenderar att bli stora, vilket kan medföra oacceptabelt långa beräkningstider för standardlösarna.

5.3 Litteratur

Boken [Wol98] av Wolsey presenterar metoder för heltalsoptimering. En MIP-modell för korttidsplanering av kraftvärmesystem beskrivs av Eriksson [Eri94]. En Branch-and-bound-algoritm för elkraftplanering presenteras i [CoY90].

Kapitel 6

Dynamisk programmering

Många praktiska tillämpningar har ett naturligt förlopp över tiden, dvs processen är dynamisk. Detta innebär även att det resulterande optimeringsproblemet får en speciell struktur, en struktur som kan utnyttjas i lösningsalgoritmen. Dynamisk programmering (eng. Dynamic Programming, förkortat DP) är en metod som utnyttjar just detta.

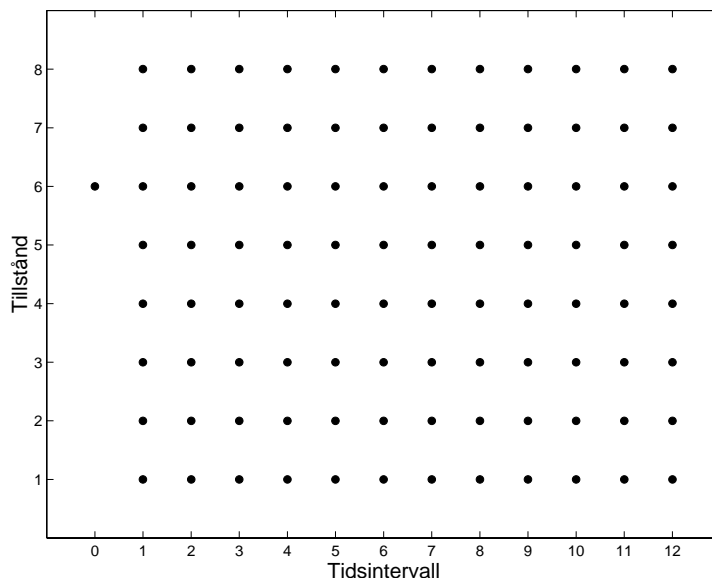
Principen är att först lösa en mängd mindre och relativt enkla problem för varje tidsintervall och sedan koppla dessa med beräkningar som successivt arbetar sig från den sista tidsperioden till den första. Definitionsmässigt görs beräkningarna bakåt över tidshorisonten men man kan lika gärna låta algoritmen jobba framåt över tiden. De problem som löses för varje tidsintervall är vart och ett associerade till ett "tillstånd" (eng. state). Talar man om storleken på tillståndsrummet (eng. state space) avses antalet tillstånd i respektive tidsintervall.

Beräkningsgången för Dynamisk programmering illustreras enklast med praktiska exempel:

6.1 Elproduktionsplanering

Betrakta följande "Unit Commitment"-problem för elproduktion:

$$\begin{aligned} \min_{p,u} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^2 (p_{i,k})^2 + \alpha_{i,k}^1 p_{i,k} + \alpha_{i,k}^0 \right) u_{i,k} \right. \\ & \left. + \sum_{i=1}^I \sum_{k=1}^K (1 - u_{i-1,k}) u_{i,k} \gamma_{i,k} \right] \\ \text{s.t.} & \sum_{k=1}^K p_{i,k} = p_{i,D} \\ & \sum_{k=1}^K \bar{p}_{i,k} u_{i,k} \geq p_{i,R} \\ & \underline{p}_{i,k} u_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} u_{i,k} \\ & u_{0,k} = u_k^{init} \\ & u_{i,k} \in \{0, 1\}. \end{aligned} \tag{6.1}$$



Figur 6.1. Tillståndsrummet för "Unit Commitment"-problemet. Tillstånden motsvarar olika enhetsuppsättningar.

För att problemet ska bli mer gynnsamt ur matematisk synvinkel formuleras produktionskostnaderna som konvexa, dvs $\alpha_{i,k}^2 > 0$ gäller. För en mer formell presentation av teorin kring konvexa optimeringsproblem hänvisas till litteratur inom området icke-linjär optimering. Notera också att den konstanta startkostnaden är modellerad som icke-linjär, till skillnad mot problem (3.1) där motsvarande kostnad modelleras som linjär.

Låt oss anta att systemet som ska modelleras innefattar tre produktionsenheter, $K = 3$, och att vi vill planera produktionen för de närmaste tolv timmarna, $I = 12$. Var och en av de tre produktionsenheterna kan vara i eller ur drift i varje tidsintervall. Med andra ord, i varje tidsintervall finns totalt $2^3 = 8$ möjliga enhetsuppsättningar $u_i = (u_{i,1}, u_{i,2}, u_{i,3})$. Dessa enhetsuppsättningar får definiera tillståndsrummet, dvs vi har åtta tillstånd i tidsintervall i . Figur 6.1 illustrerar det totala tillståndsrummet. I figuren motsvarar tillståndet i period noll den kända initiala enhetsuppsättningen u_k^{init} , $k = 1, 2, 3$.

För varje nod (i, s) i tillståndsrummet löser man ett optimeringsprob-

lem med enbart kontinuerliga variabler:

$$\begin{aligned}
f_i^s = \min_p \left[\sum_{k=1}^K \left(\alpha_{i,k}^2 (p_{i,k})^2 + \alpha_{i,k}^1 p_{i,k} + \alpha_{i,k}^0 \right) u_{i,k} \right] \\
s.t. \quad \sum_{k=1}^K p_{i,k} = p_{i,D} \\
\sum_{k=1}^K \bar{p}_{i,k} u_{i,k} \geq p_{i,R} \\
\underline{p}_{i,k} u_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} u_{i,k},
\end{aligned} \tag{6.2}$$

där $u_{i,k}$, $k = 1, 2, 3$, är givna som den enhetsuppsättning som gäller för den aktuella noden. Det optimala objektfunktionsvärdet betecknas f_i^s . För vissa enhetsuppsättningar, t.ex. $u_i = (0, 0, 0)$, så saknar problem (6.2) lösning. I dessa fall tilldelas f_i^s ett stort värde, förslagsvis $f_i^s = \infty$.

Låt funktionen $g(u_{i-1}^s, u_i^t)$ beteckna de eventuella startkostnader som uppstår när man förändrar enhetsuppsättningen från u_{i-1}^s till u_i^t , dvs då man hoppar från nod $(i-1, s)$ till nod (i, t) i tillståndsrummet. Definiera följande rekursiva funktion:

$$J_{i-1}^s = \min_{t=1, \dots, 8} [f_{i-1}^s + g(u_{i-1}^s, u_i^t) + J_i^t]. \tag{6.3}$$

Låt nu $J_I^s = f_I^s$, $s = 1, \dots, 8$, och applicera (6.3) på tillståndsrummet från tidsintervall $I-1$ och bakåt över tiden. Eftersom initial enhetsuppsättning u_k^{init} , $k = 1, 2, 3$, antas vara känd kan slutligen J_0^{init} beräknas.

Det beräknade värdet J_0^{init} är optimalt objektfunktionsvärde för problem (6.1). Optimal lösning till problem (6.1) hittas genom att följa den (optimala) väg som beräkningarna tagit genom tillståndsrummet. Vägen spåras från tidsintervall noll (dvs från den nod som motsvarar u_k^{init} , $k = 1, 2, 3$) och framåt över tiden.

6.2 Värmeproduktionsplanering

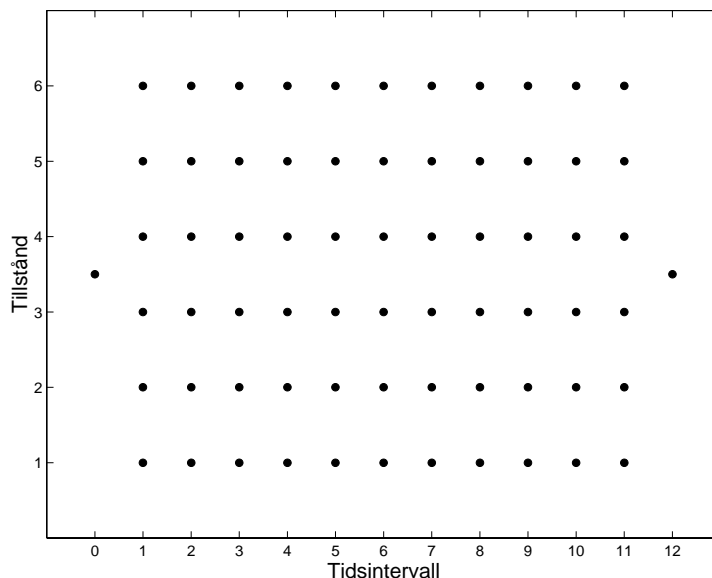
Betrakta följande ”Economic Dispatch”-problem för värmeproduktion:

$$\begin{aligned}
 \min_{q,e} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0 \right) + \sum_{i=1}^I \alpha_{i,S} (q_{i,S})^2 \right] \\
 \text{s.t.} & \sum_{k=1}^K q_{i,k} + q_{i,S} = q_{i,D} \\
 & e_{i,S} = (1 - l_{i,S})e_{i-1,S} - q_{i,S} - l'_{i,S} \\
 & \underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k} \\
 & \underline{q}_{i,S} \leq q_{i,S} \leq \bar{q}_{i,S} \\
 & \underline{e}_{i,S} \leq e_{i,S} \leq \bar{e}_{i,S} \\
 & e_{0,S} = \underline{e}_{0,S} \\
 & e_{I,S} = \underline{e}_{I,S}.
 \end{aligned} \tag{6.4}$$

Av samma anledning som (6.1) formulerades med $\alpha_{i,k}^2 > 0$ formuleras (6.4) så att $\alpha_{i,k}^2 > 0$ och $\alpha_{i,S} > 0$ gäller. För att kunna lösa problemet med Dynamisk programmering diskretiseras de kontinuerliga variabler som beskriver energiinnehållet i ackumulatortanken. Låt säga att vi bedömer det lämpligt att de nya heltalsvariablerna ska kunna anta sex möjliga värden. Vi ersätter därmed de kontinuerliga variablerna $e_{i,S}$ med heltalsvariablerna $e_{i,S}^s$, $s = 1, \dots, 6$. En planeringshorisont på $I = 12$ timmar ger tillståndsrummet i Figur 6.2. I figuren motsvarar tillstånden i period noll och tolv de kända initial och slutnivåerna $\underline{e}_{0,S}$ respektive $\underline{e}_{I,S}$.

En förflyttning i tillståndsrummet från nod $(i-1, s)$ till nod (i, t) motsvarar en förändring i ackumulatorinnehållet från $e_{i-1,S}^s$ till $e_{i,S}^t$. Kostnaden för att åstadkomma förändringen betecknas $g(e_{i-1,S}^s, e_{i,S}^t)$ och beräknas genom att lösa följande problem med kontinuerliga variabler:

$$\begin{aligned}
 g(e_{i-1,S}^s, e_{i,S}^t) = \min_q & \left[\sum_{k=1}^K \left(\alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0 \right) + \alpha_{i,S} (q_{i,S})^2 \right] \\
 \text{s.t.} & \sum_{k=1}^K q_{i,k} + q_{i,S} = q_{i,D} \\
 & e_{i,S}^t = (1 - l_{i,S})e_{i-1,S}^s - q_{i,S} - l'_{i,S} \\
 & \underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k} \\
 & \underline{q}_{i,S} \leq q_{i,S} \leq \bar{q}_{i,S}.
 \end{aligned} \tag{6.5}$$



Figur 6.2. Tillståndsrummet för "Economic Dispatch"-problemet. Tillstånden motsvarar olika energiinnehåll i ackumulatortanken.

Låt $J_I^1 = 0$ och applicera den rekursiva funktionen

$$J_{i-1}^s = \min_{t=1, \dots, 6} [g(e_{i-1,S}^s, e_{i,S}^t) + J_i^t], \quad (6.6)$$

på tillståndsrummet bakåt över tiden tills J_0^1 har beräknats. J_0^1 är det optimala objektionsvärdet för problem (6.4). Optimal lösning till (6.4) hittas genom att följa den (optimala) vägen från noden som motsvarar $\underline{e}_{0,S}$ till noden som motsvarar $\underline{e}_{I,S}$.

6.3 Litteratur

Dynamisk programmering behandlas av Bertsekas i [Ber95]. Metodiken tillämpad på "Unit Commitment"-problem hittas i [HHW88] och [SPR87]. Ravn och Rygaard [RaR94] löser ett "Economic Dispatch"-problem med Dynamisk programmering.

Kapitel 7

Lagrangerelaxering

Det är i princip omöjligt att beskriva en fysikalisk process exakt med matematiska formler. Med andra ord, i alla matematiska modeller finns mer eller mindre stora fel inbyggda. Då detta även gäller optimeringsmodeller kan man fråga sig varför man ska spendera tid på att hitta den optimala lösningen på ett problem som i sig är fel. En lösning som ligger en bit från modellens optimum kan i praktiken vara minst lika bra. Om man med någon metod kan hitta en sådan lösning på kortare tid kan detta vara att föredra. Det handlar alltså om att hitta en tillräckligt bra lösning tillräckligt snabbt snarare än om att lösa problemet exakt.

För att veta när en lösning är tillräckligt bra behövs ett kriterium. Den undre gräns för den optimala lösningen som beräknades med LP-relaxering i Kapitel 5 ger ett sådant mått. En känd tillåten lösning är tillräckligt bra om avståndet till den undre gränsen är tillräckligt litet. I detta kapitel diskuteras Lagrangerelaxering som är en annan metod för att beräkna undre gränser. Lagrangerelaxering (eng. Lagrangian Relaxation, förkortat LR) kan både användas fristående och i kombination med Branch-and-bound-metoden (Kapitel 5).

Betrakta följande optimeringsproblem:

$$\begin{aligned} \min_x [f(x)] \\ \text{s.t. } \quad & g_j(x) = 0, \quad j = 1, \dots, m_e \\ & h_j(x) \leq 0, \quad j = 1, \dots, m_i \\ & x \in \Omega. \end{aligned} \tag{7.1}$$

I problemet finns två typer av bivillkor: ”enkla” bivillkor som beskrivs av mängden Ω och ”komplicerande” bivillkor som beskrivs med funktionerna $g(x)$ och $h(x)$. Med komplicerande bivillkor menas att om de tas bort (relaxeras) blir det resterande problemet väsentligt mycket enklare att lösa.

Med hjälp av Lagrangemultiplikatorerna $\lambda = (\lambda_1, \dots, \lambda_{m_e})$ och $\mu =$

$(\mu_1, \dots, \mu_{m_i})$ definieras det relaxerade problemet:

$$\Phi(\lambda, \mu) = \min_x \left[f(x) + \sum_{j=1}^{m_e} \lambda_j g_j(x) + \sum_{j=1}^{m_i} \mu_j h_j(x) \right] \quad (7.2)$$

s.t. $x \in \Omega,$

där $\Phi(\lambda, \mu)$ är den duala objektfunktionen. Motsvarande duala problem är:

$$\max_{\lambda, \mu} [\Phi(\lambda, \mu)] \quad (7.3)$$

s.t. $\mu \geq 0.$

I relation till (7.3) kallas problem (7.1) det primala problemet. Den duala objektfunktionen är en undre gräns för det optimala objekt-funktionsvärdet i det primala problemet.

Principen bakom Lagrangerrelaxering är att först lösa det duala problemet (7.3), och sedan, givet de optimala duala variablerna λ^* och μ^* , konstruera en lösning till det primala problemet (7.1) från lösningen till motsvarande relaxerade problem (7.2). För vissa problemtyper ges den optimala lösningen till det primala problemet direkt som den relaxerade lösningen. För de problemtyper detta inte gäller måste den relaxerade lösningen modifieras. Detta görs normalt med en heuristisk metod som är anpassad till det aktuella problemets speciella struktur (en heuristisk metod är en metod som arbetar med "tumregler"). I det senare fallet kan man dock inte garantera att man hittar primala problemets optimala lösning, men som nämndes ovan, strategin är ju att försöka hitta en lösning som är tillräckligt bra. Användandet av heuristiska metoder innebär också att man nödvändigtvis inte behöver lösa det duala problemet exakt. Den heuristiska metoden kan lika gärna appliceras för andra värden på λ och μ än dom optimala.

Lösningssgoritmer för det duala problemet (7.3) arbetar normalt på så sätt att de i varje iteration löser det relaxerade problemet (7.2) och uppdaterar värdet på de duala variablerna λ och μ . Vill man lösa det duala problemet effektivt krävs därmed bra metoder för båda dessa beräkningsprocesser. Det normala är att lösningsmetoden för det relaxerade problemet anpassas till det aktuella problemets struktur medan uppdateringen av de duala variablerna görs med någon generell metod. Vi går här inte närmare in på några generella uppdateringsmetoder utan konstaterar bara att det duala problemet ur den aspekten har både positiva och negativa egenskaper.

Här följer nu två tillämpningsexempel:

7.1 Elproduktionsplanering

Betrakta "Unit Commitment"-problemet (6.1), dvs:

$$\begin{aligned}
 \min_{p,u} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^2 (p_{i,k})^2 + \alpha_{i,k}^1 p_{i,k} + \alpha_{i,k}^0 \right) u_{i,k} \right. \\
 & \left. + \sum_{i=1}^I \sum_{k=1}^K (1 - u_{i-1,k}) u_{i,k} \gamma_{i,k} \right] \\
 \text{s.t.} & \sum_{k=1}^K p_{i,k} = p_{i,D} \\
 & \sum_{k=1}^K \bar{p}_{i,k} u_{i,k} \geq p_{i,R} \\
 & \underline{p}_{i,k} u_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} u_{i,k} \\
 & u_{0,k} = u_k^{init} \\
 & u_{i,k} \in \{0, 1\},
 \end{aligned} \tag{7.4}$$

där $\alpha_{i,k}^2 > 0$. Problem (7.4) är det primala problemet.

Låt $\lambda = (\lambda_1, \dots, \lambda_I)$ och $\mu = (\mu_1, \dots, \mu_I)$. Definiera det relaxerade problemet:

$$\begin{aligned}
 \Phi(\lambda, \mu) = \min_{p,u} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^2 (p_{i,k})^2 + \alpha_{i,k}^1 p_{i,k} + \alpha_{i,k}^0 \right) u_{i,k} \right. \\
 & \left. + \sum_{i=1}^I \sum_{k=1}^K (1 - u_{i-1,k}) u_{i,k} \gamma_{i,k} \right. \\
 & \left. + \sum_{i=1}^I \lambda_i \left(p_{i,D} - \sum_{k=1}^K p_{i,k} \right) + \sum_{i=1}^I \mu_i \left(p_{i,R} - \sum_{k=1}^K \bar{p}_{i,k} u_{i,k} \right) \right] \\
 \text{s.t.} & \underline{p}_{i,k} u_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} u_{i,k} \\
 & u_{0,k} = u_k^{init} \\
 & u_{i,k} \in \{0, 1\}.
 \end{aligned} \tag{7.5}$$

Det duala problemet blir:

$$\begin{aligned}
 \max_{\lambda, \mu} & [\Phi(\lambda, \mu)] \\
 \text{s.t.} & \mu \geq 0.
 \end{aligned} \tag{7.6}$$

Vi ser att det relaxerade problemet (7.5) delas upp i K oberoende

problem: ett för varje produktionsenhet k :

$$\begin{aligned}
\min_{p,u} & \left[\sum_{i=1}^I \left(\alpha_{i,k}^2 (p_{i,k})^2 + (\alpha_{i,k}^1 - \lambda_i) p_{i,k} + \alpha_{i,k}^0 - \mu_i \bar{p}_{i,k} \right) u_{i,k} \right. \\
& \left. + \sum_{i=1}^I (1 - u_{i-1,k}) u_{i,k} \gamma_{i,k} \right] \\
s.t. & \quad \underline{p}_{i,k} u_{i,k} \leq p_{i,k} \leq \bar{p}_{i,k} u_{i,k} \\
& \quad u_{0,k} = u_k^{init} \\
& \quad u_{i,k} \in \{0, 1\}.
\end{aligned} \tag{7.7}$$

För problem (7.7) finns effektiva lösningsmetoder. En av vinsterna med att tillämpa Lagrangerelaxering på problem (7.4) är just att man kan lösa det relaxerade problemet på ett effektivt sätt.

Vi poängterar här att den optimala lösningen till det duala problemet (7.6) inte nödvändigtvis direkt genererar den primala lösningen. I detta fall måste därmed heuristik användas till att konstruera en primalt tillåten lösning.

7.2 Värmeproduktionsplanering

Betrakta "Economic Dispatch"-problemet (6.4), dvs:

$$\begin{aligned}
\min_{q,e} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0 \right) + \sum_{i=1}^I \alpha_{i,S} (q_{i,S})^2 \right] \\
s.t. & \quad \sum_{k=1}^K q_{i,k} + q_{i,S} = q_{i,D} \\
& \quad e_{i,S} = (1 - l_{i,S}) e_{i-1,S} - q_{i,S} - l'_{i,S} \\
& \quad \underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k} \\
& \quad \underline{q}_{i,S} \leq q_{i,S} \leq \bar{q}_{i,S} \\
& \quad \underline{e}_{i,S} \leq e_{i,S} \leq \bar{e}_{i,S} \\
& \quad e_{0,S} = \underline{e}_{0,S} \\
& \quad e_{I,S} = \underline{e}_{I,S},
\end{aligned} \tag{7.8}$$

där $\alpha_{i,k}^2 > 0$ och $\alpha_{i,S} > 0$. Problem (7.8) är det primala problemet.

Låt $\lambda = (\lambda_1, \dots, \lambda_I)$ och definiera det relaxerade problemet:

$$\begin{aligned} \Phi(\lambda) = \min_{q,e} & \left[\sum_{i=1}^I \sum_{k=1}^K \left(\alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0 \right) + \sum_{i=1}^I \alpha_{i,S} (q_{i,S})^2 \right. \\ & \left. + \sum_{i=1}^I \lambda_i \left(e_{i,S} - (1 - l_{i,S}) e_{i-1,S} + q_{i,S} + l'_{i,S} \right) \right] \\ \text{s.t.} & \sum_{k=1}^K q_{i,k} + q_{i,S} = q_{i,D} \\ & \underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k} \\ & \underline{q}_{i,S} \leq q_{i,S} \leq \bar{q}_{i,S} \\ & \underline{e}_{i,S} \leq e_{i,S} \leq \bar{e}_{i,S} \\ & e_{0,S} = \underline{e}_{0,S} \\ & e_{I,S} = \underline{e}_{I,S}. \end{aligned} \tag{7.9}$$

Det duala problemet blir:

$$\max_{\lambda} [\Phi(\lambda)]. \tag{7.10}$$

Det relaxerade problemet (7.9) delas upp i I oberoende problem: ett för varje tidsintervall i :

$$\begin{aligned} \min_{q,e} & \left[\sum_{k=1}^K \left(\alpha_{i,k}^2 (q_{i,k})^2 + \alpha_{i,k}^1 q_{i,k} + \alpha_{i,k}^0 \right) + \alpha_{i,S} (q_{i,S})^2 + \lambda_i q_{i,S} \right. \\ & \left. + (\lambda_i - \lambda_{i+1} (1 - l_{i+1,S})) e_{i,S} + \lambda_i l'_{i,S} \right] \\ \text{s.t.} & \sum_{k=1}^K q_{i,k} + q_{i,S} = q_{i,D} \\ & \underline{q}_{i,k} \leq q_{i,k} \leq \bar{q}_{i,k} \\ & \underline{q}_{i,S} \leq q_{i,S} \leq \bar{q}_{i,S} \\ & \underline{e}_{i,S} \leq e_{i,S} \leq \bar{e}_{i,S} \\ & e_{I,S} = \underline{e}_{I,S}, \quad \text{då } i = I. \end{aligned} \tag{7.11}$$

Notera att optimalt $e_{i,S}$ kan beräknas oberoende av $q_{i,k}$ och $q_{i,S}$. Problem (7.11) har en gynnsam struktur och förhållandevis få variabler vilket innebär att problemet kan lösas relativt enkelt.

Här gäller att för de optimala duala variablerna λ^* så är optimal lösning till det relaxerade problemet (7.9) också optimal lösning till det primala problemet (7.8).

7.3 Litteratur

Lagrangerrelaxering presenteras av Bertsekas i boken [Ber82]. Även Fisher [Fis85] beskriver metoden på ett bra sätt. Muckstadt och Koenig [MuK77] var först med att tillämpa Lagrangerrelaxering på "Unit Commitment"-problemet. Sedan dess har många utvecklat och förfinat metodiken, se t.ex. [Dot01], [VIM89] och [ZhG88]. Ett "Economic Dispatch"-problem för värmeproduktion löses med Lagrangerrelaxering i [DHR99].

Kapitel 8

Genetiska algoritmer

Tanken med genetiska algoritmer (eng. Genetic Algorithms, förkortat GA) är att efterlikna den evolution (förändring) som pågår i kroppens celler. I cellerna finns kromosomer (DNA-monekyler) som i sin tur består av gener. Generna innehåller den information som bestämmer cellernas och därmed en arts egenskaper. Under evolutionens gång korsas och muteras geninformationen så att arterna blir allt bättre anpassade till sin omgivning. När man löser ett optimeringsproblem med genetiska algoritmer så startar man med en uppsättning (population) av lösningar som sedan korsas och muteras tills den optimala lösningen ”utvecklats” fram.

Beräkningsgången illustreras med ett exempel:

8.1 Elproduktionsplanering

Vi vill lösa ett ”Unit Commitment”-problem över ett tidsintervall:

$$\begin{aligned} \min_{p,u} & \left[\sum_{k=1}^K (\alpha_k^2 (p_k)^2 + \alpha_k^1 p_k + \alpha_k^0) u_k \right] \\ \text{s.t.} & \sum_{k=1}^K p_k = p_D \\ & \underline{p}_k u_k \leq p_k \leq \bar{p}_k u_k \\ & u_k \in \{0, 1\}. \end{aligned} \tag{8.1}$$

Systemet har $K = 15$ produktionsenheter, vilket innebär 2^{15} möjliga enhetsuppsättningar $u^j = (u_1, \dots, u_{15})$, $j = 1, \dots, 2^{15}$. Frågan är vilken som är den optimala.

För en given enhetsuppsättning u blir det resterande problemet ett problem med kontinuerliga variabler:

$$\begin{aligned} f(u) = \min_p & \left[\sum_{k=1}^K (\alpha_k^2 (p_k)^2 + \alpha_k^1 p_k + \alpha_k^0) u_k \right] \\ \text{s.t.} & \sum_{k=1}^K p_k = p_D \\ & \underline{p}_k u_k \leq p_k \leq \bar{p}_k u_k. \end{aligned} \tag{8.2}$$

Här betecknar $f(u)$ den optimala produktionskostnaden. För de u där problem (8.2) saknar tillåtna lösningar tilldelas $f(u)$ ett stort värde, t.ex. $f(u) = \infty$.

Beräkningarna startas med en population av 100 godtyckligt valda enhetsuppsättningar (individer), $u_{pop} = (u^1, \dots, u^{100})$. Tag nu tio slumpvis utvalda uppsättningar ur populationen och beteckna dessa $\tilde{u} = (\tilde{u}^1, \dots, \tilde{u}^{10})$. Sannolikheten att en lösning blir utvald är omvänt proportionell mot $f(u)$, dvs enhetsuppsättningar u med lågt $f(u)$ har större sannolikhet att bli utvalda än de med högt.

Antag att följande uppsättningar valdes:

$$\tilde{u} = \begin{cases} \tilde{u}^1 = (1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1) \\ \tilde{u}^2 = (1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1) \\ \tilde{u}^3 = (0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1) \\ \tilde{u}^4 = (1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0) \\ \tilde{u}^5 = (1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1) \\ \tilde{u}^6 = (0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1) \\ \tilde{u}^7 = (0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0) \\ \tilde{u}^8 = (1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0) \\ \tilde{u}^9 = (1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0) \\ \tilde{u}^{10} = (1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1) \end{cases} \quad (8.3)$$

Dessa ska nu korsas två och två så att två nya uppsättningar (avkommor) skapas från de två ursprungliga (föräldrarna). Korsningen kan göras enligt följande. Först slumpar man fram ett heltal v , $v \in \{1, \dots, 15\}$. Givet v så bildas den ena avkomman av de första v komponenterna från den första föräldern och de sista $15 - v$ komponenterna från den andra föräldern. Den andra avkomman bildas på motsvarande sätt men med den skillnaden att de första v komponenterna tas från den andra föräldern och de sista $15 - v$ från den första. Om vi i vårt exempel antar att $v = 9$ för föräldraparet

$$\begin{cases} \tilde{u}^1 = (\mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}) \\ \tilde{u}^2 = (1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1) \end{cases} \quad (8.4)$$

så blir resultatet avkommorna

$$\begin{cases} \tilde{u}^{1*} = (\mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}) \\ \tilde{u}^{2*} = (1, 1, 0, 0, 1, 0, 1, 1, 0, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}) \end{cases} \quad (8.5)$$

Från de tio föräldrarna $\tilde{u} = (\tilde{u}^1, \dots, \tilde{u}^{10})$ genereras totalt tio avkommor $\tilde{u}^* = (\tilde{u}^{1*}, \dots, \tilde{u}^{10*})$.

Avkommorna ska nu muteras. För var och en kan detta göras genom att först slumpa fram ett heltal v , $v \in \{1, \dots, 15\}$, och sedan ändra värdet på motsvarande position. Till exempel, om vi för avkomman \tilde{u}^{1*} i (8.5) slumpat fram värdet $v = 3$ blir den muterade avkomman lika med:

$$\tilde{u}^{1**} = (1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1). \quad (8.6)$$

Resultatet blir tio muterade avkommor $\tilde{u}^{**} = (\tilde{u}^{1**}, \dots, \tilde{u}^{10**})$.

De tio nya enhetsuppsättningarna \tilde{u}^{**} ska nu ersätta tio gamla i populationen u_{pop} . Detta kan göras genom att t.ex. ta bort föräldrarna \tilde{u} . Filosofin är att bättre lämpade individer (enhetsuppsättningar) ska ersätta de som är sämre lämpade. Resultatet blir att vi fortfarande har en population som består av 100 enhetsuppsättningar.

Vi har därmed genomfört en iteration med den genetiska algoritmen. Den population som erhållits har förhoppningsvis ett totalt sett bättre (dvs lägre) $f(u)$. Genom att upprepa beräkningarna och göra nya korsningar och nya mutationer så kommer förhoppningsvis populationens totala $f(u)$ att minska ytterligare, och ju fler iterationer man gör desto större är sannolikheten att den optimala lösningen till problem (8.1) till slut återfinns i populationen. I den meningen är den genetiska algoritmen en statistisk metod, med andra ord, när man löser ett problem med genetiska algoritmer finns inga garantier för att man verkligen hittar den optimala lösningen.

Det finns många varianter av algoritmen som beskrivits ovan. Till exempel, antal individer i populationen och initial population kan väljas på olika sätt, antal föräldrar och vilka föräldrar som ska korsas kan väljas på olika sätt, korsningarna och mutationerna kan göras på olika sätt, och slutligen, vilka individer som ska ersättas av de muterade avkommorna kan väljas på olika sätt.

8.2 Litteratur

Genetiska algoritmer diskuteras i [Mic96]. Kazarlis, Bakirtzis och Petridis [KBP96] löser ett "Unit Commitment"-problem med genetiska algoritmer. Cheng, Liu och Liu [CLL00] kombinerar Lagrangerrelaxering med genetiska algoritmer.

Kapitel 9

Stokastisk optimering

I praktiken är det exakta värdet på många av de parametrar som påverkar en process okända på förhand. Till exempel, lastprognoser och elprisprognoser för de närmaste dygnet är mer eller mindre osäkra. Denna typ av osäkerhet beaktas i modeller för stokastisk optimering.

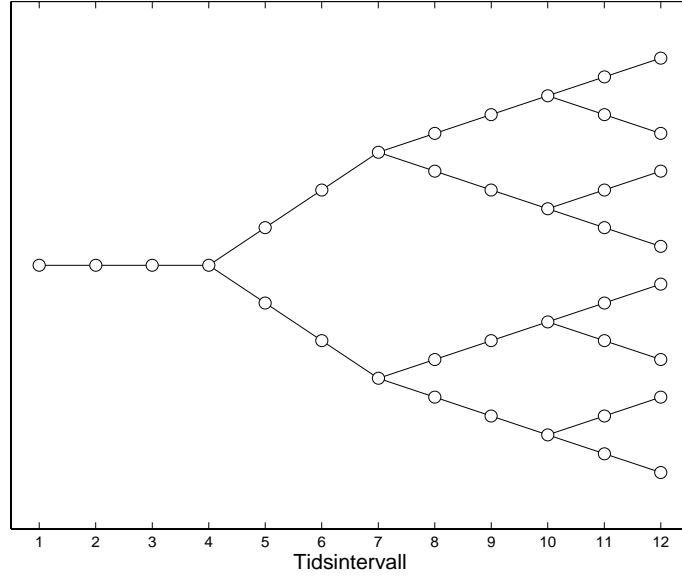
Vid byggandet av en stokastisk modell skapas först en mängd scenarios som tillsammans bildar en trädstruktur, ett så kallat scenarioträd. Till varje scenario associeras en parameter som beskriver sannolikheten att just det scenariot kommer att inträffa i framtiden. Optimeringsproblemet formuleras så att scenariorna viktas i förhållande till sin sannolikhet.

Vi illustrerar med ett exempel:

9.1 Värmeproduktionsplanering

Låt säga att lastprognosen $q_{i,D}$, $i = 1, \dots, I$, i problem (3.2) är osäker. Vi bedömer att lasten kommer att följa ett av åtta möjliga scenarios. Med en tidshorisont på $I = 12$ timmar erhåller vi scenarioträdet i Figur 9.1. Trädet är uppbyggt av N noder. Trädets rotnod $n = 1$ (motsvarande tidsintervall $i = 1$) har ingen föregående nod. Övriga noder $n = 2, \dots, N$ har alla en unik föregående nod som betecknas n_- . En sannolikhet $P_{n/n_-} > 0$ associeras till nod n . P_{n/n_-} tolkas som sannolikheten att nod n är efterföljare till nod n_- . För varje nod n_- är summan av alla P_{n/n_-} lika med ett. Sannolikheten P_n att nod $n \neq 1$ ska inträffa beräknas rekursivt med $P_n = P_{n/n_-} P_{n_-}$. Per definition gäller $P_1 = 1$. Noder som saknar efterföljande noder kallas lövnoder. Mängden av alla lövnoder betecknas N^{leaf} . Ett scenario motsvarar en väg från rotnoden till en lövnod.

I den stokastiska versionen av problem (3.2) ersätts det tidsinter-



Figur 9.1. Scenariotråd med åtta scenarios.

vallbaserade modellerandet (index i) med det nodbaserade (index n):

$$\begin{aligned}
 \min_{q,e} & \left[\sum_{n=1}^N P_n \left(\sum_{k=1}^K \left(\alpha_{n,k}^2 (q_{n,k})^2 + \alpha_{n,k}^1 q_{n,k} + \alpha_{n,k}^0 \right) + \alpha_{n,S} (q_{n,S})^2 \right) \right] \\
 \text{s.t.} & \sum_{k=1}^K q_{n,k} + q_{n,S} = q_{n,D} \\
 & e_{n,S} = e_{n-,S} - q_{n,S} \\
 & \underline{q}_{n,k} \leq q_{n,k} \leq \bar{q}_{n,k} \\
 & \underline{q}_{n,S} \leq q_{n,S} \leq \bar{q}_{n,S} \\
 & \underline{e}_{n,S} \leq e_{n,S} \leq \bar{e}_{n,S} \\
 & e_{0,S} = \underline{e}_{0,S} \\
 & e_{n,S} = \underline{e}_{n,S}, \quad n \in N^{leaf}.
 \end{aligned} \tag{9.1}$$

Observera att ackumulatorbivillkoret (3.7) är modifierat till att följa aktuellt scenario från den aktuella noden mot rotnoden. Optimal lösning till problemet är den produktionsplan som bäst beaktar alla scenarios.

Studerar man problem (9.1) inses att ett stokastiskt optimeringsproblem i princip kan ses som ett ”vanligt” optimeringsproblem. Stokastiska problem blir dock ofta stora, vilket i praktiken innebär att

lösningsmetoden måste anpassas till problemets speciella struktur.

9.2 Litteratur

En introduktion till metoder för stokastisk optimering ges av Birge och Louveaux [BiL97]. Stokastiska planeringsmodeller för elproduktion diskuteras i [DeR98]. Motsvarande problem för värmeproduktion återfinns i [Dot01].

Kapitel 10

Linjära tidsserier

En tidsserie är en sekvens med variabler y_i där index i representerar motsvarande tidsintervall. Om y_i kan beskrivas som en linjär funktion av serien självt och andra tidsserier säger man att serien är en linjär tidsserie.

Betrakta y_i som en utsignal och antag att den påverkas av två insignaler: en styrsignal u_i och en störningssignal betecknad s_i . Låt följande samband gälla:

$$y_i + \alpha_1 y_{i-1} + \dots + \alpha_m y_{i-m} = \beta_1 u_{i-1} + \dots + \beta_m u_{i-m} + s_i + \gamma_1 s_{i-1} + \dots + \gamma_m s_{i-m}. \quad (10.1)$$

Denna modell kallas ARMAX-processen (eng. AutoRegressive, Moving Average, with eXogenous input). Namnet kommer av att $y_i + \alpha_1 y_{i-1} + \dots + \alpha_m y_{i-m}$ representerar vad som brukar kallas en autoregression (AR), $s_i + \gamma_1 s_{i-1} + \dots + \gamma_m s_{i-m}$ är ett glidande medelvärde (MA) av störningen och $\beta_1 u_{i-1} + \dots + \beta_m u_{i-m}$ beskriver inverkan av den ”extra” insignalen (X).

Det vanligaste användningsområdet för ARMAX-processen när det gäller produktionsplanering är som lastprognosmodell, dvs den används till att bestämma elbehovet $p_{i,D}$, $i = 1, \dots, I$, och värmebehovet $q_{i,D}$, $i = 1, \dots, I$. Beräkningarna går till på så sätt att man först ansätter en lämplig modell. Därefter anpassas modellparametrarna till historiska uppmätta data. Lastprognosen konstrueras slutligen genom att kombinera modellen med en väderprognos.

Vi illustrerar principen med ett exempel:

10.1 Ellastprognoser

Låt säga att vi vill göra en prognos för ellasten i ett kraftsystem. Vi analyserar systemet och drar slutsatsen att utomhustemperaturen tillsammans med konsumenternas beteende har störst påverkan på lasten. Det är ganska enkelt att inse att konsumenternas inverkan varierar både under dygnet och mellan veckodagar. Det normala är att kundernas förbrukning är högre på dagen än på natten och att förbrukningen

under vardagar är högre än vid veckoslut. Vi drar även slutsatsen att vindhastigheten kan ha en viss inverkan på lasten. Andra faktorer som kan tänkas påverka är vindriktning, solinstrålning och nederbörd, men vi antar i vårt exempel att inverkan från dessa är försumbar.

Utifrån analysen ansätter vi följande modell:

$$\begin{aligned}
p_{i,D} = & \delta - \alpha_1 p_{i-1,D} - \alpha_2 p_{i-2,D} - \alpha_3 p_{i-3,D} \\
& - \alpha_{23} p_{i-23,D} - \alpha_{24} p_{i-24,D} - \alpha_{25} p_{i-25,D} \\
& + \beta_0^t t_i + \beta_1^t t_{i-1} + \beta_2^t t_{i-2} \\
& + \beta_{23}^t t_{i-23} + \beta_{24}^t t_{i-24} + \beta_{25}^t t_{i-25} \\
& + \beta_0^w w_i + \beta_1^w w_{i-1} + \beta_2^w w_{i-2}.
\end{aligned} \tag{10.2}$$

Variabeln t_i betecknar utomhustemperaturen timme i och variabeln w_i betecknar vindhastigheten. Man har i modellen antagit att lasten beror av sina egna värden från timmarna strax innan aktuell timme och från timmarna under samma period föregående dygn. Utomhustemperaturen antas ha motsvarande påverkan. Vindens inverkan modelleras enbart med timmarna strax innan aktuell timme. Man har i modellen även lagt till en konstant δ med syfte att korrigera lastprognosens nivå.

Nästa steg är att bestämma värdet på modellparametrarna. Detta kan göras med minstakvadratmetoden:

$$\min_{\alpha, \beta, \delta} \left[\sum_{i=1}^H \left(\hat{p}_{i,D} - \delta + \sum_{j=1}^3 \alpha_j \hat{p}_{i-j,D} + \sum_{j=23}^{25} \alpha_j \hat{p}_{i-j,D} - \sum_{j=0}^2 \beta_j^t \hat{t}_{i-j} - \sum_{j=23}^{25} \beta_j^t \hat{t}_{i-j} - \sum_{j=0}^2 \beta_j^w \hat{w}_{i-j} \right)^2 \right], \tag{10.3}$$

där $\hat{p}_{i,D}$, \hat{t}_i och \hat{w}_i är uppmätta data från motsvarande period föregående år och H är antalet timmar vi har data för. Genom att lösa problem (10.3) så hittar man de värden på α , β och δ som bäst matchar historiska data.

Modellen är nu klar och vi är därmed redo att göra en prognos $\tilde{p}_{i,D}$, $i = 1, \dots, I$. Prognosen konstrueras genom att kombinera (10.2) med prognoser för utomhustemperatur \tilde{t}_i , $i = 1, \dots, I$, och vindhastighet \tilde{w}_i , $i = 1, \dots, I$.

Vi noterar här att metoden som beskrivits ovan på ett bra sätt illustrerar principen för hur modellering med ARMAX-processer går till men att man vid praktiska tillämpningar gör beräkningarna något

annorlunda. Till exempel förbehandlas ofta tidsserierna på så sätt att en series medelvärde dras ifrån innan beräkningarna påbörjas för att sedan läggas till igen i prognostiseringskedet. En annan variant är att vikta in en dygnsprofil i modellen. I exemplet löstes endast ett minstakvadratproblem. Man kan även tänka sig att man löser flera problem där man först låter algoritmen identifiera den komponent som har störst påverkan på lasten och beräknar dess parameter, algoritmen identifierar sedan komponenten med näst störst påverkan och beräknar dess parameter, osv.

Vi avslutar tidsseriekapitlet med att kort nämna Kalmanfiltret. När man bygger en matematisk modell av en fysikalisk process kan det hända att man i modellen får tillstånd (variabler) som inte går att mäta i praktiken. Till exempel, låt säga att man modellerat ett fjärrvärmenät som ett nätverk med noder och bågar. Till noderna associeras tidsserier för vattentemperatur och till bågarna (distributionsledningarna) associeras tidsserier för vattenflöde. Normalt finns inte mätning vid alla noder och bågar. Om så är fallet vill man estimeras (uppskatta) motsvarande tidsserier. För vissa typer av modeller kan man med Kalmanfiltret göra estimeringen så att störningssignalerna hanteras på ett optimalt sätt.

10.2 Litteratur

ARMAX-processen och Kalmanfiltret beskrivs av Bertsekas [Ber95]. Båda metoderna tillämpas av Arvastson [Arv01] vid modellering av fjärrvärmenät.

Kapitel 11

Artificiella neurala nätverk

Endast det bästa är gott nog. Varför inte använda samma metod som används av den mest geniala modellen av dom alla: den mänskliga hjärnan. Det är grundidén för modeller som baseras på artificiella neurala nätverk (eng. Artificial Neural Network, förkortat ANN).

I hjärnan finns neuroner som är en typ av celler med ”sensorer” som känner av impulser. Neuronerna reagerar på impulserna med att själva ge ifrån sig impulser. En neuron kan ses som en cell med flera ”insignaler” och en ”utsignal”. Hur utsignalen konstrueras från insignalerna beror på utseendet hos den individuella neuronerna. Utsignalsimpulserna påverkar andra neuroner, dvs neuronerna är kopplade i ett nätverk: ett neuralt nätverk. När man på konstgjord väg med datorns hjälp försöker efterlikna sådana nät talar man om artificiella neurala nätverk.

I produktionsplaneringsammanhang används ANN till att generera lastprognoser. På motsvarande sätt som man med tidsseriemodellen i Kapitel 10 ansatte ett lämpligt uttryck definierar man här först ett nätverk. Modellparametrarna anpassas till historiska data och lastprognosen konstrueras slutligen från en väderprognos.

Tekniken illustreras med ett exempel:

11.1 Värmelastprognoser

Vi beräknar här (för enkelhetens skull) enbart ett prognosvärde: värmelasten $\tilde{q}_{i,D}$ för timme i .

Innan nätverket konstrueras måste neuronerna definieras. Det finns några olika typer men vi väljer att använda den så kallade tanh-neuronerna:

$$y = \tanh \left(\sum_{k=1}^K \omega_k x_k - \omega_0 \right). \quad (11.1)$$

Neuronerna har K stycken insignaler x_k , $k = 1, \dots, K$. Hur neuronerna reagerar på insignalerna bestäms av parametrarna ω_k , $k = 0, \dots, K$.

Det enklaste ANN vi kan konstruera med neuronerna (11.1) är:

$$y = \sum_{j=1}^J \nu_j \tanh \left(\sum_{k=1}^K \omega_{j,k} x_{j,k} - \omega_{j,0} \right), \quad (11.2)$$

där J är antalet neuronerna. Parametrerna ν_j , $j = 1, \dots, J$, och $\omega_{j,k}$, $j = 1, \dots, J$, $k = 0, \dots, K$, bestämmer nätets egenskaper.

Precis som när man konstruerade ellastprognosen i Kapitel 10 analyserar vi här det aktuella fjärrvärmesystemet. Antag att vi från analysen drar slutsatsen att värmelasten $q_{i,D}$ kan beskrivas av sina egna tidigare värden $q_{i-1,D}$, $q_{i-2,D}$, $q_{i-3,D}$, $q_{i-23,D}$, $q_{i-24,D}$ och $q_{i-25,D}$, samt av utomhustemperaturerna t_i , t_{i-1} , t_{i-2} , t_{i-23} , t_{i-24} och t_{i-25} . I exemplet används ett ANN med $J = 4$ neuronerna som var och en har $K = 3$ insignaler, dvs nätverket har totalt tolv insignaler och en utsignal:

$$\begin{aligned} q_{i,D} = & \\ & \nu_1 \tanh \left(\sum_{k=1}^3 \omega_{1,k} q_{i-k,D} - \omega_{1,0} \right) + \nu_2 \tanh \left(\sum_{k=23}^{25} \omega_{2,k} q_{i-k,D} - \omega_{2,0} \right) \\ & + \nu_3 \tanh \left(\sum_{k=1}^3 \omega_{3,k} t_{i-k+1} - \omega_{3,0} \right) + \nu_4 \tanh \left(\sum_{k=23}^{25} \omega_{4,k} t_{i-k} - \omega_{4,0} \right). \end{aligned} \quad (11.3)$$

Nätverket är nu definierat. Nästa steg är att anpassa parametrarna ν_j och $\omega_{j,k}$ till uppmätta data för last $\hat{q}_{i,D}$, $i = 1, \dots, H$, och utomhustemperatur \hat{t}_i , $i = 1, \dots, H$. Man brukar säga att man "tränar" nätverket. Även här kan detta göras genom att försöka minimera kvadratfelet, men till skillnad mot problem (10.3) blir det resulterande optimeringsproblemet mycket svårare. Vi går inte närmare in på några lösningsstrategier utan konstaterar bara att en metod som ofta används heter Backpropagation. Metoden arbetar på så sätt att parametervärdena ändras stegvis och i proportion till deras bidrag till det totala felet. Den stegvisa förändringen upprepas tills man bedömer att modellen passar uppmätta data tillräckligt bra.

När nätverket är färdigtränat kan värmelastprognosen $\tilde{q}_{i,D}$ beräknas genom att evaluera (11.3) med temperaturprognosen \tilde{t}_i och det senaste dygnets uppmätta last och temperatur.

Vi påpekar åter igen att det ANN som används i exemplet är en mycket enkel variant som knappast skulle ge tillfredsställande prognoser i praktiken. För att bli praktiskt användbar måste metodiken dessutom generaliseras till att beräkna en serie av prognosvärden $\tilde{q}_{i,D}$,

$i = 1, \dots, I$. Hur som helst, exemplet ger en bild av principen. För en mer ingående teoretisk beskrivning hänvisas till litteratur inom området.

11.2 Litteratur

ANN beskrivs av Simon Haykin i [Hay94]. Hippert, Pedreira och Souza [HPS01] beskriver användningen av ANN vid lastprognostisering.

Kapitel 12

Modelleringspråk

När man bestämt vilken typ av modell och vilken optimeringsmetod man ska använda måste dessa implementeras, dvs programmeras som ett datorprogram. Detta kan göras i ett högnivåspråk där de i dessa sammanhang mest använda är C, C++ och Fortran. Många kommersiella optimeringslösare som finns på marknaden är skrivna i dessa språk och är därmed relativt enkla att koppla till sitt eget skrivna program. Grundregeln kan sägas vara att använda standardmjukvara när detta är möjligt, men många gånger räcker inte det utan man måste själv implementera sina algoritmer.

Ett alternativ är att programmera modellerna i ett modelleringspråk. Till dessa språk hör t.ex. GAMS och AMPL. Till många kommersiella optimeringslösare finns kopplingar till modelleringspråken så dessa är ofta enkla att använda. En nackdel med modelleringspråken är att de inte klarar allt för komplicerade modeller.

Vi illustrerar programmering i GAMS med ett enkelt exempel:

12.1 Modellering i GAMS

Modelleringspråket GAMS är anpassat för utveckling av optimeringsmodeller. Syntaxen är sådan att man relativt enkelt kan översätta ett matematiskt formulerat problem till programmeringskod.

Betrakta följande problem:

$$\begin{aligned} \min_q \quad & \left[\sum_{i=1}^4 \sum_{k=1}^3 \alpha_{i,k} q_{i,k} \right] \\ \text{s.t.} \quad & \sum_{k=1}^3 q_{i,k} = q_{i,D} \\ & q_{i,k} \geq 0. \end{aligned} \tag{12.1}$$

Figur 12.1 visar motsvarande GAMS-kod. Man börjar med att definiera två mängder (SET). Mängden I innehåller fyra element (I1,...,I4) och mängden K innehåller tre (K1,...,K3). Parametervärden för värmelast och kostnad definieras med kommandot PARAMETER. Två typer av

```

SET I Time intervals /I1*I4/;
SET K Production units /K1*K3/;
PARAMETER heatDemand(I) Heat demand /
I1 264
osv. /;
PARAMETER alpha(I,K) Production cost /
I1.K1 405
osv. /;
VARIABLES
OBJ Objective function
Q(I,K) Heat production;
POSITIVE VARIABLE Q;
EQUATIONS
TotCost Objective function
Demand(I) Heat demand;
TotCost.. OBJ =E= SUM((I,K),alpha(I,K)*Q(I,K));
Demand(I).. SUM(K,Q(I,K)) =E= heatDemand(I);
MODEL EXAMPLE /ALL/;
SOLVE EXAMPLE USING LP MINIMIZING OBJ;

```

Figur 12.1. GAMS-kod.

variabler (VARIABLES) definieras: OBJ betecknar objektfunktionsvärdet och $Q(I,K)$ motsvarar $q_{i,k}$. Variabeln $q_{i,k}$ i problem (12.1) är ickenegativ (POSITIVE VARIABLE). Två typer av bivillkor (EQUATIONS) definieras: TotCost beskriver produktionskostnaden och Demand(I) formulerar behovsvillkoren. Med kommandot MODEL skapas ett optimeringsproblem med namnet EXAMPLE. För att slutligen lösa (SOLVE) problemet anropar vi en optimeringslösare för LP-problem (LP). Vi talar om för lösaren att det är variabeln OBJ som ska minimeras.

12.2 Litteratur

Modelleringspråken GAMS och AMPL beskrivs i [BKM98] respektive [FGK93]. På Internet, www.ece.nwu.edu/OTC, finns en lista med optimeringsmjukvara.

Kapitel 13

Litteratur

Detta kapitel listar referenser relevanta för området.

13.1 Litteraturlista

- [Arv01] Lars Arvastson, *Stochastic Modeling and Operational Optimization in District Heating Systems*, Avhandling, Mathematical Statistics, Lund University, Lund, Sweden, 2001.
- [Ber82] Dimitri P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982.
- [Ber95] Dimitri P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 1995.
- [BeT97] Dimitris Bertsimas och John N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1997.
- [BiL97] John R. Birge och Francois Louveaux, *Introduction to Stochastic Programming*, Springer, 1997.
- [BKM98] A. Brooke, D. Kendrick, A. Meeraus och R. Raman, *GAMS - A User's Guide*, GAMS Development Corporation, 1998.
- [CLL00] C. P. Cheng, C. W. Liu och C. C. Liu, *Unit Commitment by Lagrangian Relaxation and Genetic Algorithms*, IEEE Trans. on Power Systems, vol. 15, sid. 707-714, 2000.
- [ChR90] B. H. Chowdhury och S. Rahman, *A Review of Recent Advances in Economic Dispatch*, IEEE Trans. on Power Systems, vol. 5, sid. 1248-1259, 1990.
- [CoY90] Arthur I. Cohen och Miki Yoshimura, *A Branch-and-Bound Algorithm for Unit Commitment*, IEEE Trans. on Power Apparatus and Systems, vol. 102, sid. 444-451, 1983.
- [DeR98] D. Dentcheva och W. Römisch, *Optimal Power Generation Under Uncertainty via Stochastic Programming*, Stochastic

Programming Methods and Technical Applications, Editors: K. Marti och P. Kall, Lecture Notes in Economics and Mathematical Systems, Springer Verlag, vol. 458, sid. 22-56, 1998.

- [DHR99] E. Dotzauer, K. Holmström och H. F. Ravn, *Optimal Unit Commitment and Economic Dispatch of Cogeneration Systems with a Storage*, Proceedings 13th Power Systems Computation Conference, Norway, vol. 2, sid. 738-744, 1999.
- [Dot01] Erik Dotzauer, *Energy System Operation by Lagrangian Relaxation*, Avhandling, Division of Optimization, Department of Mathematics, Linköping University, Linköping, Sweden, 2001.
- [Eri94] Hans Eriksson, *Short Term Operation of District Heating Systems*, Avhandling, Department of Energy Conversion, Chalmers University of Technology, Gothenburg, Sweden, 1994.
- [Fis85] Marshall L. Fisher, *An Applications Oriented Guide to Lagrangian Relaxation*, Interfaces, vol. 15, sid. 10-21, 1985.
- [Fle87] Roger Fletcher, *Practical Methods of Optimization*, Wiley, 1987.
- [FGK93] R. Fourer, D. M. Gay och B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press/Wadsworth Publishing Company, 1993.
- [FrW93] Svend Fredriksen och Sven Werner, *Fjärrvärme - Teori, Teknik och Funktion*, Studentlitteratur AB, 1993.
- [Hay94] Simon Haykin, *Neural Networks - A Comprehensive Foundation*, MacMillan Collage Publishing Company, 1994.
- [HPS01] H. S. Hippert, C. E. Pedreira och R. C. Souza, *Neural Networks for Short-Term Forecasting: a Review and Evaluation*, IEEE Trans. on Power Systems, vol. 16, sid. 44-55, 2001.
- [HHW88] W. J. Hobbs, G. Hermon, S. Warner och G. B. Sheble, *An Enhanced Dynamic Programming Approach for Unit Commitment*, IEEE Trans. on Power Systems, vol. 3, sid. 1201-1205, 1988.

- [KBP96] S. A. Kazarlis, A. G. Bakirtzis och V. Petridis, *A Genetic Algorithm Solution to the Unit Commitment Problem*, IEEE Trans. on Power Systems, vol. 11, sid. 83-92, 1996.
- [Mic96] Zbigniew Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, 1996.
- [MuK77] John A. Muckstadt och Sherri A. Koenig, *An Application of Lagrangian Relaxation to Scheduling in Power-Generation Systems*, Operations Research, vol. 25, sid. 387-403, 1977.
- [Nil97] Olof Nilsson, *Short Term Scheduling of Hydrothermal Power Systems with Integer Hydro Constraints*, Avhandling, Electric Power Systems, Royal Institute of Technology, Stockholm, Sweden, 1997.
- [RaR94] Hans F. Ravn och Jens M. Rygaard, *Optimal Scheduling of Coproduction with a Storage*, Eng. Opt., vol. 22, sid. 267-281, 1994.
- [SeK98] Subir Sen och D. P. Kothari, *Optimal Thermal Generating Unit Commitment: a Review*, Electrical Power & Energy Systems, vol. 20, sid. 443-451, 1998.
- [ShF94] Gerald B. Sheble' och George N. Fahd, *Unit Commitment Literature Synopsis*, IEEE Trans. on Power Systems, vol. 9, sid. 128-135, 1994.
- [SPR87] W. L. Snyder, H. D. Powell och J. C. Rayburn, *Dynamic Programming Approach to Unit Commitment*, IEEE Trans. on Power Systems, vol. 2, sid. 339-350, 1987.
- [VIM89] S. Virmani, K. Imhof och S. Mukherjee, *Implementation of a Lagrangian Relaxation Based Unit Commitment Problem*, IEEE Trans. on Power Systems, vol. 4, sid. 1373-1379, 1989.
- [Win95] Wayne L. Winston, *Introduction to Mathematical Programming - Applications and Algorithms*, Duxbury Press, 1995.
- [Wol98] Laurence A. Wolsey, *Integer Programming*, John Wiley and Sons, 1998.

- [Zha95] Hongping Zhao, *Analysis, Modelling and Operational Optimization of District Heating Systems*, Avhandling, Centre for District Heating Technology, Laboratory of Heating and Air Conditioning, Technical University of Denmark, Lyngby, Denmark, 1995.
- [ZhG88] F. Zhuang och F. D. Galiana, *Towards a More Rigorous and Practical Unit Commitment by Lagrangian Relaxation*, IEEE Trans. on Power Systems, vol. 3, sid. 763-773, 1988.

