

Kursinformation TAOP62, Optimeringslära fortsättningskurs 6 hp, VT1-2026

1. Kursmål & innehåll

1.1 Mål med kursen

Inom optimeringslära behandlas matematiska teorier **och** metoder som syftar till att analysera och lösa beslutsproblem som uppkommer inom teknik, ekonomi, medicin, etcetera. Kursen ger, tillsammans med grundkursen, en bred orientering om optimeringslära, med inriktning mot grundläggande teori och metoder för diskreta optimeringsproblem i ändlig dimension, samt en inblick i dess tillämpning för att analysera praktiska optimeringsfrågeställningar. Efter fullgjord kurs skall studenten:

1. kunna redogöra för viktiga klasser av optimeringsproblem och kunna klassificera optimeringsproblem utifrån deras egenskaper, som till exempel i nätverk eller diskreta problem
2. kunna modellera matematiska modeller av enkla optimeringsproblem
3. kunna redogöra för grundläggande begrepp, som till exempel optimalitetsvillkor, svag och stark dualitet, samt giltiga olikheter
4. ha kunskap om och kunna använda grundläggande teori för några vanliga typer av optimeringsproblem, som till exempel dualitetsteori för linjära (nätverks)problem, och ha kännedom om och kunna utnyttja optimalitetsvillkor, som till exempel Bellmans ekvationer, för att avgöra optimalitet för ett en föreslagen lösning
5. kunna redogöra för olika grundläggande algoritmer och kunna sammanfatta principerna bakom algoritmerna för att lösa några vanligt förekommande typer av optimeringsproblem, som till exempel trädsökning för diskreta problem
6. kunna utnyttja relaxeringar, och speciellt Lagrange-dualitet, för att approximera optimeringsproblem, samt kunna stänga in optimalvärden med hjälp av optimistiska och pessimistiska uppskattningar
7. kunna använda vanligt förekommande optimeringsprogramvara för att lösa standardmässiga optimeringsproblem
8. ha viss kunskap om praktiska tillämpningar av optimeringsproblem.

1.2 Förkunskaper

Optimeringslära grundkurs, t.ex. TAOP52.

1.3 Kursinnehåll

Kursen omfattar följande områden¹:

- Nätverksoptimering: Problem med nätverksstruktur, linjärprogrammering med heltalsegenskap, problem av billigaste väg-typ, flöden i nätverk, minskostnadsflödesproblem.

¹ På många ställen där kursens olika områden beskrivs, kommer de ofta i ordningen nedan, även om undervisningsordningen under 2026 har sista två områdena i omvänd ordning.

- Heltalsprogrammering: Heltalsproblem med nätverks/grafstruktur, optimeringsmodeller med diskreta variabler, lösningsmetoder baserade på trädsökning, plansnittning, heuristiker och metaheuristiker.
- Lagrangerrelaxation: Grundläggande teori och principer för lösningsmetoder baserade på Lagrangerrelaxation. Fokus på hur skattningar kan genereras och tillämpningar på nätverksproblem och heltalsproblem.
- Dynamisk programmering: Problemformulering, optimalitetsprincipen, tillämpningar på lagerhållningsproblem och resursallokeringsproblem.

2. Administration & Organisation

2.1 Kurshemsida

Kursens användar LISAM för senaste info, kursmaterial, mm.

2.2 Lärare/Föreläsare

Stefan Engevall	Examinator, Kursansvarig, Medverkar på samtliga kursmoment Kontaktinformation: E-post: stefan.engevall@liu.se , Telefon: 073-6569326, Rum: B-huset, A-korridor mellan 21&23, Bottenvåning, 2A:646
Stefan Engevall	Lektioner (a/b-klass), Labbar, Redovisningar
Jan Rolfes	Lektioner (c/d/e-klass) samt Extrapass lektioner (alla klasser), Labbar (huvudsakligen engelskspråkig), Redovisningar
Sachin Rajendran	Lektioner (f/g-klass), Labbar (huvudsakligen engelskspråkig), Redovisningar

2.3 Disposition

Kursen omfattar 6 hp = 160 h, och en uppskattning kan vara följande fördelning av tidsanspråk:

Schemalagt, totalt 67 h/student:

- Storseminarier: 24 h
- Inspelade Föreläsningar 16 h
- Lektioner: 16 h
- Utförande Laborationer: 4 h
- Redovisningar av Laborationer: 2 h (genomförande)
- Tenta: 5 h

Icke Schemalagt: Totalt 93 h/student:

- Teoriinhämtning, förberedelser för, och avrapportering av, laborationer: 21 h
- Teoriinhämtning, egen räkning, inför tenta: 72 h

2.4 Organisation

Kursen bedrivs i form av storseminarier, inspelade föreläsningar, lektioner, samt datorlaborationer med tillhörande muntliga redovisning. Momenten beskrivs utförligare nedan. En kursplanering för kursen, där planerat innehåll för varje tillfälle finns angivet, finns på LISAM senast från och med kursstart. Kursplaneringen utvecklas till viss del under kursens gång, framför allt detaljjusteringar av på vilket tillfälle vad behandlas.

Formellt sett har kursen inga obligatoriska **tillfällen**, men det är obligatoriskt att innan omtentan i augusti, **genomföra redovisningar** av projekt/labbar – och det underlättar för alla inklusive dig

själv – att genomföra dem vid schemalagda tillfällen för redovisningar. Tentan är självklart också obligatorisk att genomföra. Det är också synnerligen lämpligt att delta på schemalagda labbtider, då detta är de enda tider som man kan få handledning i datorprogrammen som behöver användas. Närvaro kommer att tas på labbtillfällena, och om man inte är närvarande, kan det eventuellt leda till krav på mer utförlig redovisning. Detta görs för att motverka otillåten användning av generativ AI, se mer detaljer i avsnitt 2.6.5 och i Labinfo. Självklart är övriga undervisningsmoment också högst lämpliga att aktivt delta på, för att nå den förståelse som krävs för att klara labbar och tentor.

2.4.1 Storseminarier

Storseminarierna liknar till viss del vad som är föreläsningar i många andra kurser, men det genomförs också en hel del räknegenomgångar. Djupare teoriinnehåll går igenom på det som i kursens (inspelade) föreläsningar. Det innebär att storseminarier i huvudsak ägnas åt att beskriva problemtyper som studeras inom kursen, att gå igenom Lösingsstrategier och algoritmer som behövs för att lösa problemen som studeras, och illustrera detta med exempel som löses på tavlan. Viss grundläggande nödvändig teori för Lösingsstrategier och algoritmer går igenom på storseminarierna, men de djupare detaljerna täcks på inspelade föreläsningar. .

Storseminarierna kräver i normalfallet varken specifika förberedelser eller aktivt deltagande, även om det är fördelaktigt att åtminstone översiktligt sätta sig in i vad ett storseminarium skall handla om (via kursboken), och att delta så aktivt som möjligt på storseminarierna.

Eventuellt kan rekommenderade förberedelser tillkomma under kursens gång, men storseminarierna kommer *inte* att förutsätta att rekommenderade förberedelser genomförs.

PPT-underlag för storseminarierna planeras att publiceras senast strax före storseminariet ges. Dessa är föremål för uppdateringar varje år, inklusive på hur just aktuell kurs går, och därför kan dessa tyvärr inte läggas ut i god tid före schemalagda tillfället..

2.4.2 Inspelade föreläsningar

De inspelade föreläsningarna ägnas åt mer detaljerad teorigenomgång, och t.ex. förklaring eller härledning av *varför* en viss algoritm funkar på ett visst sätt, snarare än *hur* den fungerar. Illustration av hur algoritmer fungerar erhålls på storseminarier.

De inspelade föreläsningarna är schemalagda, men det är självklart upp till var och en när man själv tycker det är lämpligast att titta på inspelningen.

Om frågor på innehållet i en föreläsning uppstår, hänvisas till att fråga i samband med annan undervisning (storseminarier, eller lektioner/labbar som Stefan håller), eller mailkontakt med Stefan. Man kan självklart också fråga lektionsassistenterna, även om de inte nödvändigtvis är pålästa på all bakomliggande teori, utan de fokuserar på problemlösning och teori/algoritmer som krävs för att lösa lektionsuppgifterna.

Eftersom de inspelade föreläsningarna kan ses vid valfritt tillfälle, kommer dessa ibland att ligga på lite obekväma tider, eller i vissa fall parallellt med andra schemalagda aktiviteter. Förväntningen är att varje student anpassar sin personliga planering så att det fungerar på bästa möjliga sätt för just hen.

Huvuddelen av de inspelade föreläsningarna är från 2024, men kan eventuellt kompletteras av några ytterligare från 2026. Vissa avsnitt av filmer inspelade under 2021, och 2025 används också.

Filmer och PPT-underlag planeras att publiceras på LISAM (i en preliminär version) fredag veckan innan föreläsningen är schemalagd, dock senast strax före föreläsningen hålls.

2.4.3 Lektioner

I början av de flesta lektioner går läraren igenom en eller ett par tentauppgifter som relaterar till lektionsinnehållet. Detta kommer i huvudsak att gälla tentafrågor på s.k. G-nivå (se avsnitt 2.6.1). Därefter är det ”fri räkning” på lektionerna. Vid önskemål kan läraren gå igenom vissa fler/andra uppgifter, eller vissa delar, för hela lektionsgruppen, eller för mindre grupper på schemalagd tid.

Det finns betydligt fler rekommenderade uppgifter, än vad som förväntas hinnas med på schemalagd lektionstid. Det är därför viktigt att du även räknar tal hemma.

Det är ungefär en ”ordinarie” lektion per vecka, men det ligger också en extra lektion varje vecka (oftast fredagar), om du vill ha mer hjälp än vad som hinns på den ordinarie lektionen och/eller om du t.ex. är sjuk på det ”ordinarie” tillfället.

Lektionerna är frivilliga, men det finns en mycket stor korrelation mellan de som inte går på lektioner och de som inte klarar tentan. Det är varmt rekommenderat att sitta på lektionen och räkna, oavsett hur man ligger till i kursen, eftersom man ju då har möjligheter att fråga i stället för att fastna för länge på småsaker, på egen hand.

2.4.4 Laborationer

I kursen ingår två omgångar av laborationer, som innehåller uppdrag kring modellering och lösning av nätverks- respektive heltalsproblem. Laborationerna genomförs i grupper, Arbetslag, om högst 4 studenter.^{2,3} All information om uppdragen som varje Arbetslag skall lösa ges i Labinfo. Anmälan till Arbetslag sker under anmälningar på LISAM.

Redovisning av labbar sker muntligt på de tillfällen som i schemat heter Redovisning, mot slutet av kursen. Anmälan till vilket av 3 möjliga pass man skall redovisa, görs på LISAM. **Om man inte är närvarande och både genomför redovisningar och lyssnar på övriga gruppers redovisningar, krävs alternativt redovisningstillfälle.**⁴ Vid förhinder ska examinator i förväg meddelas via e-post för att ett extra redovisningstillfälle ska erbjudas. Redovisningen kan också komma att kompletteras med individuell redovisning för examinatorn.

Inget samarbete får förekomma mellan Arbetslagen, när det gäller uppgifterna. Det tillåtna samarbetet är i nivå med en skriftlig tenta, dvs. inget alls. Inom ett Arbetslag är det dock både meningen och uppmuntrat att samarbeta med Arbetslagets samtliga uppdrag.

Laborationerna kan ta relativt mycket tid i anspråk, där heltalsproblemet kanske tar 60 % av tiden avsatt för momentet LAB, medan nätverksproblemen kanske tar 40 %. *Underskatta inte tiden dessa moment tar!* Det är också viktigt att vara väl förberedd enligt instruktionerna, vid labbtillfället som är schemalagt. De schemalagda labbtillfällena är enda tiden man kan få handledning i programvaran som används för att lösa labbarna/projektuppgifterna, och är därför fundamentala för att klara uppgifterna.

² Färre studenter betyder inte (nödvändigtvis) att färre uppdrag behöver genomföras.

³ Examinator förbehåller sig rätten att göra om Arbetslag, inklusive lägga till eller flytta gruppmedlemmar, för Arbetslag som är färre än 4 personer.

⁴ Om en individ inte kan närvara vid något av redovisningstillfällena för ett projekt (pga. sjukdom eller annat skäl), kan ett uppsamlingstillfälle för redovisning ordnas, men detta ligger troligen i VT2. Upplägget på ett sådant tillfälle beror på hur många studenter det berör.

Utanför kursens ordinarie undervisningsperiod (=VT1, 2026), sker ingen handledning relaterat till labbar, överhuvudtaget. Examination som inte färdigställts under ordinarie undervisningsperiod, kan endast ske i anslutning till kursens omtentatillfällen, efter överenskommelse med examinator.

Se även avsnitt 2.6.4 om fusk/plagiat, och avsnitt 2.6.5 om generativ AI i förhållande till labmomentet.

2.5 Litteratur

Kurslitteraturen består i av följande bok:

- Lundgren, J., Rönnqvist, M., och Värbrand, P: (2008) *Optimeringslära fortsättningskurs*, 3 uppl. Studentlitteratur, ISBN 9789144053141

Dessutom:

- Exempelsamling (Exempelsamling, Optimering, fk (Engevall)), som till viss del kommer att uppdateras under kursens gång. Finns tillgänglig på LISAM.
- Kompletterande material (Lektionshandledningar, Laborationshandledningar, Kompletterande övningar, PPT-bilder, mm) som kan hämtas från LISAM.

Notera specifikt att gamla tentor och tillhörande lösningsförslag **inte** räknas som kurslitteratur, annat än de uppgifter som specifikt anges som rekommenderade uppgifter. Det innebär att övriga tentor och lösningsförslag inte kvalitetssäkras utifrån ev. oklarheter i frågeställningar eller fel i lösningsförslag; och de används av studenter på "egen risk". Gamla tentor och lösningsförslag läggs dock ut på LISAM som en service till studenterna.

2.6 Examination

Kursen har följande två examinationsmoment:

Moment	Kurspoäng (hp)	Betyg
Laborationer (LAB1)	1	U/G
Tentamen (TEN1)	5	U/3,4,5

2.6.1 Tentamen TEN1

Tentaupplägget är exakt detsamma som 2025⁵. Tentafrågorna är på två nivåer, där den Grundläggande nivån (G-nivån) är frågor av grundläggande karaktär där förväntningen är att man egentligen skall kunna precis allt, för att få godkänt i kursen, men för att ge utrymme för slarvfel eller tolkningssvårigheter gäller att man skall få 75% godkänt på denna del, men också en viss minimipoäng på varje område, vilket automatiskt garanterar betyg 3. För betyg 4 eller 5 krävs, förutom uppfyllda kraven för 3:a, också lösning av ytterligare uppgifter, vanligtvis av svårare eller mer omfattande karaktär, på den så kallade Betygsnivån, B -nivån.

För G-nivån, och för B-nivån, finns kunskapskrav specificerade i kapitel 4. Notera dock, att vissa marginella justeringar av kunskapskraven kan ske under kursens gång, delvis beroende på om ev. något innehåll får ett justerat fokus i samband med den årliga omarbetningen av storseminarier och föreläsningar. Om justeringar av kunskapskraven sker, görs en uppdaterad version av kursinfo tillgänglig senast vid kursens sista storseminarium. Efter kursens sista storseminarium görs inga förändringar i kunskapskraven förrän till förstagångstentan för läsåret 26/27.

⁵ Äldre tentor än 2023 hade ett helt annorlunda upplägg, och tentor 2024-2025 hade hälften så många poäng som nuvarande upplägg på tentorna.

För godkänt på G-nivån, dvs. (minst) betyg 3 krävs poängfördelning och totalpoäng enligt nedan:

G-nivå	Nätverk	Heltal	Lagrange	DynP	Totalt
Max	10–15	10–15	5–10	5–10	40
Min	5	5	3	3	30

För högre betyg än 3 krävs förutom godkänt på G-nivån poäng enligt nedan (Notera dock att utan Godkänt på G-nivå är ev. poäng på B-nivå värdelösa (och uppgifterna rättas inte ens))

	Nätverk	Heltal	Lagrange	DynP	Totalt
Max	15–25	15–25	5–15	5–15	$(G/20)+60$
Krav för 3	0	0	0	0	$(G/20)+0$
Krav för 4	0	0	0	0	$(G/20)+15$
Krav för 5	25%*	25%*	25%*	25%*	$(G/20)+30$

*För 5.a krävs att man förutom ett krav på totala antalet poäng också får minst 25% av poängen på minst 3 av 4 områden.⁶ För 4.a skulle det kunna räcka med att man t.ex. (självklart har godkänt på G-nivån men förutom det) har i stort sett alla rätt på ett område, och några ytterligare poäng.

Det är också självklart att inslag av G-nivå också kan vara en del av frågorna på B-nivå, om det t.ex. är någon extra ”knorr” på frågan (t.ex. kan det vara problem där man har obegränsade lösningar, alternativa optimallösningar, eller saknar tillåtna lösningar, eller alternativa val som behöver göras som inte algoritmen explicit dirigerar hur de skall göras, eller ”konstiga” kostnadskomponenter), eller om det t.ex. kräver en längre kedja av korrekta beräkningar.

Frågorna på B-nivå kan var mer omfattande, lösningsgången är lika viktig som svaren, och det är sannolikt lätt att otydligheter eller slarvfel ger poängavdrag. Det kan alltså kräva mer kunskap och/eller mer säkerhet för att (hinna) samla ihop poäng på B-nivån än på G-nivån. En del frågor på B-nivå kan också vara av teori-karaktär. Dessa kräver ofta en precis och korrekt terminologi, beteckningar och matematiskt resonemang för att ge poäng, men kan å andra sidan vara ganska icke-tidskrävande om man kan dem. På B-nivån kan frågorna bygga helt eller delvis på både frågor och (korrekta) lösningar på G-delen.

Frågor på B-nivån kan också relatera till allt i boken eller på föreläsningar/storseminarier/lektioner som ingår i kursen, utan att det explicit nämns i punktlistorna i kapitel 4

Slutligen kan också frågor på B-nivå relatera till att tillämpa för kursen ingående kunskap, på helt nya områden eller helt ny kunskap. Detta kan t.ex. vara att man själv skall utveckla eller anpassa en heuristik, eller förklara hur algoritmer i kursen skulle behöva anpassas för problem som angränsar till de problem som ingått i kursen, men kanske inte studerats förut.

Vid rättning tillämpas s.k. avbruten rättning, dvs. när det står helt klart att högre betyg (inkl. från U till G) inte kan erhållas, rättas eller poängsätts inte alla uppgifter (de som inte kan påverka betyget). Studenter erbjuds att specifikt be om ytterligare återkoppling, om det önskas.

Det är tillåtet att ha med sig ett A4-blad med valfria av dig själv handskrivna anteckningar i blyerts⁷ på båda sidorna; kompletterat med överstrykningspennor⁸, samt ett lexikon HELT utan anteckningar.

⁶ 0+10+10+10 skulle t.ex. ge 5:a, medan 25+0+0+5 inte skulle ge 5:a

⁷ Det finns både pedagogiska och rättvise-skäl till detta. Fråga gärna.

⁸ Många föredrar att använda en skrivplatta/Ipad/motsv. för att göra anteckningar. Tyvärr är utskrifter från dessa inte tillåtna, utan du måste lägga tid på att skriva om det du behöver, i blyerts.

Första tentamenstillfället är fredag 27 mars 2026, kl. 8-13.

2.6.2 Laboration LAB1

Kraven för momentet *LAB1*, beskrivs i avsnitt 2.4.5.

Alla delar av *LAB1* måste avklaras helt under ett och samma läsår (inklusive omtentaperioden i augusti som följer på läsåret). Om bara vissa deluppgifter avklarats och godkänts, måste man göra om *hela* kursmomentet (hela lab-serien), vid en senare kursomgång.

2.6.3 Övergripande bedömning

Kursen är godkänd när samtliga kursmoment som ger kurspoäng är godkända. Kursbetyget är lika med tentamensbetyget.

2.6.4 Fusk och plagiat

Eftersom en stor del av arbetet med labbar sker utan övervakning, är det viktigt att förstå vad som utgör fusk och plagiat. Plagiat är ett sätt att fuska, och är kortfattat när man använder eller lämnar in någon annans arbete (inklusive utdrag ur texter), som om det vore ens egen (t.ex., att inte ange (korrekta) referenser). Det är också att återanvända någon annans text eller presentation, ord för ord, även om du anger referens. Även bilder eller programkod följer samma generella regler som text, dvs. att använda (hela eller delar av) andras programkod eller liknande är också plagiering.

All form av samarbete mellan grupper (dvs. det som kallas Arbetslag i labbar), i förberedelser såväl som genomförande är också fusk, liksom självklart att dela material mellan Arbetslag (såvida inte detta sker via examinator). Detta gäller även om man tar hjälp av andra personer, t.ex. tidigare studenter, eller tar del av tidigare studenter arbete.

Om tveksamhet råder, kontrollera för säkerhets skull med examinator om det är tillåtet eller inte.

I fusk vid tentamen ingår t.ex. all form av kommunikation med andra studenter, eller otillåtna hjälpmedel.⁹

Misstanke om fusk rapporteras till disciplinnämnden, i enlighet med lärarnas instruktioner från universitetsledningen. Observera att om misstanke om fusk förekommer gällande någon eller några inom ett Arbetslag, kan det utsätta hela Arbetslaget för anmälan till disciplinnämnden, så att disciplinnämnden får ta ställning till eventuell olika påföljd för olika medlemmar.

2.6.5 Generativ AI

Det uppmuntras att om man så önskar, använda generativ AI vid inläringen av kursens olika moment. Generativ AI är kraftfullt, och det finns flera delar av kursen där det bara det senaste året visat att AI kan stödja eller utföra många uppgifter. Emellertid är det för tidigt att lägga ned kursen till förmån för en handfull promptar, då det är oklart vad som kommer att krävas av framtida Civilingenjörer i Industriell Ekonomi.¹⁰

För material som ligger till grund för examination, dvs. det som har med labbar att göra, gäller följande:

⁹ Det har hänt att studenter haft med sig miniräknare. Det är **inte** tillåtet!

¹⁰ Jämför multiplikationstabellen som fortfarande lärs ut 50 år efter miniräknarens introduktion.

Att skapa matematiska eller grafiska modeller, och överföra dessa till programkod, syftar till att ge en förståelse i hur man dels angriper problem, dels formulerar modeller. Om man på något vis låter AI (som t.ex. ChatGPT) hjälpa till i detta (eller utföra hela delen), går man miste om en mycket viktig del i tankeprocessen som skapar lärande och förståelse, genom att man undviker olika överväganden i arbetet och eventuella diskussioner med övriga i Arbetslaget, som kanske inte nödvändigtvis blir en del av det slutliga resultatet. Användning av generativ AI för att skapa dessa delar är helt otillåtet, och betraktas som fusk. Det är också otillåtet att använda generativ AI för att beskriva de modeller man har skapat, eller för att förklara och analysera de lösningar till uppgifterna man får ut. I skalan i bilden nedan, motsvaras det av att när det gäller uppgifterna kopplade till labbar, gäller nivå 2, som tillåtet. Notera dock risken med att vid en "brainstorming" och "generera förbättringsidéer" ligger det nära tillhands att AI i kursens relativt små och förenklade uppgifter ger förslag som når ända ned till nivå 4.

Scale Levels and Descriptions

1	NO AI	The assessment is completed entirely without AI assistance. This level ensures that students rely solely on their knowledge, understanding, and skills. AI must not be used at any point during the assessment.
2	AI-ASSISTED IDEA GENERATION AND STRUCTURING	AI can be used in the assessment for brainstorming, creating structures, and generating ideas for improving work. No AI content is allowed in the final submission.
3	AI-ASSISTED EDITING	AI can be used to make improvements to the clarity or quality of student created work to improve the final output, but no new content can be created using AI. AI can be used, but your original work with no AI content must be provided in an appendix.
4	AI TASK COMPLETION, HUMAN EVALUATION	AI is used to complete certain elements of the task, with students providing discussion or commentary on the AI-generated content. This level requires critical engagement with AI generated content and evaluating its output. You will use AI to complete specified tasks in your assessment. Any AI created content must be cited.
5	FULL AI	AI should be used as a 'co-pilot' in order to meet the requirements of the assessment, allowing for a collaborative approach with AI and enhancing creativity. You may use AI throughout your assessment to support your own work and do not have to specify which content is AI generated.

Table 1 The AI Assessment Scale

Källa: <https://doi.org/10.53761/q3azde36>

När det gäller de presentationer (med tillhörande muntliga redovisningar) som skall göras för att presentera resultaten (Se Labinfo), skall utkast på presentation och eventuellt manusskrivande utföras av student(gruppen). Där är dock tillåtet att använda generativ AI för återkoppling på era presentationer (inkl. layout), och manus. Detta motsvarar att använda generativ AI på nivå 3 ovan. Om AI används överhuvudtaget där är det dock i sånt fall absolut nödvändigt att i slutet av presentationen beskriva på vilket sätt AI har använts, inklusive att skriva vilka promptar man gett till AI och vilka svar man har fått, och säkerställa att hela arbetet har utförts enligt tillåtna nivåer beskrivet ovan. Det är också viktigt att man i inlämningen på LISAM, lämnar in material som det såg ut innan man skickade till AI-återkoppling. Om inget är nämnt om AI i materialet som lämnas in på LISAM, eller om materialet före AI inte lämnas in, tolkas det som att ingen AI har använts, och att om misstanke finns att AI trots allt har använts, blir detta automatiskt en misstanke om vilseledande vid examination, dvs. fusk. Notera att olika verktyg och metoder kan komma att användas för att upptäcka misstänkt otillåtet användande av AI, även om den slutliga bedömningen alltid är manuell.

Vid misstanke om otillåtet användande av AI uppstår två olika frågor. Den ena frågan är om det är vilseledande vid examination. Om det är så, så är det ett ärende som i förlängningen bör/skall behandlas av disciplinnämnden vid LiU. Den andra frågan är om de rätta kunskaperna i ämnet har erhållits. För att säkerställa att de rätta kunskaperna har erhållits, och för att skapa en förståelse för i vilken grad vilseledning har skett, kan hela eller delar av Arbetslaget behöva göra en kompletterande redovisning/diskussion av sina uppdrag, både gällande arbetsprocess, olika val som gjorts, resulterade modeller och resultat. Detta meddelas i sådant fall i samband med granskningen av respektive slutinlämning av labbarna/uppdragen.

2.7 Gruppkontrakt

Gruppkontrakt för labben är obligatoriskt att upprätta och lämna in på LISAM, i en särskild inlämning tillgänglig vid kursstart. Gruppkontraktet skall innefatta huruvida, och i sådant fall hur, man avser att dela upp arbetet och hur avser att dela med sig av arbetet till övriga i Teamet. Det är lämpligt att även inkludera andra samarbetsrelaterade punkter i gruppkontraktet.

2.8 Alternativ examination

Möjligheten till examination utanför ordinarie kursomgång är som delvis beskrivits i kapitlen ovan. Tentor examineras bara vid de schemalagda tentamenstillfällena. Laborationer kan examineras utanför ordinarie kursomgång, i samband med de schemalagda tentamenstillfällena, men detta måste särskilt diskuteras med examinator (i andra hand med ämnets studierektor). Ingen handledning sker utanför ordinarie kursomgång.

Alternativ examination (t.ex. muntlig i stället för skriftlig, eller tvärt om) är normalt endast möjligt om man har intyg från koordinator för lika villkor som styrker att alternativ examination är nödvändig och om det också är praktiskt och resursmässigt genomförbart.

3. Undervisningsplanering

En undervisningsplan finns i form av ett Excelark, och uppdateras kontinuerligt på LISAM. Notera att det innehåller flera flikar, och möjligheten att "filtrera" för att se endast ett urval av rader.

4. Kunskapsområden

Nedan följer vilka kunskaper som är nödvändiga för att dels bli godkänd på tentan (3), dels för att nå ett högre betyg (4 eller 5).

Observera att ytterligare kunskaper än de som listas nedan; framför allt från tidigare kurser, eller från vad som bedöms som enklare moment eller problemtyper till viss del inom ramen för denna kurs, också är nödvändiga utan att det explicit nämns, och att detta får anses falla inom ramen för den sista punkten på varje område (se nedan). Som exempel på kunskaper som kan krävas på G-nivå, utan att specifikt omnämnas, är grafisk lösning av LP/HP-problem (som kan vara delsteg inom andra metoder, där examinationen ligger på förståelsen för slutsatser – även om det kan innebära att om man inte kan de enklare momenten så klarar man inte uppgiften), eller lösning av ett Heltalsproblem i flera variabler med ett kappsäcksvillkor (genom sortering av variablerna). Inte heller omnämns ju ekvationslösning, addition, eller beräkning av olikheter, som är eller kan vara en del av problemlösningen som faktiskt krävs.

En tydligare mappning av kunskaper mot kursmål kommer att göras under kursens gång, 2024 (eventuellt i ett annat dokument).

4.1 Nätverk

4.1.1 Kunskaper på G-nivå

- Lösa ett billigaste-väg-problem med Dijkstras algoritm
- Identifiera kritisk väg/kritisk linje, och förstå vad den innebär, i ett projektnätverk
- Förstå och motivera möjliga och mest lämpade metoder (Bellmans ekv., Dijkstras, Fords) i givna nätverk (utan att lösa), inklusive relaterade teorifrågor
- Modellera Minkostnadsflöden med kostnader, undre gränser (>0), övre gränser ($\leq M$).
- Modellera Minkostnadsflöden med intäkter
- Hantera modellering minkostnadsflöden där maximal kapacitet $\neq$ total efterfrågan (välja rätt nodstyrkor, samt "överskottsbågar" eller "cirkulärt flöde")
- Hantera justering av modeller för att ta hänsyn till kostnader, undre gränser eller övre gränser på flödet *genom* en nod
- Hantera justering av modeller där alternativa kostnader finns för samma väg.
- Identifiera om en given lösning är en tillåten baslösning eller varför den inte är det i ett MKF-problem
- Identifiera ett (unikt) basträd i ett MKF-problem, inkl. att förstå varför ett unikt basträd inte kan identifieras
- Beräkna nodpriser i en tillåten baslösning i MKF-problem (utan knorrrar)
- Beräkna reducerade kostnader i en tillåten lösning i MKF-problem, och utifrån dessa, både kunna avgöra om lösningen är optimal, och om inte, vilken som skulle vara en (unik) inkommande basbåge
- Givet en inkommande basbåge i ett MKF-problem, identifiera en (unik) utgående basbåge, samt beräkna nätverkets nya flöde.
- Känslighetsanalys MKF: Kostnadskrav på en ny båge för att den skall vara intressant att använda eller inte
- Känslighetsanalys MKF: Kostnadskrav/kostnadsförändringar på en existerande båge för att den skall vara intressant att använda eller inte
- I detta ingår självklart att kunna förstå begreppen som relaterar till uppgifterna (som ETT exempel förstå begreppet nodpris)!

4.1.2 Kunskaper på högre-betygs-nivå

- Förstå och redogöra för kopplingar mellan LP-modeller och Billigaste-väg-problem respektive Minkostnadsflödesproblem
- Förstå och hantera kopplingar mellan optimalitetsvillkoren för LP-problem och Bellmans ekvationer
- Förstå och diskutera konsekvenser av heltalsegenskap för nätverksproblem
- Modellera ett "helt" MKF-problem, inklusive justeringar som anges under grundläggande kunskaper, och med ev. ytterligare "knorrrar"
- Lösa ett billigaste-väg-problem med Bellmans ekvationer, eller med Fords algoritm
- Lösa billigaste-väg-typ-problem med alternativa målfunktioner; t.ex. min-max, multiplikativ målfunktion
- Modellera och lösa ett projektnätverksproblem
- Förstå och föra matematiska resonemang som relaterar till LP-teori kopplat till simplexmetoden för nätverk, inklusive primal-duala relationer och optimalitetsvillkor.

- Klara att göra hela iteration(er) i MKF-problem samt även där t.ex. basträdet inte är unikt, liksom där inkommande eller utgående basbåge inte är unik, eller t.ex. samma båge är utgående som inkommande, eller där reducerade kostnaden för en icke-basbåge är 0.
- Förstå om alternativa optimallösningar finns i ett MKF-problem, och i sådant fall hitta en sådan.
- Tillämpa det faktum att det finns en frihetsgrad i valet av ett nodpris
- Använda Fas1-metoden för Simplex för nätverk, för att hitta en tillåten lösning; både modelleringsmässigt och att genomföra en (del)iteration.
- Känslighetsanalys MKF: Kostnadskrav på en icke-basbåge för att den inte längre skall ha sitt nuvarande min- eller maxflöde
- Känslighetsanalys MKF: Kostnadskrav på en basbåge för att baslösningen skall vara oförändrad
- Känslighetsanalys MKF: Hitta nya flödet i ett nätverk om en ny båge läggs till, med en viss kostnad.
- Känslighetsanalys MKF: Hitta nytt flöde om nodstyrkor ändras i ett par noder.
- Känslighetsanalys MKF: Hitta nytt flöde och kostnadsskillnad om kapaciteten i en båge ändras (undre gräns minskar eller övre gräns ökar)
- Känslighetsanalys MKF: Totalkostnadsskillnaden i nätverket om kostnaden på en basbåge eller icke-basbåge ändras

4.2 Heltal

4.2.1 Kunskaper på G-nivå

- Formulera relationer mellan binära och linjära variabler, även med flera index
- Formulera logiska villkor mellan binära variabler, även med flera index
- Matematisk formulering av en fullständig enkel matematisk heltalsmodell (utan knorror) – definition av variabler, formulering av målfunktion, bivillkor (inklusive krav på variabler), med variabler med max 2 index.
- Förstå om en ritad eller ritningsbar olikhet är en giltig olikhet eller inte
- Identifiera ett Gomorysnitt (heltal respektive fraktionellt) ur en simplex-tablå (utan knorror, som t.ex. negativa koefficienter), och illustrera detta grafiskt i ursprungliga variabler
- Identifiera om en ny tablå är optimal, efter att en given ursprungstablå är given, ett tillagt Gomorysnitt är givet, och en ny tablå är given.
- I ett givet LDD-sökträd, identifiera vilka grenar som kan kapas och inte, samt identifiera pessimistiska och optimistiska skattningar för problemet
- I ett givet LDD-sökträd, kunna dra slutsatser hur optimistiska skattningar kan skärpas när man t.ex., har heltaliga koefficienter i målfunktionen.
- I ett problem som löses med LDD, identifiera vilka villkor som bör genereras i en förgrening av en lösning funnen i en viss nod
- Lösa Billigaste uppspännande träd (även med knorror, som t.ex. att en viss båge **måste ingå** i lösningen) till optimum med valfri metod av Prims eller Kruskals
- Lösa handelsresandeproblem med närmsta-granne-heuristiken
- Utföra enkel lokalsökning utan knorr i ett handelsresandeproblem
- Förstå skillnaden mellan/kopplingen mellan lokalt och globalt optimum
- I detta ingår självklart att kunna förstå begreppen som relaterar till uppgifterna (som ETT exempel förstå begreppet binär variabel)!

4.2.2 Kunskaper på högre-betygs-nivå

- Matematisk formulering av mer avancerad komplett heltalsmodell, inklusive variabler med många index i ett (blandat) heltalsproblem, och dessutom hantera mer komplicerade mängder
- Hantering av lagerbalansvillkor med fasta kostnader, och flera index.
- Förstå och resonera kring olika formuleringars styrka
- Förstå och resonera kring konvexa höljet av ett tillåtet område
- Identifiera och illustrera giltiga olikheter, samt förstå om en giltig olikhet är en fasett eller inte
- Identifiera ett eller flera Gomorysnitt ur en simplextablå, eventuellt med knorrar, samt lägga till ett Gomorysnitt, genomföra en iteration med duala simplexmetoden.
- Identifiera övertäckningar resp. minimala övertäckningar i kappsäcksvillkor, samt härleda giltiga olikheter ur dessa
- Kunna lyfta koefficienter i bivillkor för att generera starkare giltiga villkor
- Lösa ett problem med hjälp av LDD, och utnyttja problemspecifika egenskaper (som t.ex. om man vet att man "enkelt" hittar tillåtna lösningar t.ex. med hjälp av avrundningar, eller om man vet att optimala målfunktionsvärdet alltid är heltaligt, oavsett de lösta LP-problemens målfunktionsvärde)
- Aktivt använda kunskapen att senaste tillagda villkoret i LDD, alltid är aktivt i "nästa" problem (vilket t.ex. gör att man grafiskt kan lösa problem i 3 variabler).
- Lösa varianter av träd-problem eller handelsresandeproblem med icke-standardmässiga heuristiker (som t.ex. kan presenteras på tentatillfället, t.ex. som anpassningar på "standard"-heuristiker).
- Lägga till subtursförbjudande villkor i en TSP-lösning där subtursförbjudande villkor relaxerats
- Identifiera delproblem/grenar i trädsökningsstrategi, i en TSP-lösning där subtursförbjudande villkor relaxerats
- Lösa handelsresandeproblem med Närmsta insättning, Längst-bort insättning eller billigaste insättning
- Genomföra en eller ett par iterationer av Tabusökning, med givna definitioner
- Förstå och beskriva principer för genetiska algoritmer respektive simulated annealing, och kunna tillämpa dessa lösningsprinciper på ett mindre, enklare problem

4.3 Lagrange

4.3.1 Kunskaper på G-nivå

- Formulera Lagrangefunktionen för blandade heltalsproblem (MILP), med ett eller flera relaxerade bivillkor
- Formulera Lagrangeduala funktionen $h(v)$ för MILP, med ett eller flera relaxerade bivillkor
- Formulera alla separata Lagrangeduala subproblem då huvudproblemet är ett MILP, med ett eller flera relaxerade bivillkor
- Lösa Lagrangeduala (sub)problem när endast ett villkor har relaxerats, i ett huvudproblem som är ett MILP, för givna värden på Lagrangemultiplikatorn v
- Avgöra om en funnen lösning x , är tillåten eller inte i ursprungsproblemet i ett MILP

- Beräkna (olika) värden på $h(v)$ (för olika v) och olika lösningar x (både tillåtna och otillåtna) för att dra slutsatser om (gränser för) z^* och v^* , när endast ett villkor har relaxerats i ett MILP
- I detta ingår självklart att kunna förstå begreppen som relaterar till uppgifterna (som ETT exempel förstå begreppet LBD), liksom att kunna utföra lösning av enklare LP/MILP/heltalsproblem.

4.3.2 Kunskaper på högre-betygs-nivå

- Samma kunskaper som på G-nivå, men kunna göra detta även för nätverksproblem (TSP, MST, BVP, MKF), inklusive med sidovillkor, ett eller fler än ett bivillkor relaxeras.
- Förstå och tillämpa en Lösningstrategi i form av en iterativ process där Lagrangemultiplikatorerna uppdateras, enligt angivna sökriktning- och steglängdskriterier
- Rita men även tolka och dra slutsatser av grafiska illustrationer av $h(v)$
- Förstå begreppen subgradient och subdifferential, och även hur man kan identifiera att $0 \in \partial h(\bar{v})$, samt vilken tolkning man kan göra av det
- Förstå vad icke-styrbarhet kan få för konsekvenser när man jobbar med Lagrangerelaxering som metod

4.4 Dynamisk programmering

4.4.1 Kunskaper på G-nivå

- Lösa ett formulerat partiformnings/lagersyrningsproblem eller ett kappsäcksproblem med Dynamisk programmering utan osäkerhet, med bakåtrekursion (dvs. både ”fylla i tabeller” med nödvändiga beräkningar, och göra uppnystning för att hitta optimallösning), i problem med unik optimallösning
- Förstå när och varför det är fördelaktigare att använda bakåtrekursion i stället för framåtrekursion
- I detta ingår självklart att kunna förstå begreppen som relaterar till uppgifterna (som ETT exempel förstå begreppet överföringsfunktion)!

4.4.2 Kunskaper på högre-betygs-nivå

- Rita, förstå och tolka kopplingen mellan Nätverk och DynP
- Förstå hur Bellmans ekvationer används (implicit) i lösning av DynP
- Formulera ett DynP-problem från en text, med korrekta begrepp – även andra typer av problem än rena partiformnings/lagerstyrningsproblem
- Lösa ett DynP som kan vara andra problem än rena partiformnings/lagerstyrningsproblem som dessutom dels kan innehålla osäkerhet, dels ha alternativa optimallösningar
- Formulera och lösa DynP-problem med mer komplexa målfunktioner