



FORSVARETS
FORSKNINGSINSTITUTT

Forsvarets forskningsinstitutt

Norwegian Defence Research Establishment

BIBLIOTEKET

Postboks 25

NO-2027 Kjeller, Norway

EKS. 3

~~BEGRENSET~~

~~i h t Sikkerhetsinstruksen,~~

AVGRADERT

Dato: 2/3-16 Sign: S. Glan

FFI RAPPORT

(U) MODELL FOR FLERSCENARIO STRUKTUROPTIMERING - Modellbeskrivelse

BRAATHEN Sverre, GROTMOL Øyvind, LANGSÆTER Tor

FFI/RAPPORT-2000/04739

~~BEGRENSET~~

~~i h t Sikkerhetsinstruksen~~

FFISYS/734-161.2

~~BEGRENSET~~

Godkjent
Kjeller 30 november 2000



Jan Erik Torp
Forskningsjef

**(U) MODELL FOR FLERSCENARIO
STRUKTUROPTIMERING - Modellbeskrivelse**

BRAATHEN Sverre, GROTMOL Øyvind, LANGSÆTER
Tor

FFI/RAPPORT-2000/04739

FORSVARETS FORSKNINGSINSTITUTT
Norwegian Defence Research Establishment
Postboks 25, 2027 Kjeller, Norge

~~BEGRENSET~~

FORSVARETS FORSKNINGSINSTITUTT (FFI)
Norwegian Defence Research Establishment

UNCLASSIFIED

P O BOX 25
NO-2027 KJELLER, NORWAY

SECURITY CLASSIFICATION OF THIS PAGE
(when data entered)

REPORT DOCUMENTATION PAGE

1) PUBL/REPORT NUMBER FFI/RAPPORT-2000/04739	2) SECURITY CLASSIFICATION RESTRICTED	3) NUMBER OF PAGES 91
1a) PROJECT REFERENCE FFISYS/734-161.2	2a) DECLASSIFICATION/DOWNGRADING SCHEDULE 5 YEARS	
4) TITLE (U) MODELL FOR FLERSCENARIO STRUKTUROPTIMERING - Modellbeskrivelse (U) Model for multi-scenario structure optimization - Model description		
5) NAMES OF AUTHOR(S) IN FULL (surname first) BRAATHEN Sverre, GROTMOL Øyvind, LANGSÆTER Tor		
6) DISTRIBUTION STATEMENT Distribution limited. (Spredning begrenset iht. Beskyttelsesinstruksen eller Offentlighetsloven)		
7) INDEXING TERMS IN ENGLISH:		IN NORWEGIAN:
a) <u>Operations Analysis</u>		a) <u>Operasjonsanalyse</u>
b) <u>Force structure analysis</u>		b) <u>Styrkestruktur analyse</u>
c) <u>Optimization</u>		c) <u>Optimering</u>
d) <u>Mathematical programming</u>		d) <u>Matematisk programmering</u>
e) <u>Scenario aggregation</u>		e) <u>Scenarioaggregering</u>
THESAURUS REFERENCE:		
8) ABSTRACT This report describes a model for cost-effective force structuring analysis using mathematical programming and optimization with game-theoretic utility measure. Force components including C2 and sensor/information are included in a multi-scenario setting to analyse a best overall expected force structure within given budget cost constraints. Each scenario uses a graph with nodes and connections having characteristics describing terraintypes, axis capacities and distances. Components manoeuvre and Lanchester type engagements take place in graph nodes using effectiveness data from lower level models. A main challenge for model development is the representation of strategies for application of game-theoretic methods with maximin solutions. Detailed user specified strategies, node restricted strategies as well as model optimized strategies are possible. A special algorithm for alternating side optimization is developed for approximate maximin mixed strategy solutions in an iterative loop for each scenario. Maximum expectation over all scenarios gives the resulting force structure. The model is developed using the AMPL mathematical programming tool with solver MINOS5.5 and a special heuristic solver HOTDOG also developed as part of the model.		
9) DATE 30 November 2000	AUTHORIZED BY This page only Jan Erik Torp	POSITION Director of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE
(when data entered)

INNHold

	Side	
1	INNLEDNING	7
2	MODELL	9
2.1	Generelt	9
2.2	Matematisk modell	10
2.2.1	Algoritme/metode	11
2.2.1.1	Struktur	11
2.2.1.2	Innfasing av struktur	14
2.2.2	Modellvariable	14
2.2.3	Effektivitetskriterium/målfunksjon	17
2.2.4	Beskrankninger	19
2.2.4.1	Scenario/Trussel	20
2.2.4.2	Kostnad	21
2.2.4.3	Operasjonskonsept / oppdrag	22
2.2.4.4	K2 og informasjon	22
2.2.4.5	Dynamikk og manøver	24
2.2.4.6	Taktisk støtte	27
2.2.4.7	Logistikk	28
2.2.4.8	Strukturavhengigheter / arv	28
2.2.5	Gitt struktur / konsekvensanalyse	28
2.2.6	Kompleksitet	29
3	DATAGRUNNLAG OG RESULTATBEHANDLING	30
3.1	Kostnadsdata	30
3.2	Effektivitetsdata	31
3.3	Resultatbehandling	31
3.3.1	Resultatfiler	32
3.3.2	Excel resultatbehandling	33
4	KONKLUSJON	34
A	MODELL I AMPL	37
A.1	Sannsynlighetsmodell for målfunksjon	37
A.2	Modell i AMPL	39
A.2.1	Modelldefinisjon: Mengder, variable og relasjoner	39
A.2.2	Beskrankninger	47
A.2.3	Kjørefil	49
A.3	HOTDOG/MICIO løsningsverktøy	52

A.3.1	Algoritme for HOTDOG/MICIO	53
A.3.2	HOTDOG mal for modellutvikling	62
A.4	Eksempel	65
A.4.1	Datafil	65
A.4.2	Kjøreoppsett	70
A.4.3	Resultatfiler (utdrag)	71
A.4.4	Excel resultatbehandling	80
A.5	MIKS: Modell for innfasing av kosteffektiv struktur	82
A.5.1	Eksempel på MIKS resultatfil	85
	Litteratur	88
	Fordelingsliste	91

(U) MODELL FOR FLERSCENARIO STRUKTUROPTIMERING - Modellbeskrivelse

1 INNLEDNING

Dette er en beskrivelse av en generell analysemodell/metodikk for styrkestrukturering basert på kost/effektivitet som er utviklet i prosjekt 734/161 "Operativt konsept for landstrid". Prosjektets målsetning har vært å utvikle et metode/modell-hierarki for analyse av landoperasjoner og -strukturer under hensyn bl a til ønsket om at:

- strukturanalyser skal bygge på kvantitative kost/effektivitetsberegninger
- metode/modellutvikling for strukturanalyser på høyt nivå skal bidra til økt analyseomfang (dvs inkludere f eks funksjoner som K2, IRS, og tosidig handlemåte-valg)
- fler-scenario analyse skal kunne inngå som grunnlag for strukturanbefalinger

En samlet vurdering av disse hensyn peker i retning av optimeringsteori som metode for å kunne behandle det store antall faktorer og sammenhengene mellom disse med et rimelig tidsforbruk. Nyere verktøy og ny datamaskinteknologi har etterhvert gjort bruk av optimering som metode mulig i større grad, selv om det vil være nødvendig med en vel avbalansert detaljeringsgrad i de problemstillinger som kan modelleres. Dette metodevalg vil også bidra til metodeutvikling innenfor langtidsplanlegging ved å utnytte teori og verktøy innen feltet matematisk optimering (som benyttes i operasjonsanalyse i mange sammenhenger forøvrig). Derved vil arbeidet i et lengre perspektiv generelt kunne bidra til et utvidet spenn i strukturanalyser ved å fange opp flere variasjoner enn det som en mer konvensjonell 'manuell' analyse av tidsmessige årsaker vil kunne.

Den analysemodellen som er beskrevet er i prosjektet benyttet som hjelpemiddel for å utprøve idéer og forslag i praksis basert på de scenarier og oppgaver som prosjektet har behandlet. Modellen er således også et resultat i seg selv og inngår som en del av det modellhierarkiet som prosjektet har utviklet.

Modellen tar bl a hensyn til forflytninger og oppdragsvariasjon i definerte scenarier, slik at også disse dimensjonene vil kunne variere. Dette gir mulighet for å ta hensyn også til operasjonskonsept for styrker i scenarier, men antall dimensjoner medfører samtidig økt kompleksitet i modellen. En for stor kompleksitet vil kunne gjøre det umulig å finne kost/effektive optimale struktursammensetninger innen rimelig beregningstid, og en utfordring er derfor å balansere graden av detaljering opp mot en tilstrekkelig realistisk beskrivelse av problemet. Analysemodellen skal kunne benyttes til følgende problemstillinger :

- styrkestrukturanalyse for et sett av scenarier (evt også inkludert internasjonale operasjoner) med gitt kostnadsramme (helst direkte kosteffektiv struktur gitt både effekt- og kost-beskrivende (aggregerte) funksjoner som del av analysemodellen)
- styrkedimensjonering i et gitt scenario (dvs kosteffektiv komponentstruktur, helst også direkte som del av kosteffektiv styrkestrukturanalyse over)

- evaluering av “beste” handlemåter i et scenario for en gitt styrkestruktur (basert på (kanskje flere) effektivitetsmål, også for manøverorienterte operasjoner)

Hovedoppgaver som utfra utkast til Allied Joint Operations Doctrine, nasjonalt doktrineutkast og prosjektdiskusjoner skal kunne gjennomføres vil være :

- **Invasjonsforsvar/NATO Art V** : C2W, bekjempe luft-,land-, og sjømål innen operasjonsområdet, utføre manøverorienterte fellesoperasjoner, utføre støtteoperasjoner
- **Krisehåndtering** : Ledelse, Etterretning, Overvåkning og kontroll, Sikring, Anti-terror
- **Internasjonalt militært engasjement** : Reaksjonsstyrke bidrag, Non Art V (CJTF) bidrag

Noen av de kapasiteter som forutsettes av oppgavene vil være :

- Ledelse i krise og krig
- Samvirke nasjonalt og internasjonalt
- Slå
- Manøvrere (strategisk, operasjonelt og taktisk)
- Overvåke

I modellen betraktes ledelse som en styrkekomponent på linje med andre komponenter. Fordi kommando og kontroll antas å få økt betydning i fremtidige styrkesammensetninger vil en kost/effektivitetsbetraktning også av denne funksjonen være av interesse.

Det følgende er en beskrivelse av en analysemodell som søker å innlemme flest mulig av de funksjoner og oppgaver som er beskrevet over. En matematisk programmeringsmetodikk er valgt for å konsentrere beskrivelsen til en analysemodell med de viktigste faktorer inkludert. Det er ikke klart at analysemodellen kan løses med tilgjengelige standardmetoder, og mulige heuristiske løsningsmetoder kan derfor være nødvendige. Analysemodellen er en initiell skisse og må videreutvikles med variable og relasjoner etterhvert som ny kunnskap innenfor ulike delområder etableres (1)(2)(3)(4)(5)(6)(7)(9)(10)(11). Andre anvendelser av optimering i ulike struktursammenhenger omfatter modeller som spenner fra generell deployering av styrker til ressursallokering og planlegging av kampanjer (8)(12)(15)(18)(19)(20). Det er også beskrevet en tilnærming til strukturanalyse/optimering der risiko-vurdering over et sett av oppgaver (“tasks”) og styrkestrukturer er benyttet (17). Den analysemodell som her er beskrevet er et forsøk på å optimere struktur for komponentstyrker med støttekomponenter i et gitt scenario og tilpasset for også å optimere over f eks en sannsynlighetsfordeling av scenarier for å finne en samlet og mest mulig robust struktur (13)(14)(15)(21)(25)(26)(27). Med struktur menes her kun antallet og settet av komponenter i strukturen; implementering av en struktur i løpet av en planperiode utføres vha en annen modell, f eks basert på optimering av anskaffelse over planperioden. Det er også utviklet en enkel delmodell for optimering av slik innfasing av en optimal struktur under hensyn til kostnad og teknologisk fordyrelse av materiell.

Analysemodellen for strukturoptimering kalles MSO².1 (Modell for StrukturOptimering i Multi-Scenario Operasjoner, versjon 1) og innfasingsmodellen MIKS (Modell for Innfasing av Kosteffektiv Struktur).

Rapporten inneholder en generell problembeskrivelse med detaljert diskusjon av de ulike

funksjoner i modellen i kapittel 2. I kapittel 3 beskrives datagrunnlag og resultatutskrift fra et eksempel med bruk av modellen. Konklusjon og mulig videreutvikling er angitt i kapittel 4. Et sammendrag av modellen med eksempeldata er vist i Appendiks A sammen med et uttrykk for sannsynligheten for å lykkes i et scenario som kan benyttes som effektivitetskriterium eller nyttefunksjon/målfunksjon i modellen.

2 MODELL

2.1 Generelt

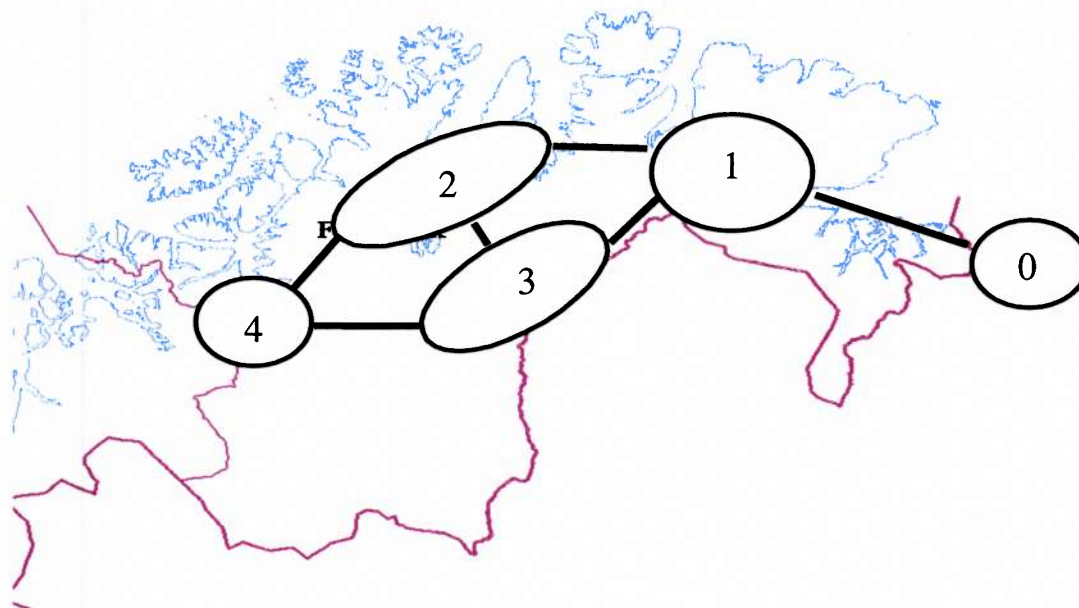
Analysemetodikken i MSO².1 baseres på spillteori, der strategier for hver side kan velges utfra et potensielt sett av oppdrag innen faser/perioder mot motstanderenheter i mulige noder i en terreng-graf. Engasjements-resultat i aktive noder kan finnes (helst som sannsynlighet for å vinne = Prob (vinn | node), se (2)(4)(5)(9)), evt også med avbruddskriterier (se samme og (11)) som grunnlag for nye engasjement i neste fase/periode. En spillteoretisk maximin-løsning er kun unntaksvis mulig å bestemme pga det store antall mulige strategier, og en metode for å finne en tilnærming til maximin-løsningen er beskrevet i kapittel 2.2.

En kost/effektivitetsmodell basert på optimering innebærer automatisk søk etter maksimalverdi av et kvantitativt mål i hvert scenario, f eks sannsynlighet for å vinne (basert på styrketap utfra effektivitetsdata). Fler-scenario analyse kan da integreres f eks som maksimal forventningsverdi over scenariene med gitt sannsynlighet (prioritet) for hvert scenario, og der kun én og samme startstruktur skal benyttes i alle scenariene (prioritet av hvert scenario må angis, men kan selvfølgelig varieres for å se virkning på struktur). Fordi det kvantitative mål og/eller beskrankninger i modellen har ulineariteter, vil en optimering også måtte ta hensyn til at maksimalverdien kan være lokal, og at et søk etter globalt maksimum må inkluderes. Dette er nærmere detaljert i kapittel 2.2.1.

Dynamikk og manøver skjer i terrenggrafen som er beskrevet ved et antall noder forbundet med akser som vist på Figur 2.1 nedenfor. Aksene kan beskrives med mobilitetskapasiteter, deteksjonsforhold, kuperingsgrad, etc. Grafenes mobilitetskapasiteter og komponent-enheters ytelse kan reduseres ved mottiltak som KOMØD, C2W, etc, og dette er beskrevet som del av ledelse/informasjon i kapittel 2.2.4.4. Samband mellom HK- og komponent-enhetene kan også evt beskrives i grafen med kapasiteter (tilsvarende TADKOM, MRR, HF, satellitt, etc), men dette er ikke med i første versjon av modellen.

Effekten av informasjon og K2 kan tilnærmes ved at komponenters effektivitetsdata gjøres avhengig av K2- og ISR-ressurser basert på analyse fra lavere nivå detaljmodeller (1)(10)(22)(23)(30)(31). Ved å inkludere oppdrag mot K2- og oppklaringskomponenter (C2W) vil det derved bli mulig å balansere en struktur også under hensyn til ledelse og informasjon. Dette er nærmere beskrevet i kapittel 2.2.4.4.

For å bestemme en optimal struktur må realistiske budsjettammer angis for materiellrelatert investering og drift. Aggregerede kostnads-funksjoner for alle komponenter må også inngå for å bestemme beskrankede optimeringsløsninger (17)(18)(19)(20)(26)(27).



Figur 2.1 *Terrenggraf med noder og akser. Noder er områder med terrengegenskaper og er knyttet sammen med akser med mobilitetskapasiteter.*

Scenarier vil i hovedsak være bestemt av trusselen: d v s antall og type av motstander-komponenter, terreng-grafens egenskaper, ROE (“Rules of engagement” utfra politisk nivå, NON art V oppdragsbegrensninger i fler-side konflikter), mens egen struktur kan finnes fra det maksimalt mulige settet av komponenter, dog begrenset av kostnadsrammer og ROE, fordi Maximin løsningen med beskranket kostnad vil kunne bestemme optimal egen sammensetning. Således vil ulike scenarier kunne tilpasses med rimelig små endringer, forutsatt at effektivitetskriterier tilpasset hvert scenario er gitt, men felles kriterier for hvert scenario antas foreløpig. Dersom en sannsynlighetsfordeling for ulike scenarier kan spesifiseres, kan en forventet “robust” løsning finnes som tar hensyn til alle scenarier ved ulike algoritmer (13)(14)(15)(21)(25)(26)(27)(28). Det kan forøvrig også være mulig å angi usikkerheter i alle scenarier ved å benytte stokastisk programmering, f eks ved å kreve at beskrankninger skal gjelde med en gitt sannsynlighet og ikke absolutt (18), eller ved å ta med risiko-ledd (varians) i effektivitetskriteriene (17)(25)(29).

Tilsammen gir dette en metodikk som kan anvendes for de problemstillinger som ble krevet innledningsvis. Behandling av K2 er mest usikker, ihvertfall hva angår en Maximin-løsning utfra et spillteoretisk grunnlag. I det følgende konkretiseres betraktningene i større grad matematisk, der teori og resultater fra ulike kilder er benyttet.

2.2 Matematisk modell

I det følgende er ulike aspekter ved analysemodellen MSO^{2.1} nærmere definert og beskrevet. Modellen setter sammen ulike komponenter til en samlet struktur basert på effektiviteter og kostnader. En komponent er en “brikke” som kan defineres i forhold til problemstilling og

scenarionivå (f eks Ibn, Mekbn, Pbn, FAbn, Hk, Okp, Fly-”pakke”, etc), og den samlede strukturen vil bestå av et optimalt antall av de ulike typer komponenter. Hver komponenttype kan evt selv være resultat av studier hvor en tilsvarende metodikk benyttes på lavere nivå, eller være basert på forslag til strukturelementer fra delanalyser forøvrig.

I det følgende gis en oversikt over algoritme, variable, symboler og beskrankninger som tilsammen utgjør modellen.

2.2.1 Algoritme/metode

Utgangspunkt for analysemodellen er et scenario, her definert som en gitt motstander's OOB med utgangsgruppering i en scenariograf. Gitt en prioritet eller sannsynlighetsfunksjon p_s over et sett av scenarier, kan et fler-scenario analyseproblem løses vha algoritmer som f eks beskrevet i (13)(15)(21)(25)(26), der forventet ytelse over alle scenarier optimeres (evt også med tillegg for risikobetraktninger utfra varians eller ved mer generelle nyttefunksjoner). Selv om slike prioriteter eller sannsynligheter kan være vanskelige å anslå, vil det være enkelt å variere disse for å belyse strukturkonsekvenser og komponenters marginale bidrag og påvirkning.

Analysemodellen består egentlig av to deler, hvor den første bestemmer selve strukturen, mens en annen etterfølgende del behandler optimal innfasing av materiell i den optimale strukturen over en periode (f eks 20 år). En beskrivelse av disse to delmodellene er gitt i det følgende.

2.2.1.1 Struktur

For å optimere over et sett av scenarier som forekommer med gitte sannsynligheter, p_s , kan det overordnede problemet mer generelt beskrives slik utfra en spillteoretisk betraktning:

Finn den egne struktur X som optimerer forventet spillverdi over alle scenarier, der spillverdi $V_s(X)$ i hvert scenario bestemmes av en Maximin blandet strategi med sannsynlighetsvektor P over settet av strategier $S_{m,X}$ og $S_{n,Y}$ for egne og motstanders styrker utfra en målfunksjon (effektivitetskriterium) $f_s(S_{m,X}, S_{n,Y})$ i hvert scenario.

Matematisk kan dette uttrykkes som følger:

$$\begin{aligned} & \text{Max}_X \sum_s p_s V_s(X) \\ V_s(X) &= \text{Max}_P \text{Min}_n \left(\sum_m P_m f_s(S_{m,X}, S_{n,Y}) \right) \end{aligned}$$

der P_m er sannsynlighetene i den blandede strategien P . Den optimale strukturen X består av en mulig arv og et antall komponenter x_i som skal være felles for alle scenarier, mens motstander OOB Y tilsvarende består av et initielt antall komponenter y_j som kan variere mellom scenarier, men som er fast i hvert av scenariene. Dersom man har alle mulige strategier representert vil

den spillteoretiske løsningen gi en struktur X som garanterer spillverdien uavhengig av motstanderens strategivalg i hvert scenario. Dette er kun teoretisk mulig og tilnærmede løsninger vil opptre i praktiske tilfelle. Dersom et stort nok antall strategier inkluderes for motstander vil en god tilnærmede likevel kunne oppnås.

Som nevnt over kan også optimeringen utvides med risiko-betraktninger over ulike scenarier der f eks varians- eller avviklsledd (av typen $\sum p_s V_s^2(X)$ eller $\sum p_s |V_s(X)|$) kan tas med vektet med en parameter, r , for risikovillighet (25)(26)(28)(29). Den utvidede optimeringen kan da f eks uttrykkes slik:

$$\text{Max}_X \left[r \cdot \sum_s p_s V_s(X) - (1-r) \cdot \sum_s p_s |V_s(X) - \sum_i p_{s'} V_{s'}(X)| \right]$$

$$V_s(X) = \text{Max}_p \text{Min}_n \left(\sum_m P_m f_s(S_{m,X}, S_{n,Y}) \right)$$

der absoluttavviket fra forventningsverdien er inkludert vektet med graden av risikovillighet.

I det følgende er optimering for et scenario beskrevet, og en separat indeks over alle scenarier er derfor ikke tatt med, men den inngår i den totale modellen som er listet i Appendiks A. En egen brukerveiledning for denne modellen er beskrevet i (37).

Som styringsfaktor for optimering innen et scenario, vil allokering av ulike oppdrag og manøvrering mellom noder være de grunnleggende dynamiske element i modellen. En av utfordringene vil være å opparbeide settet av representative strategier for hver side som grunnlag for en Maximin optimal struktur som angitt ovenfor. For dette formål tillater modellen tre typer strategivalg – detaljert, begrenset og fritt. Detaljert strategivalg vil si å spesifisere alle variabelverdier til alle tider i scenariet, noe som er krevende på grunn av kravet til konsistens og mengden variable. Begrenset strategivalg er enklere og kan spesifiseres ved å angi delmengder av noder i scenariografen som styrkene kun tillates å benytte, mens fritt strategivalg overlater til modellen selv å optimere alle de frie variable inklusive oppdragsvalg og manøver mellom noder.

Dersom man kan benytte kun detaljerte strategier vil en Maximin-løsning kunne finnes ved optimering som et matrisespill. Med begrensede og frie strategier, som er enklere å spesifisere, kan man finne en løsning ved iterativ optimering i løkke der hver side alternativt optimeres med den andre side holdt konstant fra forrige løkke. Dersom man optimerer mot alle tidligere strategier, vil dette kunne konvergere mot en Maximin optimal løsning. Dette er hovedidéen for modellen, selv om antallet strategier må begrenses på grunn av krav til lagringsplass og en Maximin optimal løsning derfor ikke kan garanteres. Modellens algoritme for å finne en Maximin optimal struktur kan i stort beskrives som følger:

Initialisering:

1. Spesifiser settet av egne og motstanders typer av komponenter med effektivitetsdata for de oppdrag som kan utføres, samt komponentkostnader og samlet kostnadsramme for egne styrker. Definer et scenario, dvs motstanders OOB og utgangsgruppering med en scenariograf bestående av et sett av noder og akser med avstands- og kapasitetsdata.

- a. Optimer egen struktur inklusive blandingen av strategier uten motstander i scenariografen.

Iterativ løkke:

2. Hold egen startstruktur og blandede strategi i strategisettet fra forrige løkke fast.
 - a. Optimer en og en av strategiene i motstanders strategisett med resten av strategiene holdt konstant mot egen faste startstruktur og blandede strategi fra forrige løkke.
 - b. Finn deretter optimal blandet løsning for motstander utfra matrisen av optimerte strategier fra punkt 2a.
3. Hold motstanders startstruktur og blandede strategi i strategisettet fra punkt 2b fast.
 - a. Optimer egen blandet løsning for strategisett og struktur simultant mot motstanders faste startstruktur og strategisett fra punkt 2b.

Test:

4. Gjenta punkt 2 og 3 inntil konvergens og/eller et maksimalt antall iterasjoner.

Maximin valg:

5. Velg den struktur fra punkt 2b blant alle iterasjonene som gir størst egen effektivitet når motstander optimerer. Ved konvergens er dette den siste 2b-iterasjonen, ellers må det søkes blant alle 2b-iterasjonene. Med et stort nok antall motstanderstrategier inkludert vil en tilnærmet korrekt Maximin-løsning oppnås.

For lineære kriterier og beskrankninger vil optimum være enkelt å bestemme i hver iterasjon. Med ulineære kriterier og beskrankninger vil algoritmen for optimering i hver iterasjon søke etter globalt optimum. Det kan da generelt ikke garanteres at globalt optimum finnes, men det er laget løsningsalgoritmer som søker etter flere lokale optimum, og deretter bruker det beste av disse som estimat av globalt optimum. En slik løsningsalgoritme, kalt HOTDOG (Heuristisk Oppskrift Til å Determinere Optimum Globalt / Heuristic On The Determination of Optimum Globally), er utviklet som et tillegg til modellen og er kort beskrevet i Appendiks A (se også (37)). Denne algoritmen har vist seg rask og robust også for store ulineære problemer.

Selv om konvergens til en Maximin optimalløsning ikke kan garanteres i det generelle tilfelle, vil de strukturene som er generert gi verdifull informasjon om komponenters relative egenskaper og marginalbidrag utfra både hyppighet av de ulike komponenters forekomst i strukturene og deres deriverte som uttrykker hvordan effektiviteten påvirkes av endring i komponentmengden. Duale variable gir også informasjon om de bindende beskrankninger og hvordan effektiviteten påvirkes av endring i disse. Tilsammen gir dette viktig informasjon om hovedkarakteristika ved komponenter og ressursbeskrankninger i de aktuelle scenarier som kjøres.

Ulike effektivitetskriterier eller målfunksjoner kan benyttes for optimering i algoritmen i punkt 1-5 over; for to-sidige scenarier er et uttrykk for sannsynlighet for å lykkes (Prob(egen vinn)) benyttet (se Appendiks A), mens fler-sidige scenarier bruker en vektet og prioritert lineær styrkedifferanse mellom ulike sider. Det er meget enkelt å endre disse funksjonene for å belyse virkningen av andre valg av effektivitetskriterier, noe som er en stor fordel ved denne type optimeringsmodeller.

2.2.1.2 Innfasing av struktur

Etter optimeringen av struktur kan man også se på innfasing av komponenter og tilhørende materiell over en strukturperiode (f eks 20 år) under hensyn til ulike teknologisk fordyrelse av materiell. Fra KOSTMOD (40) vil grunnpriser og rater for teknologisk fordyrelse samt mengden av materiell være gitt. Dermed kan man se på minimering av total innfasingskostnad for materiellet i strukturen som et nytt optimeringsproblem, som er et enkelt lineært problem å formulere når optimal struktur er bestemt og kun en innfasingstakt med minimal total kost skal bestemmes. Det er valgt å separere innfasingen fra strukturoptimeringen i forrige avsnitt, men det er ikke annet enn en økt kompleksitet som er konsekvensen av å bake også innfasing inn i den totale strukturoptimeringen i forrige avsnitt.

Et forslag til en separat lineær optimeringsmodell MIKS for å minimere total kostnad for innfasing av materiell i optimal struktur er gitt i Appendiks A. En problemstilling som berører samtidig effektivitetsøkning som følge av teknologisk fordyrelse (med evt tilsvarende trusselendring) er ikke behandlet her, men vil kunne tas med i strukturoptimeringen dersom relasjoner og data er gitt.

Driftskostnader kan også relativt enkelt tas med som tillegg til strukturoptimeringen i forrige avsnitt gitt antall personell av ulike kategorier og andel av øving, opplæring, B&A, etc. fordelt relativt på de enkelte strukturkomponenter fra KOSTMOD (f eks noe tilsvarende som i en driftsanalyse foretatt tidligere i FA2000).

En fullstendig versjon av modellene for hhv optimering av struktur MSO².1 og for strukturinnfasing MIKS i modellspråket AMPL (38) er gitt i Appendiks A.

2.2.2 Modellvariable

I det følgende gis en beskrivelse av variable og relasjoner som er benyttet i analysemodellen. Beskrivelsen er inndelt i beslutningsvariable (de som skal optimeres), konstanter/parametre (som er faste inputdata), mengder med indekser (som grupperer beslutningsvariable), og selve modellen som består av målfunksjon(er) og beskrankningene. Her gis kun en oversikt over hovedtrekkene - en fullstendig oversikt over alle variable og parametre er gitt i Appendix A.

I selve modellformuleringen i språket AMPL er det av hensiktsmessighet benyttet kun én startstyrke-variabel med en peker til den side styrken tilhører ("*start(type(side), område*"). For å lette beskrivelsen av modellen er det i det videre skilt mellom sider ved å bruke *x* for egne og *y* for motstanders startstyrke. Selv om en scenario- og en strategi-indeks også inngår i modellens variable, er disse også sløffet i beskrivelsen nedenfor fordi de alltid vil inngå.

Modellen i AMPL inneholder også interne hjelpevariable og –mengder som er innført kun for å lette uttrykkene, og disse er heller ikke tatt med her (se også (37)).

Beslutningsvariable

Analysemodellen MSO^{2.1} skal bestemme en startstruktur med disponering over områder og oppdragstyper for hvert tidspunkt som scenariet inneholder. For å begrense mengden variable er det innført et minimumsantall frie beslutningsvariable som skal optimeres. Øvrige variable som inngår i relasjoner er definert vha interne deklarasjoner i AMPL, og vil derfor kunne kalkuleres utfra de frie variable til enhver tid. Dette er en nyttig egenskap i AMPL som er utnyttet maksimalt for å begrense kompleksiteten så mye som mulig (37).

De frie typer beslutningsvariable som må være med er startstruktur, andel av de gjenværende bakkestyrkene som flytter mellom områder og andel som utfører de enkelte oppdrag, samt andel av de gjenværende flystyrker som utfører bakkeoppdrag i de enkelte områder til enhver tid. Modellen inkluderer også arv av komponenter som faste startstyrker som ikke optimeres, slik at en realistisk tilpasning utfra en gitt status er mulig.

- x_{iot} = antall egne komponenter av type i som er i område o ved tid t ;
antall ved $t = 1$ gir egen struktur (arv kan tas med og påvirker ikke total kostnad)
- y_{jot} = antall motstanderkomponenter av type j som er i område o ved tid t ;
antall ved $t = 1$ gir OOB og trussel-nivå som input.
- $xfr_{ioo't}$ = andel av x_{iot} som flytter til et annet område o' (sum ≤ 1)
- $yfr_{joo't}$ = andel av y_{jot} som flytter til et annet område o' (sum ≤ 1)
- $x\alpha_{ioht}$ = andel av x_{iot} med oppdragstype b (sum = 1)
- $y\alpha_{joat}$ = andel av y_{jot} med oppdragstype a (sum = 1)
- $xac_{ioo't}$ = andel av x_{iot} (i = egne fly) som utfører oppdrag mot områder o' (sum ≤ 1)
- $yac_{joo't}$ = andel av y_{jot} (j = motstanderfly) som utfører oppdrag mot områder o' (sum ≤ 1)
- px_m = sannsynlighet for strategi m i egen blandede strategi (sum = 1)
- py_n = sannsynlighet for strategi n i motstanders blandede strategi (sum = 1)

Dette utgjør tilsammen de frie beslutningsvariable som skal optimeres med den algoritmen som er beskrevet i kapittel 2.2.1.1. Tilsammen vil dette kunne utgjøre flere tusen frie variable i et representativt scenario. Som angitt over er det flere variable som er definert i modellen utfra de frie variable, og disse er samlet i en felles-variabel *calc* som beskrives for de enkelte beskrankningene i de senere respektive underkapitler der de benyttes.

Parametre

Modellen krever en rekke inputdata som inngår som konstante parametre. De viktigste av disse omtales i det følgende (en samlet oversikt er gitt i (37)). Effektivitetsdata og engasjementsrater kan finnes fra modeller på lavere nivå for de komponenter som inngår. Vekter for ulike komponenter kan baseres på teori om ideelle lineære vekter som f.eks egenverdi-teori eller tilsvarende (3)(17)(19)(20).

p_{jioah} = Effektivitet pr tidsenhet av motstander komponent type j med oppdrag a mot egen komponent type i med oppdrag b i terrenntypen i område o (22).

k_{jioah}^p = Engasjementsrate av motstander komponent type j med oppdrag a mot egen komponent type i med oppdrag b i terrenntypen område o

q_{jioah} = Effektivitet pr tidsenhet av egen komponent type i med oppdrag b mot motstander komponent type j med oppdrag a i terrenntypen område o

k_{jioah}^q = Engasjementsrate av egen komponent type i med oppdrag b mot motstander komponent type j med oppdrag a i terrenntypen i område o

w_i, w_j = Relativ vekt av egen komponent type i og motstander komponent type j basert på f eks egenverdieregninger eller tilsvarende i modeller på lavere nivå

b = Anslått årlig materiellrelatert investerings- og driftsbudsjett. Kan også enkelt omgjøres til gjennomsnittlig budsjettramme over en planperiode (som i KOSTMOD). Med en nøkkel for relativ fordeling av fredsdrift på komponenter kan totalbudsjett inklusive fredsdrift også inkluderes.

c_i = Anslått årlig materiellrelatert kostnad (dvs forenkling av KOSTMOD inklusive teknologisk fordyrelse og materiellrelatert drift) for en komponent av type i . En mulig versjon pr komponent kan f. eks. være som i (17) (sum av årlige kapitalkostnader normert for levetid pluss årlige driftskostnader pga personell-kategorier og fredstidsaktivitet). Det er også enkelt å benytte samme kostnadsvariable som i KOSTMOD direkte, og med tilhørende budsjett gitt av årlig materiellrelaterte investerings- og driftskostnader. Fredsdrift kan også inkluderes som for budsjett dersom personellbehov, skole og øvingsandel og felleskostnader kan fordeles etter en relativ nøkkel på komponenter.

Mengder og indekser

Komponenter, oppdrag og områder i terrenntypen i et scenario er deklartert som elementer i mengder. Hovedtyper av mengder er :

- ty = Mengden av komponenttyper med delmengder :
 $man=manøver, fir=ildstøtte, c2, isr=ledelse, oppklaring, ac, aac=fly, luftvern$
- ta = Mengden av oppdragstyper (for komponenter)
- ar = Mengden av områder (noder) i terrenntypen
- ad = Mengden av naboer til hvert område i terrenntypen (gir forbindelsene)
- tr = Mengden av terrenntyper for områder (noder) i terrenntypen

Elementene i disse mengdene angis med indekser som vist for de frie variable og parametrene over. De viktigste indeksene er:

- i, j = Komponenttyper : C2, ISR (oppkl, artjag, UAV, JSTARS, Satellitt) , Manøver (Inf,

Mek, Pn, Para/LL, Jeger), Ildstøtte (FA, Rak, Missil, Heli, LV, UCAV), Luft (fly/heli uten OCA). Mobilitet/Beskyttelse (Ing, Tren, Helitransport)), Logistikk (Fors), Amfibiske (MEB), Spesialstyrke er mulige tilleggstyper som ikke er med i denne versjonen av modellen. Det diskuteres kort hvorledes disse kan inngå i forbindelse med beskrivelse av K2/ledelse nedenfor.

a, b = Oppdragstyper : C2W, Bekjempe/Avskjære, Sinke/Stanse og AA for manøver og ildstøtte/luftvern; DCA, Anti-Surface, SEAD for fly. Mulige tillegg kan være Anti-mobilitet (KOMØD/sperre/miner), Førbekjempe/SPD, Anti-Log (bekjempe forsyninger).

o = Områder for manøver med terrengetyper: Åpent / Delvis åpent / Kanaliserende + lavvokst/skog/tettsted. En terrenggraf som vist i Figur 2.1 beskriver scenario og manøver ved et antall noder/områder N med forbindelser/akser k, m mellom dem, og der matrise-elementer r_{km} i en $N \times N$ matrise kan være vektorer som angir max-styrke i noder, vei-kapasitet (Rv, Kv, lende, umulig) og lengde i km, etc, for å bestemme tidsforbruk ved forflytning av komponenter mellom områder.

t = Tidsperiode indeks, varighet av periode er L (f eks 6 timer, 20 perioder gir 5 døgn).

2.2.3 Effektivitetskriterium/målfunksjon

Det struktureringsproblem som analysemodellen skal løse er formulert som følgende matematiske programmeringsproblem i kapittel 2.2.1.1 over:

$$\begin{aligned} & \text{Max}_X \sum_s p_s V_s(X) \\ V_s(X) &= \text{Max}_p \text{Min}_n \left(\sum_m P_m f_s(S_{m,X}, S_{n,Y}) \right) \end{aligned}$$

Det er mulig å benytte ulike effektivitetskriterier eller målfunksjoner $f_s(S_{m,X}, S_{n,Y})$ i forskjellige scenarier. Som nevnt benyttes to ulike typer funksjoner for hhv to-sidig (tradisjonell invasjonssituasjon) og fler-sidig (internasjonale operasjoner) scenario. For to-sidige scenarier er et uttrykk for sannsynlighet for å lykkes (Prob(egen vinn)) benyttet (se (32)(33) og Appendiks A), mens fler-sidige scenarier bruker en vektet lineær Styrkedifferanse mellom ulike sider som kriterium. Det er meget enkelt å endre disse funksjonene for å belyse virkningen av andre valg av effektivitetskriterier, noe som er en stor fordel ved denne type optimeringsmodeller.

Som grunnlag for effektivitetskriteriene er samlet styrke X_{ot}, Y_{ot} til enhver tid beregnet som en vektet kombinasjon av de komponenter som inngår i egen og motstanders struktur. De vektene w_i, w_j som er benyttet er basert på egenverdier fra lavere nivå modeller (standardverdier er som en første tilnærming satt til anslåtte kostnader for hver komponent). Samlet styrke inngår deretter i uttrykket for Prob(egen vinn) for to-sidige og i Styrkedifferanse for fler-sidige scenarier. Mer detaljert fås følgende målfunksjoner:

1. To-sidige scenarier: Effektivitetskriteriet for to-sidige scenarier $f_s(S_{m,X}, S_{n,Y})$ er en vektet

sum av Prob(egen vinn) i hvert område og hver periode = $\Pr_s(X, Y)_{m,n}$ definert ved:

$$X_{ot} = \sum_i w_i x_{iot} ; Y_{ot} = \sum_j w_j y_{jot}$$

$$R_{ot} = \frac{\sqrt{3} \cdot d \left(\sqrt{k_x / k_y} \cdot X_{ot} - Y_{ot} \right)}{\sqrt{\sqrt{k_x / k_y} \cdot (X_{ot} + Y_{ot})}} \quad (2.1)$$

$$f_s(S_{m,X}, S_{n,Y}) = \Pr_s(X, Y) \equiv \sum_{o,t} v_o \cdot \frac{1}{2} \left(1 + \operatorname{sgn}(R_{ot}) \sqrt{1 - e^{-\frac{-2R_{ot}^2}{\pi}}} \right)$$

der R_{ot} er et tilnærmet normalfordelt styrkeforhold i node o ved tid t , d en skaleringsfaktor som kalibrerer til et nominelt styrkenivå, v_o er prioritet av node o , og k_x/k_y det relative effektforhold mellom egne og motstanders styrker beregnet som forholdet mellom motstanders og egne tap (32)(33). I dette uttrykket er strategiene m,n også underforstått å inngå i hver styrkevariabel.

To varianter av dette uttrykket kan også enkelt anvendes. Den ene benytter kun styrkene i siste tidsperiode for å relatere kriteriet mer til begrepet 'end-state'. Dette kan enkelt tilpasses i ligning 2.1 ved å bruke styrkene X, Y i siste tidsperiode T i uttrykket for R_{ot} . Den andre varianten betrakter sannsynligheten for å vinne i et scenario som helhet ved å benytte start- og sluttstyrkene i uttrykket i ligning (2.1) og anvende Lanchesters 2. lov på disse. Dette gir følgende uttrykk for det samlede styrkeforhold R som deretter kan benyttes til å beregne sannsynligheten for å vinne i scenariet som helhet:

$$X_t = \sum_o X_{ot} ; Y_t = \sum_o Y_{ot}$$

$$R = \frac{\sqrt{3} \cdot c \left(\sqrt{(Y_1^2 - Y_T^2) / (X_1^2 - X_T^2)} \cdot X_1 - Y_1 \right)}{\sqrt{\sqrt{(Y_1^2 - Y_T^2) / (X_1^2 - X_T^2)} \cdot (X_1 + Y_1)}} \quad (2.2)$$

$$f_s(S_{m,X}, S_{n,Y}) = \Pr_s(X, Y) \equiv \frac{1}{2} \left(1 + \operatorname{sgn}(R) \sqrt{1 - e^{-\frac{-2R^2}{\pi}}} \right)$$

Med denne varianten kan eventuelt noder gis individuell prioritet ved også å vekte bidragene til totalstyrkene X_b, Y_t i hver node med deres prioriteter v_o (ikke vist i ligning 2.2).

Når sannsynligheten for å vinne for de ulike strategikombinasjonene $S_{m,X}, S_{n,Y}$ er beregnet, kan en blandet strategi P_m finnes som alternativt optimerer egen og motstanders spillverdi:

$$V_s(X) = \operatorname{Max}_P \operatorname{Min}_n \left(\sum_m P_m f_s(S_{m,X}, S_{n,Y}) \right)$$

Minimering for motstander er i praksis implementert som maksimering av negativ spillverdi. I fler-scenario sammenheng vil forventningsverdien over alle scenariene optimeres med gitt prioritet p_s (normert som sannsynlighet) for hvert scenario for å bestemme Maximin-løsning som beskrevet i algoritmen i kapittel 2.2.1.1:

$$\operatorname{Max}_X \sum_s p_s V_s(X)$$

2. Fler-sidige scenarier : Effektivitetskriteriet for fler-sidige scenarier er en vektet lineær styrkedifferanse mellom sider i hvert område og hver periode:

$$\begin{aligned} X_{ot} &= \sum_i w_i x_{iot} ; Y_{ot} = \sum_j w_j y_{jot} ; \dots \\ f_s(S_{m,X}, S_{n,Y}) &= \sum_{o,t} v_o (u_x X_{ot} + u_y Y_{ot} + \dots) / T \end{aligned} \quad (2.3)$$

der X_{ot}, Y_{ot}, \dots representerer vektet styrkenivå for de ulike koalisjoner av sider som kan opptre, T er varighet av scenariet, og u_x, u_y er relative prioriteter for de ulike sidene (standardverdi 1 for samme side og -1 for ulike sider, men det kan gis f eks størst og noe mindre positiv prioritet til hhv egne og allierte, og tilsvarende negative prioriteter til ulike motparter). Dette er en første enkel versjon som kan erstattes av andre kriterier etterhvert som mer kunnskap om slike fler-sidige internasjonale situasjoner opparbeides. Som en første tilnærming vil denne funksjonen likevel kunne representere relativ viktighet og betydning av ulike parter i en komplisert konflikt ved å variere prioritene u_x, u_y i ligning 2.3.

Som for to-sidige scenarier kan deretter styrkedifferansene for de ulike strategikombinasjonene $S_{m,X}, S_{n,Y}$ beregnes, og en blandet strategi P_m finnes som alternativt optimerer egen og motstanders spillverdi:

$$V_s(X) = \text{Max Min} \left(\sum P f_s(S_{m,X}, S_{n,Y}) \right)$$

I fler-scenario sammenheng vil som for to-sidige scenarier tilsvarende forventningsverdien over alle scenariene optimeres med gitt prioritet p_s (eventuelt normert som sannsynlighet) for hvert scenario for å bestemme Maximin-løsning som beskrevet i algoritmen i kapittel 2.2.1.1:

$$\text{Max}_X \sum_s p_s V_s(X)$$

For både to- og fler-sidige scenarier er modellen utviklet med én målfunksjon for hvert scenario. Det kan også være mulig å bruke fler-målsanalyse, men dette gir ytterligere kompleksitet med Pareto effektive løsninger som må finnes, og dette er ikke behandlet videre men kan være en mulig senere utvidelse.

2.2.4 Beskrankninger

For å beskrive styrkeutvikling og dimensjonering i forhold til en kostnadsramme, må et sett av beskrankninger etableres. Det vil være disse som i stor grad bestemmer hvor realistisk en analysemodell blir. I det følgende er beskrevet et sett av beskrankninger som i en første versjon kan inngå. Det er tatt hensyn til mulige avhengigheter mellom komponenter (for å gi en rimelig balansert struktur), til arv og til at det skal være mulig å teste egne strukturforslag i modellen.

K2 og oppklaring, samt begrensninger i egen og motstanders manøver er det mest krevende å få modellert på en akseptabel måte. I det følgende er de viktigste typer beskrankninger beskrevet slik de uttrykkes innen optimering.

For selve implementasjonen i AMPL med bruk av løsningsmetoden HOTDOG (37)(38) gjøres det bruk av definerte variable for likhetsbeskrankninger i størst mulig grad, mens de resterende beskrankningene gjøres om til en form som er tilpasset HOTDOG (37). Hver beskrankning er gjort om til en definert variabel $v(x)$ slik at beskrankninger av typen $g(x) \leq h(x)$ blir til uttrykket $g(x) - h(x)$, og $g(x) = h(x)$ blir til $|g(x) - h(x)|$. Dessuten skal det for hver beskrankning også fastsettes en toleransegrense $Feas_Tol$ og en såkalt straffevekt Pen som tilsammen skal sikre at brudd på beskrankningene gir ugunstig optimeringsresultat i form av straffeledd, samtidig som resultatet er mest mulig upåvirket så lenge beskrankningen er tilfredsstillt. Straffeleddene er en kombinasjon av en kvadratisk funksjon innenfor toleransegrensen og en lineær funksjon utenfor. Hvis uttrykket for beskrankningen er $v(x)$ blir straffeleddet:

$$L(v(x)) = \frac{Pen}{2 \cdot Tol} \cdot \min(Feas_Tol, \max(0, v(x)))^2 + Pen \cdot \max(0, v(x) - Feas_Tol)$$

En samlet objektfunksjon blir så optimert som består av selve målfunksjonen pluss summen av alle beskrankningenes straffeledd (37):

$$Obj_s(S_{m,x}, S_{n,y}) = f_s(S_{m,x}, S_{n,y}) + \sum_k L_k(v_k(x))$$

Avhengig av bidraget til resultatet fra de enkelte straffeledd relativt til målfunksjonen, kan det være behov for justering av straffevektene Pen_k og toleransegrensene $Feas_Tol_k$. Dette blir beskrevet nærmere for hver beskrankning. Detaljert implementert versjon i AMPL er vist i Appendiks A.

2.2.4.1 Scenario/Trussel

Trussel og scenarioegenskaper bestemmes av motstanders OOB ved tid 1 og av terrenggrafan. Beskrankningene for trusselen skal uttrykke at summen av hver motstander komponenttype over alle noder i terrenggrafan ved start må være mindre eller lik OOB trusselnivå for den samme komponenttypen. For egne styrker kan startstrukturen også spesifiseres med en arv.

$$\begin{aligned} \sum_o y_{joi} &\leq OOB y_j (= arv_j) \quad , j = 1, \dots, J \\ \sum_o x_{ioi} &\geq arv_i \quad , i = 1, \dots, I \end{aligned} \tag{2.4}$$

I beskrankning (2.4) er $OOB y_j$ det antall komponenter av type j som inngår i trusselen og J antall ulike komponenttyper som inngår. Dette kan også betraktes som en "arv" for motstanderen som gis inn som data til modellen. Tilsvarende kan arv_i også gis inn for egne styrker i hvert scenario slik at egen startstruktur x_{iI} alltid vil inneholde minst denne arven som ikke vil bli tatt med i totalkostnadene for strukturen. Budsjettet må også reflektere de reelle rammer utover arven.

Terrenggrafan bestemmes av antall noder, deres sett av naboer (forbindelsesakser), avstander mellom noder, terrengtype i noder og kapasitet av akser (se også Figur 2.1):

$$\begin{aligned}
 ar &= \{noder\}_o, o = 1, \dots, O \\
 ad_o &= \{naboer_o\}_l, l = 1, \dots, L \\
 r_{ol} &= [akseavstand]_{ol}, o \neq l = 1, \dots, O \\
 ak_{ol} &= [aksekapasitet]_{ol}, o = 1, \dots, O; l = 1, \dots, L \\
 nk_o &= [nodekapasitet]_o, o = 1, \dots, O \\
 trty_o &= [nodeterreng]_o, o = 1, \dots, O
 \end{aligned} \tag{2.5}$$

Terrengrafen ar (area) i (2.5) har et antall noder O som karakteriseres ved nabosettet ad til hver node (gir forbindelsesakser), avstandsforhold r og terrengtype $trty$. Noder har et maksimalt antall mulig deployerte komponenter nk_o og akser en aksekapasitet ak_{ol} uttrykt ved antall kjørefelt. Terrengtype i noder gir videre oppslag til fremrykningshastigheter på akser og også oppslag til effektivitet av ulike oppdragstyper i noder der engasjement mellom komponenter skjer, slik at resultat fra lavere nivå modeller kan tilordnes terrengegenskaper. Det er for enkelthets skyld kun mulig med engasjement i noder. Dersom akser også skal med, kan dette tilnærmes ved å definere et tilstrekkelig antall ekstra noder på aksene, men dette vil samtidig øke modell-dimensjonene.

2.2.4.2 Kostnad

Kostnadsdata for komponenter er som beskrevet over antatt å komme fra KOSTMOD evt også supplert med fredsdrift data. Det kan også være en mulighet å bruke en årlig basert modell mer "finansielt" orientert (17), der hver komponentkostnad er en årlig midlere levetidskostnad som består av en sum av årlig anskaffelseskost (= (investering + "midlife update")/levetid) + årlig personellkategorikost og materielldriftskost pr komponent + utlignede øvrige årlige driftskostnader på hver komponent (f eks proporsjonalt med dens anskaffelses- + personellkostnader. Evt arv kan f eks anses uten kost og med budsjett justert tilsvarende, eller alternativt kun ha "midlife update" kostnad/rest-levetid og driftskostnader). Uansett kostnadsmodell vil kostnadsbeskranking for strukturen (ved tid=1 initielt) være:

$$0 \leq \sum_i \left(\left(\sum_o x_{io1} \right) - arv_i \right) \cdot c_i \leq b_{struktur} \tag{2.6}$$

Rammekostnaden $b_{struktur}$ må justeres for det strukturproblem man skal analysere slik at beskrankningen reflekterer "riktig" del av den totale budsjettammen.

Som beskrevet foran for modellen med bruk av løsningsalgoritmen HOTDOG (37) gjøres beskranking 2.6 om til en definert variabel $Total_cost$ lik differansen mellom venstre- og høyre-side med et straffeledd som adderes til målfunksjonen og gir en total objektfunksjon som optimeres. Beskrankningsvariabelen må gis to parametre som er en toleranse $Feas_Tol$ og en såkalt straffevekt Pen (standardverdier for disse parameterne for $Total_Cost$ er satt til hhv 0.1 og 1000). Det kan være nødvendig å etterjustere disse parameterne avhengig av straffeleddets bidraget til total objektfunksjon fra $Total_Cost$ beskranking relativt til selve målfunksjonens verdi.

For å se strukturutviklingen i scenariet er det også hensiktsmessig å inkludere en variabel

$x_{it} = \sum_o x_{iot}$ som gir antall egne komponenter til enhver tid, der den beregnede x_{it} er den startstruktur som optimeres.

2.2.4.3 Operasjonskonsept / oppdrag

I modellen har hver komponenttype et sett av oppdragstyper ta_{ib} som er inndata sammen med effektiviteter for oppdragene som resultat av modeller på lavere nivå. Hver komponent fordeler en andel $x_{ta_{iobt}}, y_{ta_{joat}}$ av disponibel reststyrke til de ulike oppdragene i hver node til enhver tid. Oppdragsandelene for egne og motstanderen optimeres i modellen utfra effektivitetene og en foreslått engasjementsregel der de mest effektive engasjementene søkes, basert på relativ effektivitet $reff_{ijba}$ av komponentene i de ulike oppdrag. Denne relative effektivitet kan ytterligere påvirkes for spesielle oppdrag, f eks C2W, ved å gi en parameter $task_creator_{ibj}$ en stor verdi (i e 100). Derved justeres relativ andel for dette oppdraget mot motstander j tilsvarende opp ifølge den prioritet man ønsker for oppdraget. Dette gir mulighet for å styre oppdragsvalg etter et overordnet konsept, der oppdrag kan prioriteres utover de effektiviteter som lavere nivå modeller har gitt.

Oppdragsandelene for hver komponent kan ikke overstige kapasiteten, og en beskrankning sikrer at andelene summeres til 1:

$$\sum_b x_{ta_{iobt}} = 1 \quad ; \forall i, o, t \quad (2.7)$$

Beskrankning (2.7) er vist for egne komponenter, men en tilsvarende finnes også for motstander. Denne type beskrankning er implementert modell med HOTDOG i Appendix A også gjort om til en definert variabel $task_limit$ med tilhørende $Feas_Tol$ og Pen parametre som må gis som inndata (standardverdier hhv 0.03 og 2), og evt etterjusteres som for $Total_Cost$ i forrige avsnitt.

For komponenter i delmengden ac (luftrelatert støtte) er det også en beskrankning som sikrer at sum av andelene i hver node som kan utføre oppdrag mot ulike områder totalt ikke kan overstige 1 til enhver tid:

$$\sum_{o'} x_{ac_{ioo't}} \leq 1 \quad ; \forall i \in ac, o, t \quad (2.8)$$

Disse beskrankningene gjelder også for motstander og er implementert som en definert variabel $acfrac_limit$ tilsvarende som for $task_limit$ over med tilhørende $Feas_Tol$ og Pen parametre som må gis som inndata (standardverdier hhv 0.1 og 1), og evt etterjusteres.

2.2.4.4 K2 og informasjon

K2 og informasjon (ISR) skal betraktes som strukturkomponenter på like linje med andre komponenter. Effekten av K2 og informasjon må derfor representeres på en måte som kan påvirke målfunksjonen direkte eller indirekte. I tidligere analyser og i tilsvarende sammenhenger er effekten av K2 og informasjon søkt beskrevet som påvirkning dels på strategisettet og dels på effektiviteten av styrker (1)(10)(11)(30)(31).

Det er meget vanskelig (om ikke umulig?) å representere strategiendringer som følge av K2 direkte i modellen (11)(30), og dette er derfor kun tilnærmet på en indirekte måte ved en K2 kontrollspennfaktor $C2_cap$ (standardverdi 3) som bestemmer mulighet for manøver eller luftsorities avhengig av mengden K2 innenfor en K2 kontrollradius max_r . Ulik kvalitet på K2 kan uttrykkes ved disse to parametrene hva angår påvirkning på komponentenes manøver i terrenggrafene (11). Dette er gjort ved beskrankninger som begrenser totalt antall komponenter som kan flytte fra en node til andre noder avhengig av samlet K2 kontrollspenn summert over det nodesett R som ligger i avstand innenfor kontrollradius max_r fra noden:

$$\sum_{i,o'} x_{fr_{100't}} \cdot x_{10t} \leq C2_cap \cdot \sum_{o' \in R} x_{c2.o't} \quad ; \forall o,t \quad (2.9)$$

Disse beskrankningene gjelder også for motstander og er implementert som en definert variabel $C2move$ for bakkestyrker og $Aircraft_control$ (med et noe modifisert uttrykk) for luftstyrker i flybaser med tilhørende $Feas_Tol$ og Pen parametre som må gis som inndata (standardverdier hhv 0.2 og 0.1), og evt etterjusteres.

Endringer i styrkers effektivitet som følge av mengde og kvalitet av K2 og informasjon ISR (1)(11) er enklere å representere gjennom en tilleggsfaktor $C2_eff$ som angir relativ effektivitetsendring sfa antall K2- og informasjonskomponenter (x_{C2} og x_{ISR}) i både grupperings- og mål-noder. Denne tilleggsfaktoren multipliseres direkte med de nominelle effektivitetskoeffisienter fra inndata som inngår i tapsberegningene, og er i modellen beskrevet vha tre parametre: max_eff , min_eff og $rate_eff$ for K2 og ISR (standardverdier hhv 1.3, 0.7 og 0.7). Disse parameterne angir hhv hvor mye effektiviteten med maksimal K2 og ISR relativt kan forbedres, hva effektiviteten med minimal K2 og ISR vil reduseres til i forhold til de nominelle effektiviteter fra inndata. Raten uttrykker hvor mye effektiviteten vil endres for hver ekstra K2 og ISR komponent. Disse parametrene kan bestemmes av modeller på lavere nivå (1)(31), og de angitte standardverdiene (basert på eksempel i (1)) tilsvarer hhv ca økt styrkeforhold med 10-20 % for maksimal K2 og ISR, og tilsvarende reduksjon ved minimal K2 og ISR. Mer detaljert er tilleggsfaktoren for K2 og ISR uttrykt slik:

$$C2_eff = max_eff - (max_eff - min_eff) \cdot e^{-rate_eff \cdot \sqrt{x_{C2} \cdot x_{ISR}}} \quad (2.10)$$

Denne tilleggsfaktoren er direkte multiplisert inn og er i implementeringen ikke definert som egen variabel. Det er nærmere beskrevet hvordan den inngår i tapsberegningene i underkapittel 2.2.4.5 nedenfor.

Ved å gi C2W oppdrag og dermed redusere antall K2 og ISR komponenter, kan man få uttrykt avhengigheter i målfunksjonen indirekte gjennom både påvirkning på manøver og på effektivitet av oppdrag. Derved vil også ledelse og informasjon kunne inngå som strukturkomponenter, ihvertfall i en første tilnærming (1)(11)(30)(31).

Disse to hovedpåvirkningene av K2 og ISR som er beskrevet er de som foreløpig inngår i modellen. Andre faktorer kan legges til dersom sammenhenger og datagrunnlag er kjente.

2.2.4.5 Dynamikk og manøver

Den dynamiske utvikling i scenariet over tid er nedenfor uttrykt vha differensligninger som er balanseligninger i hvert tidspunkt for hvert enkelt område. Balanseligningene gir reststyrke i et område etter styrketilførsel inn, tap og styrkeoverføring til andre områder. Terrengegenskaper er beskrevet som begrensninger i samlet styrkemengde i områder og i kapasitet og forflytningshastighet på akser. Derved vil manøver begrenses utfra data fra andre modeller som kan tilnærme reell manøverhastighet fra lavere nivå analyser. Anti-mobilitet oppdrag kan også modelleres som begrensninger f eks avhengig av egen ingeniørstøtte tilsvarende K2 foran, men dette er foreløpig ikke inkludert i modellen.

Det kan også tenkes å benytte kall til andre simuleringsmodeller direkte for ytterligere realisme i dynamikk og manøver, men kompleksitet og krav til regneressurser vil da øke minst tilsvarende. En tilpasning og integrering av data fra lavere nivå modeller er derfor benyttet for å balansere disse to faktorene og likevel kunne representere scenarier på en tilfredsstillende måte.

De dynamiske balanseligningene i hvert område er laget utfra en netto-betraktning som beskrevet over, der :

$$\text{styrke}_i = \text{styrke}_{i-1} - \text{tap}_i + \text{tilført}_i - \text{avgitt}_i \quad (2.11)$$

Slik disse balanseligningene er formulert vil de være såkalte likhetsbeskrankninger som må tilfredsstilles under optimeringen. Dette gir et økt antall beskrankninger, og ved å omforme disse ligningene noe kan de istedet uttrykkes som såkalte *definerte* variable (en type makroer) som kun holdes som kalkulasjonsrelasjoner utenfor selve optimeringen. Dette er mulig ved å innføre en egen kalkulasjonsvariabel *calc(key)* der hver *key* kan defineres for de enkelte ledd i balanseligningen over (*komponenttype* og *område* indeks inngår også, men er ikke vist):

$$\begin{aligned} \text{calc}(\text{styrke}_i) &= x_i = \text{start}(\text{side}) && \text{(tilsvarende for } y_i) \\ \text{calc}(\text{styrke}_i) &= \text{Max}[0, \text{calc}(\text{disp}_i) - \text{calc}(\text{avgitt}_i)] \\ \text{calc}(\text{disp}_i) &= \text{calc}(\text{styrke}_{i-1}) + \text{calc}(\text{tilført}_i) - \text{calc}(\text{tap}_i) * L && (2.12) \\ \text{calc}(\text{avgitt}_i) &= \text{calc}(\text{disp}_i) * \text{Min}[\text{xfr}_i, \text{Max}[0, \text{calc}(\text{ratio}_i) * K1 - K2]] \\ \text{calc}(\text{tilført}_i) &= \sum \text{calc}(\text{avgitt}_{i-k}), \quad k = 1, \dots, (i-2) && \text{(tidligere avgitte)} \\ \text{calc}(\text{tap}_i) &= \text{motst.}\{ \text{eff} * \text{ned} * (\text{calc}(\text{styrke}_{i-1}) + \text{calc}(\text{tilført}_{i-1}) * \text{yta}_{i-1}) * C2 - \text{eff}_{i-1} \} \\ \text{calc}(\text{ratio}_i) &= \text{calc}(\text{disp}_{i-1}) / \text{motst.}\{ \text{calc}(\text{disp}_{i-1}) \} && \text{(styrkeforhold)} \end{aligned}$$

Dette sett av ligninger for definerte variable inneholder referanse til kun tre typer frie beslutningsvariable som skal optimeres (kfr kapittel 2.2.2): startstrukturen *start* (=x₁/y₁), andelen som manøvrerer ut av hver node *xfr/yfr* og andelen fordelt til de ulike oppdragstyper *x_{ta}/y_{ta}*. Resten av de definerte variable uttrykkes som vist iterativt ved kombinasjon av tidligere definerte variable, og kan således kalkuleres uten å inngå som likhetsbeskrankninger.

Ligningen for avgitte styrker inneholder en definert variabel *ratio* som bestemmer i hvilken grad en sides styrker kan manøvrere avhengig av graden av kontroll i et område. Derved vil den strategi en side har valgt i forrige optimeringsiterasjon kunne bli påvirket av valg av strategi for den side som i nåværende iterasjon blir optimert. Dette gjør at spillmatrisen vil kunne reflektere mer realistiske forløp utfra den dynamiske utvikling i scenariene. Styrkeforholdet *ratio*

bestemmer gjennom parametrene $K1$ og $K2$ hvor stor styrkeandel som kan tillates å manøvrere fra et område inn i et naboområde. Standardverdier for parametrene (hhv 0.6 og 0.3 ut og 1.5 og 0.5 inn) er anslått slik at den side som optimeres må ha et styrkeforhold på minst $\frac{1}{2}$ for å kunne manøvrere ut og også et styrkeforhold på minst $\frac{1}{3}$ i naboområdet for manøvrer dit. De omvendte styrkeforhold vil gjelde for påvirkning av motstanderens manøvrer fra forrige optimeringsiterasjon. Ved likt styrkeforhold i områdene vil man kunne flytte hhv $\frac{1}{3}$ av styrken ut og inntil hele styrken inn. Disse standardverdiene kan enkelt endres for å belyse konsekvens av andre vurderinger.

Tap

Ligningen for tap tar hensyn til motstanderstyrkenes effektivitet eff inklusive relativ allokering til ulike mål, evt nedholdende virkning på styrkene ned , samt virkningen av $K2$ og ISR $C2$ -eff. Effektivitet eff_{ji} med relativ allokering av motstander type j til ulike måltyper i er igjen en funksjon av antall mulige mål av type i , $styrke_i$, og av den relative effektivitet $reff_{ji}$ mot hver av disse:

$$\begin{aligned} eff_{ji} &= [calc(styrke_{i,t-1}) * reff_{ji} / \sum_i (calc(styrke_{i,t-1}) * reff_{ji})] \\ reff_{ji} &= [P_{ji} / \sum_j (P_{ji} / \sum_i P_{ji})] \end{aligned} \quad (2.13)$$

der P_{ji} er produktet av de grunnleggende effektivitets- og engasjementsparametrene $p_{ji}k_{ji}$ fra kapittel 2.2.2. For eksempel vil en situasjon med to motstandertyper j og to måltyper i med alle $k_{ji}=1$ og der $p_{11}=.8$, $p_{12}=.2$, $p_{21}=.4$ og $p_{22}=.6$, gi følgende relative effektiviteter $reff_{11}=2/3$, $reff_{12}=1/4$, $reff_{21}=1/3$ og $reff_{22}=3/4$. Med likt antall mål av begge typer i vil de relative allokeringer bli $eff_{11}=8/11$, $eff_{12}=3/11$, $eff_{21}=4/13$ og $eff_{22}=9/13$, dvs at type $j=1$ prioriterer måltype 1, mens type $j=2$ tilsvarende prioriterer måltype 2. Med dobbelt så stort antall av måltype 2 som av 1, vil motstandertype $j=1$ nær doble sin relative allokering mot mål 2 ($eff_{12}=3/7$) og type $j=2$ ytterligere øke sin relative allokering mot mål 2 ($eff_{22}=9/11$). Dette gir en rimelig dynamisk allokering avhengig av relativ ytelse mot ulike måltypesammensetninger uten at det er nødvendig å øke modellkompleksiteten ved f eks å inkludere beslutningsvariable også for allokeringene. Andre allokeringer enn ligning (2.13) er enkelt å endre i modellen dersom det er ønskelig.

For motstanders manøvrerstyrker kan deres tapspåførende effektivitet også påvirkes av hvor mye de selv blir nedholdt pga beskytning fra fly og artilleri pr tidsenhet gjennom faktoren ned . Denne faktoren bestemmes av tre parametre $norm$, $high$ og low . Som et mål på beskytningen brukes det prosentvise tap til fly og artilleri innenfor rekkevidde, slik at $norm$ skal tilsvare den beskytning som de grunnleggende effektivitetsparametrene p_{ji} representerer. Dersom beskytningen er større vil $high$ tilsvare en reduksjon i effektivitet som skyldes maksimal nedholdt effekt, mens omvendt mindre beskytning vil medføre økt effektivitet med faktoren low . Dermed kan lavere nivå modeller gi grunnlag for å tilpasse modellen til ulik grad av støtte (se også kapittel 2.2.4.6 nedenfor). Ligningen for faktoren ned er:

$$ned = high - (high - low) \cdot \left(\frac{1 - high}{low - high} \right)^{tap, I(L*norm*styrke_i)} \quad (2.14)$$

For eksempel vil verdiene 0.01 , 0.2 og 1.5 for hhv *norm*, *high* og *low* medføre at effektiviteten vil øke med en faktor 1.5 uten beskytning (og nedholdende effekt), og bli redusert med en faktor ca $2/3$ ved dobbelt beskytningseffekt (dvs $2 \cdot \text{norm} = 2\%$ tap pr tidsenhet). Maksimal reduksjonsfaktor for effektiviteten vil være 0.2 . Standardverdier er hhv 0.01 , 1.0 og 1.0 som ikke vil gi noen nedholdende effekt.

Som beskrevet i kapittel 2.2.4.4 foran er det også innført en faktor *C2-eff* (se ligning 2.10) i tapsberegningene som gir økt effektivitet til motstanderen avhengig av hans K2- og ISR-styrker i området. Dette påvirker direkte det dynamiske forløpet og gir en tilsvarende virkning på optimeringen slik at K2 og ISR vil kunne bli en del av strukturen og gi anslag på nødvendig mengde også av disse typer komponenter. Det er kun innledende betraktninger som er tatt med hva gjelder effekten av K2 og ISR, men ytterligere analyser og modeller kan bidra til faktorer som tilsvarende evt kan bygges inn (1)(10)(11)(23)(30)(31).

Tap av egne fly til motstanders fly og luftvern (variabel *calc('lossaac')*) er bestemt av de andeler egne fly $xac_{ioo't}$ som er på oppdrag fra en base-node o til andre noder o' til enhver tid. Tap av egne fly på base o til motstanders fly fordeles proporsjonalt med mengden egne fly på oppdrag fra basen og med totalt antall motstanderfly som er på oppdrag og innen rekkevidde, slik at samlet tap til motstanders fly fordeles relativt mellom flyene på egne baser. Tap av egne fly på base o til motstanders luftvern i node o' fordeles proporsjonalt med mengden fly på oppdrag fra base o til node o' og med totalt antall luftvernkomponeanter i node o' på tilsvarende måte som for tap til motstanders fly. Det er også mulig å tape fly på base o til motstanders manøverkomponeanter som befinner seg i samme base-node o ved å gi effektiviteter og engasjementsrater fra disse mot fly, men dette er ikke innført foreløpig (mangler datagrunnlag).

Manøver

Overføring av styrker fra et område o til et naboområde o' (settet $ad(o)$) styres av en rate bestemt av maksimal forflytningshastighet $v_{io'}$ (utfra akseterreng $terr_{o'}$) og akseavstand $r_{oo'}$ mellom o' og o . Det er antatt at forflytningstiden T_f alltid vil være minst en tidsperiode og ellers også være bestemt av styrkens kolonnелengde col_i slik:

$$T_f = \text{Max}[1, (r_{oo'} + col_i) / (v_{io'} \cdot L)]$$

Derved vil mengden tilførte styrker inn i et område i ligning (2.12) bestemmes av den tid tidligere avgitte styrker $xfr_{ioo't} \cdot x_{io't}$ bruker på forflytningen til dette området både gjennom styrkenes forflytningshastighet og kolonnелengde. Forflytning kan evt også gjøres avhengig av anti-mobilitetsoppdrag gjennom begrensninger tilsvarende som for K2 styrt av tilgjengelig mengde ingeniørstøtte, men dette er foreløpig ikke med i modellen..

Deployering i områder og forflytningskapasitet er også begrenset av antall styrker som totalt kan være i en node og på akser til enhver tid :

$$\begin{aligned} \sum_i x_{io't} + \sum_j y_{jot} &\leq N_o, \forall o, i \vee j \in \text{man}, t \\ \sum_i xfr_{ioo't} \cdot x_{io't} \cdot col_i + \sum_j y_{joo't} \cdot y_{jot} \cdot col_j &\leq r_{oo'} \cdot N_{oo'}, \forall o, o' \in ad(o), t \end{aligned} \quad (2.15)$$

der N_o er max-antall komponenter som kan være i node o samtidig og N_{oo} er kapasitet i antall felt på akse oo' .

Det kan også settes krav til tapsgrenser for egne styrker i områder for å angi avbruddskriterier :

$$\sum_i x_{iot} \geq (1 - \pi_x) \cdot \sum_i x_{io1}, \forall o, t$$

$$\sum_j y_{jot} \geq (1 - \pi_y) \cdot \sum_j y_{jo1}, \forall o, t$$

der π_x er tapsgrense som fraksjon av initielle styrker (kan evt deles inn i tidsfaser). Det kan også være mulig å se på kulminerings-forhold der styrkeutvikling over tid kan benyttes i istedet (9), evt også kreve et minimum styrkeforhold for offensive oppdrag utfra f eks Prob(egen vinn) som skissert i Appendix A. Disse tapsbegrensningene er foreløpig ikke med i modellen.

2.2.4.6 Taktisk støtte

Taktisk støtte består i oppdrag utført av artilleri, fly, UCAV/UAV og heli styrker innen rekkevidde $max_r(i)$ for mulige mål i scenariografen. Artilleri kan også flytte tilsvarende som manøverstyrker mellom områder, mens fly, UCAV/UAV og heli antas å operere fra gitte baser $base(i)$ som må defineres som inndata blant områdene i scenariografen.

For fly, UCAV/UAV og heli er den dynamiske utvikling i ligning (2.12) i tillegg til antallet komponenter også bestemt av maksimal total sortie-rate $f_{rt}(i)$ i hver periode gjennom et sett av beskrankninger:

$$\sum_{o,o'} (L_i - 2r_{oo'} / v_{io'}) \cdot x_{ac_{ioo't}} \cdot x_{iot} \sum_{j,a,b} k_{jjoba}^q / n_{jab} \leq L_i \cdot f_{rt}_i, \forall i \in ac, t \quad (2.16)$$

som uttrykker at tilgjengelig flytid til alle mål o' for alle oppdrag med gitt sortie-rate summert over alle flybaser o skal gi et totalt antall sorties i en periode som ikke må overstige et samlet maksimalt antall $L_i \cdot f_{rt}_i$. Disse beskrankningene gjelder også for motstander og er implementert som en definert variabel $Fire_ac$ med tilhørende $Feas_Tol$ og Pen parametre som må gis som inndata (standardverdier hhv 10 og 0.005), og evt etterjusteres.

For artilleri er det for å forenkle modellen ikke lagt inn slike totale beskrankninger i samlet antall engasjementer i en periode, fordi det også gjennom ligning (2.12) er en automatisk begrensning i mengden engasjementer for hver komponent bestemt av engasjementsparameteren k_{jjoba} . Det er mulig å legge inn tilsvarende beskrankninger som for fly dersom det er nødvendig.

Av samme grunn er det heller ikke lagt inn separate beskrankninger for bruk av f eks 'smart' ammunisjon, som derfor forutsettes styrt i hovedsak gjennom tilpassede verdier av engasjementsparameteren k_{jjoba} ved ulike oppdragstyper. Kostnad for en komponent vil derfor også inkludere en passende mengde ammunisjon fordelt til hver komponent. Det vil være mulig å beregne samlet mengde ammunisjon brukt i et scenario utfra de variable og parametre som inngår i modellen dersom det er ønskelig.

2.2.4.7 Logistikk

Logistikk kan tas med i en forenklet form, men er ikke direkte inkludert i modellen til nå. Som beskrevet over vil logistikk for artilleri være mest aktuelt, og dette er automatisk styrt indirekte av tilpassede verdier av engasjementsparameteren k_{ijoba} for ulike oppdragstyper. Det kan imidlertid være mulig å inkludere logistikk-type komponenter direkte og la dynamikklikningene (2.12) påvirkes også av potensialet for forsyninger til de enkelte områder. Derved kan begrensninger i disse gi påfølgende begrensninger i engasjementsrater fra støtte-komponenter. Dette innebærer at forsyninger inngår i logistikkavdelinger og dermed kan mistes, noe som videre kan innvirke på engasjementsratene f eks tilsvarende som for K2-komponenter i ligning (2.10) slik:

$$\log_eng = \max_eng - (\max_eng - \min_eng) \cdot e^{-rate_eng \cdot \sqrt{x_{log}}} \quad (2.17)$$

der \log_eng er en logistikkfaktor som kan multipliseres med engasjementsratene k_{ijoba} og som vil variere mellom \min_eng og \max_eng med rate $rate_eng$ avhengig av antall logistikk-type komponenter X_{log} innen rekkevidde. Dette vil gi en påvirkning på engasjementsraten pga logistikk tilsvarende påvirkningen på effektivitet pga K2/ISR-komponenter som vist i kapittel 2.2.4.4. Dette burde gi tilstrekkelig avhengighet til å kunne inkludere høynivå logistikk i modellen med rimelige krav til data fra lavere nivå modeller, dersom dette ønskes lagt inn.

Logistikk for manøvrering (POL, etc) er også mulig å inkludere dersom dette antas nødvendig ved å inkludere ekstra manøver-beskrankninger tilsvarende de som er vist for K2-komponenter i ligning (2.9), men nå med antall logistikk-type komponenter X_{log} innen rekkevidde som kontrollvariabel og med en logistisk kontrollparameter $Log_eff.$

2.2.4.8 Strukturavhengigheter / arv

For ikke å få for "grådig" struktur (12)(15)(18)(19) og også utfra flertrussel synspunkt, vil det være aktuelt å "tvinge" inn avhengigheter i strukturen gjennom f eks arv.

Arv kan evt tilpasses med tilordninger direkte i inndata gjennom parameteren $arv(i)$ for det antall av komponenttype i som skal arves ved starttid = 1. Disse komponentene vil da ikke inngå i kostnadsbeskrankningen som vist i ligning (2.6) i kapittel 2.2.4.2, men vil ellers bidra til målfunksjonen på samme måte som de optimerte beslutningsvariable. Imidlertid må rammekostnaden $b_{struktur}$ justeres tilsvarende for det strukturproblem man skal analysere, slik at beskrankningen reflekterer "riktig" del av den totale budsjettammen.

Det kan også bygges inn evt strukturavhengigheter mellom komponenttyper, f eks mellom manøver og artilleristøtte, dersom dette er ønskelig utover det som er beskrevet for K2/ISR- og mulige logistikk-komponenter over.

2.2.5 Gitt struktur / konsekvensanalyse

Det kan også være aktuelt å teste et gitt strukturforslag for å se konsekvenser av ulike valg av parametre. Modellen blir da en "hvis-så" type konsekvensmodell der oppdrag og disponering,

strategien, bestemmes utfra den gitte startstrukturen gitt som arv. Valg av strategi kan evt også “tvinges” inn; da blir optimeringsaspektet borte og modellen blir en kalkulasjon/bokføring av strategiens oppnådde målfunksjonsverdi over tid.

Beskrivelse av en gitt startstruktur blir tilsvarende som for arv, mens en strategi kan beskrives vha tre ulike typer strategivalg – detaljert, begrenset og fritt. Detaljert strategivalg vil si å spesifisere alle variabelverdier til alle tider i et scenario, noe som er krevende på grunn av kravet til konsistens og mengden variable. Begrenset strategivalg er enklere og kan spesifiseres ved å angi delmengder av noder i scenariografen som styrkene kun tillates å benytte, mens fritt strategivalg overlater til modellen selv å optimere oppdragsvalg og manøver mellom noder som frie variable. Avhengig av strategibeskrivelse vil målfunksjon og disponering i scenariet bli beregnet som en “hvis-så” type modell, som evt kan sammenlignes med tilsvarende forløp fra andre modeller.

2.2.6 Kompleksitet

Det ses at ligningene foran er ulineære gjennom både Max/Min-funksjoner og produkter av variable. Dette gir økt kompleksitet ved de løsningsverktøy som søker optimalverdier, fordi disse ofte bygger på spesialiserte forutsetninger bl a ved derivasjon. En egen løsningsalgoritme HOTDOG/MICIO er derfor også utviklet tilpasset modellen mest mulig som også har vist seg generelt anvendbar på ulineære beskrankede problemer generelt (kfr Appendiks A.3).

Et overslag over kompleksiteten for ett scenario kan gjøres ved å f eks anta totalt 20 komponenttyper (10 for hver side), 5 dagers scenario, 3 oppdrag pr komponenttype, 3 strategier pr side, 8 noder (med gjennomsnitt 3 nabonoder) og 6 times periode-lengde. Da blir antall frie variable totalt i hver iterasjon (vist for egen side x med motstanders variable fiksert) som følger :

```

start:    240
xta:     14400
xfr:     14400
xac:     1920
px:       3
annet:    2

```

```

-----
sum:    30965
-----

```

To faktorer bestemmer hvor store problem som kan løses med modellen. Den ene faktoren gjelder hukommelsesplass som blir begrensende for det overslaget som er beskrevet over, idet det vil kreve nær 2 GB hukommelse (virtuell hukommelse bør ikke brukes pga redusert CPU tilgjengelighet). Det er teknisk fullt mulig å anskaffe utstyr som kan løse et slikt scenario, men med nåværende datautstyr med 1 GB hukommelse vil man enten måtte redusere tiden til ca 3 dager eller redusere antall strategier til 2 på hver side.

Det er heller ikke gitt at et ulineært problem av en slik størrelse kan løses til optimalitet med standard type verktøy, og som nevnt er derfor et eget løsningsverktøy utviklet (se Appendiks

A.3) for å finne det beste (d v s det globale) optimum utfra et breddesøk. Selve modellformuleringen lar seg uttrykke omtrent som notasjonen i ligningene foran vha modellverktøyet AMPL (38) som tillater at mengder og indekser kan beskrives kompakt og fleksibelt. En detaljert brukerveiledning for modellen er beskrevet i (37) der inndata og kjøreoppsett er vist med et ugradert eksempel.

Det er viktig å understreke at selv om en optimalløsning beregnes, er det informasjon fra styrke-disponering og tidsutvikling, samt den "pris"- eller sensitivitets-informasjon som følger en slik løsning som er mest interessant. Bindende beskrankninger kan gi forståelse av dynamikk og "flaskehals", slik at verdi av endringer kan bedømmes og forstås.

For multi-scenario optimering, vil problemet øke på (minst) to måter. For det første må målfunksjon(er), effektivitetsdata og trussel beskrives for de nye scenariene, og dessuten vil dimensjonene øke tilsvarende, dersom problemet ikke kan dekomponeres. Dette gjør en slik global løsningsmetodikk svært krevende, men ikke umulig. En videre mulig forenkling og tilpassing av modellformuleringen foran er ønskelig som en aktivitet mot en versjon som er balansert i både omfang og detaljgrad.

3 DATAGRUNNLAG OG RESULTATBEHANDLING

I analysemodellen MSO².1 inngår et sett av data eller koeffisienter som forutsettes bestemt av andre del-modeller. Duellmodeller og separate effektivitetsberegninger kan gi slike grunnlagsdata basert på en egnet metode for aggregering/disaggregering som beskrevet i det følgende.

Et eksempel med resultater for et scenario er vist i Appendiks A.4 som utgangspunkt for resultatbeskrivelse og databehandling av informasjon fra analysemodellens optimalløsning.

3.1 Kostnadsdata

For kostnader er forutsatt at KOSTMOD kan gi nødvendige data. For de komponenter som inngår, må alle materiell- og personellkostnader for ulike kategorier som inngår summeres, slik at antallet komponenter av hver type gir korrekt kostnadsbidrag til samlet struktur (referert til f eks en KOSTMOD-kjøring (40)). Dessuten må ammunisjonskostnader ihvertfall for moderne, smart og dyr ammunisjon, også bestemmes utfra et midlere forbruk pr engasjement pr komponent, slik at dette kan inngå i f eks kostnader for artilleri- og luftkomponenter.

Det er ingen større problem forbundet med å fremskaffe slike data, selv om fordeling av felles driftsutgifter utover materiellrelatert drift kan være komplisert (dersom slike kostnader også ønskes fordelt til strukturens komponentkostnader). Det som i første omgang er enklest å foreta er en strukturberegning der kun de materiellrelaterte kostnader er inkludert, og der budsjettammer er justert i h h t til dette. En alternativ mer finansielt relatert kostnadsmodell er også benyttet (17) der en årlig levetidskost som tar hensyn til forventet levetid og en antatt MLU-kostnad er beregnet. Dette gir en modell der en struktur forventes vedlikeholdt med årlige investeringer innenfor årlig budsjett.

Som nevnt i kapittel 2.2.1.2 er det også utviklet en modell som søker å innfase en beregnet struktur fra analysemodellen over tid under hensyn til bl a teknologisk fordyrelse av materiell fra KOSTMOD. Denne MIKS-modellen er beskrevet i mer detalj i Appendiks A.5 og kan benytte ressursdata fra KOSTMOD (40) som underlag for en optimal innfasing av komponentandeler over tid der totalkostnad inkludert teknologisk fordyrelse inngår. Det er også enkelt å se evt konsekvenser av forsinket innfasing av valgt(e) komponent(er), slik at følsomhet av ulike innfasingshensyn også kan studeres.

3.2 Effektivitetsdata

I de effektivitetsdata som er angitt i ligningene foran er antatt at de kan representere en aggregering/disaggregering tilpasset det scenario som ønskes studert. Det er produktet av effektivitet pr engasjement pr komponent og engasjementsraten som bestemmer samlet effekt av komponentene i en tidsperiode i analysemodellen. Eksempeldata fra tilsvarende nivå modeller som MUNGO (11) kan danne et test-grunnlag for komponenters effektivitetsdata med supplering fra andre lavere nivå modeller som DYNACOM, TALUS oa.

Vekting av hver komponenttype til et samlet styrkemål i analysemodellen $MSO^2.1$ kan f eks baseres på egenverdi-betraktninger fra detaljsituasjoner i lavere nivå modeller. Dette gir et aggregert styrkemål som kan tilpasses en tidsutvikling fra disse modellene der vektet tapsprosent ved avbrudd skal være like. En aggregert effektivitetsparameter kan så beregnes vha f eks Lanchester-ligninger som vil gi samme tidsforløp for komponentstyrkens styrkemål som detaljsituasjonen på lavere nivå. Slike beregninger av aggregerte effektivitetsparametre er beskrevet i Appendiks A.4.1 for de data som er benyttet i eksemplet der.

En egen aktivitet vil måtte korrigere/tilpasse dataene løpende til resultater fra duellmodeller og andre delmodeller når slike foreligger. Dette bør kunne utvikles til å bli en nær automatisk prosess som kan gjennomføres ved egne verktøy. En forenklet metode for slik datatilpasning er laget i Mathematica og er beskrevet som del av sannsynlighetsmodellen for målfunksjonen i Appendiks A.1. Som en ytterligere direkte kobling til resultater fra delmodeller kan datagrunnlaget for analysemodellen også refereres ved databasekall via egne funksjoner for dette i AMPL, slik at data automatisk kan hentes inn f eks via skjemadefinisjoner og/eller SQL-spørring i databaser. Dette er vist som eksempel for innfasingsmodellen MIKS i Appendiks A.5.

Det er også viktig å påpeke betydningen av skjønnsmessige anslag som erfarne offiserer og analytikere kan fremskaffe for å fylle "huller" og validere del-modell data. En slik kontrollprosess vil være viktig for troverdigheten av de strukturresultater som beregnes

3.3 Resultatbehandling

Som beskrevet over vil det i tillegg til den optimalløsning som beregnes, være informasjon fra styrke-disponering og tidsutvikling, samt den "pris"- eller sensitivitets-informasjon som følger en slik løsning som totalt sett kan være mest interessant.

Resultatene som løsningsalgoritmen kommer frem til, skrives ut på en rekke filer. Tallmengden er så stor at det er vanskelig å sette seg inn i løsningene ved bare å se på verdiene til de enkelte variablene. Derfor beregnes det i tillegg summer og gjennomsnitt som skal gjøre det mulig å få et overblikk på kortere tid. Hvis man ønsker å få beregnet andre uttrykk enn de som inngår, kan man legge dem inn i kjørefilen eller modellfilen (se Appendiks A.2 for detaljer). De filene som skrives ut er kort beskrevet i det følgende med utdrag fra eksemplet vist i Appendiks A.4.3.

I tillegg til summer er det også utviklet Excel regneark med makroer som gir grafisk fremstilling av oversiktsdata med diagram fra eksemplet vist i Appendiks A.4.4.

3.3.1 Resultatfiler

Filen **zSummary.tmp** inneholder objektivverdien (*Obj*) og det egentlige målet på resultatet ($f=optimum$) for hver kjøring. Disse to verdiene bør være ganske like for at resultatet skal være gyldig. I tillegg inneholder denne filen verdiene på variabelen *start*, som angir antallet som er blitt kjøpt inn av hver komponent med tilhørende kostnader, og som således uttrykker den struktur som analysemodellen beregner (37).

Det er også vist spill-matrisen for de strategier som er beregnet i modellen med den blandede strategivektor (*prb*) som løsningen uttrykker. Det er en slik utskrift for hver iterasjon (*LOOP*) som er spesifisert i kjørefilen (*.run*). Den *LOOP* som har maksimal f velges som estimert optimal struktur (1).

Eksempel på utskrift av sammendragsfilen (utdrag fra *LOOP 10-12*) er vist i Appendiks A.4.3.

I tillegg lages det en rekke filer for hver kjøring. Disse skilles fra hverandre ved at filnavnene inneholder en tallkode "***-**", der det første tallet angir iterasjonen (*LOOP*), mens det andre angir kjøringen (*side*) innenfor denne iterasjonen.

zStructures-*.tmp** gir informasjon om strukturinnhold og kostnader i hver iterasjon (*LOOP*) og supplerer informasjonen i **ZSummary.tmp**.

zVar-*.tmp** gir informasjon om detaljerte variabelverdier og deres gradienter (reduisert kost) for bl a gjenværende komponentstyrker. Gir dynamisk utvikling av de enkelte komponenter i strukturen over tid.

zCon-*.tmp** gir informasjon om beskrankningene og inneholder slakkene, nedre og øvre grenser og dualverdiene for alle beskrankningene.

zExp-*.tmp** inneholder noen beregnede uttrykk som skal gi informasjon om blant annet utviklingen i striden med bl a oppdragsfordeling, tap til komponenttyper og gjenværende vektet totalstyrke sfa tid. Dette brukes i Excel regneark makro for grafisk sammendragsfremstilling.

zSol-*.tmp** kan brukes til å laste inn en tidligere løsning i AMPL som start på ny *LOOP*.

zHot-*-*.tmp** inneholder detaljert informasjon fra løsningsalgoritmen HOTDOG/MICIO med objektivverdier for hver repetisjon i søket etter beste lokalpunkt for hver iterasjon.

3.3.2 Excel resultatbehandling

For å lette oversikten over resultatene er det utviklet to Excel regneark for resultatbehandling:

1. **Sammendrag.xls** gir grafisk oversikt over vektet totalstyrke og tap, og fås ved å merke dette resultatområdet i filen **zExp**-*-*.tmp**, kopiere "Tekst til kolonne" inn i ark "Styrke", oppdatere Pivottabell og gå til "Diagram". Resulterende diameksempel er vist i Appendiks A.4.4 merket 'Sammendrags-ark'
2. **Komponentoversikt.xls** gir grafisk oversikt over de enkelte komponentstyrker, og fås ved å merke '*disp*'-resultatområdet fra filen **zVar**-*-*.tmp**, kopiere "Tekst til kolonne" inn i ark "Styrke", oppdatere Pivottabell og gå til "Diagram". Resulterende diameksempel er vist i Appendiks A.4.4 merket 'Komponent-ark'.

Begge diagrammene viser hvordan styrkenivå utvikler seg over tid gjennom de enkelte noder. Ytterligere diagram kan enkelt utvikles på tilsvarende måte for å dekke eventuelle andre behov for oversikter.

Optimalløsningen x_{iobt}, y_{jopt} er på en form som en flerdimensjonal tabell på en tekstfil **zVar**-*-*.tmp**, og vil derfor også kunne viderebehandles som input for en 'plug-in' funksjon direkte i STRIDSLOGG for å presentere løsningen dynamisk sfa tid, område (med assosiert UTM-koordinat), oppdragstype og komponenttype. Antallet av en komponenttype kan aggregeres i enheter av f eks 3 for å tilnærme en vanlig hierarkisk organisatorisk oppbygging i presentasjonene ved et enkelt filter. Dette vil gi en dokumentasjon av analysemodellens resultater som tilsvarende den form som ellers benyttes ved bruk av STRIDSLOGG (39) og vil være en del av videre arbeid med analysemodellen MSO².1.

Det kan også være mulig å generere en automatisk skriftlig rapportdokumentasjon for aggregerte enheters dynamiske utvikling utfra en enkel betegnelse (ordliste) for oppdragstyper og områdenavn.

Likevel er målfunksjonsbidrag og marginal-ytelse av komponentene fra sensitivitetsinformasjonen resultatbidrag som er sentrale i å karakterisere strukturløsningene for de scenarier som beregnes. Dette kan bygge opp kunnskap om samlet bidrag fra ulike komponenttyper og deres egenskaper.

4 KONKLUSJON

Det er gitt en beskrivelse av en generell analysemodell/metodikk for styrkestrukturering basert på kost/effektivitet som er utviklet i prosjekt 734/161 "Operativt konsept for landstrid".

Analysemodellen som er beskrevet er i prosjektet benyttet som hjelpemiddel for å utprøve idéer og forslag i praksis basert på de scenarier og oppgaver som prosjektet har behandlet. Modellen er således også et resultat i seg selv og inngår som en del av det modellhierarkiet som prosjektet har utviklet.

Modellen tar bl a hensyn til forflytninger og oppdragsvariasjon i definerte scenarier, slik at også disse dimensjonene vil kunne variere. Dette gir mulighet for å ta hensyn også til operasjonskonsept for styrker i scenarier, men antall dimensjoner medfører samtidig økt kompleksitet i modellen.

Analysemodellen kan benyttes til følgende problemstillinger :

- styrkestrukturanalyse for et sett av scenarier (evt også inkludert internasjonale operasjoner) med gitt kostnadsramme (helst direkte kosteffektiv struktur gitt både effekt- og kostbeskrivende (aggregerte) funksjoner som del av analysemodellen)
- styrkedimensjonering i et gitt scenario (dvs kosteffektiv komponentstruktur, helst også direkte som del av kosteffektiv styrkestrukturanalyse over)
- evaluering av "beste" handlemåter i et scenario for en gitt styrkestruktur (basert på (kanskje flere) effektivitetsmål, også for manøverorienterte operasjoner)

Analysemodellen optimerer struktur for komponentstyrker med støttestrukturer i et gitt scenario og er tilpasset for også å optimere over f eks en sannsynlighetsfordeling av scenarier for å finne en samlet og mest mulig robust struktur (13)(14)(15)(21)(25)(26)(27). Med struktur menes her kun antallet og settet av komponenter i strukturen; implementering av en struktur i løpet av en planperiode utføres vha en annen modell, f eks basert på optimering av anskaffelse over planperioden. Det er også utviklet en enkel delmodell for optimering av slik innfasing av en optimal struktur under hensyn til kostnad og teknologisk fordyrelse av materiell.

Analysemodellen for strukturoptimering kalles MSO².1 (Modell for StrukturOptimering i Multi-Scenario Operasjoner, versjon 1) og innfasingsmodellen MIKS (Modell for Innfasing av Kosteffektiv Struktur).

Med ulineære kriterier og beskrankninger vil algoritmen for optimering i hver iterasjon søke etter globalt optimum. Det kan da generelt ikke garanteres at globalt optimum finnes, men det er laget løsningsalgoritmer som søker etter flere lokale optimum, og deretter bruker det beste av disse som estimat av globalt optimum. En slik løsningsalgoritme, kalt HOTDOG (Heuristisk Oppskrift Til å Determinere Optimum Globalt / Heuristic On The Determination of Optimum Globally), er utviklet som et tillegg til modellen og er kort beskrevet i Appendiks A (se også (37)). Denne algoritmen har vist seg rask og robust også for store ulineære problemer.

I analysemodellen MSO².1 inngår et sett av data eller koeffisienter som forutsettes bestemt av

andre del-modeller. Duellmodeller og separate effektivitetsberegninger kan gi slike grunnlagsdata basert på en egnet metode for aggregering/disaggregering som beskrevet i Appendiks A.1.

Et eksempel med resultater for et scenario er vist i Appendiks A.4 som utgangspunkt for resultatbeskrivelse og databehandling av informasjon fra analysemodellens optimalløsning.

Resultatene som løsningsalgoritmen kommer frem til, skrives ut på en rekke filer. Tallmengden er så stor at det er vanskelig å sette seg inn i løsningene ved bare å se på verdiene til de enkelte variablene. Derfor beregnes det i tillegg summer og gjennomsnitt som skal gjøre det mulig å få et overblikk på kortere tid. Hvis man ønsker å få beregnet andre uttrykk enn de som inngår, kan man legge dem inn i modellfilen. De filene som skrives ut er kort beskrevet i utdrag fra eksemplet vist i Appendiks A.4.3.

I tillegg til summer er det også utviklet Excel regneark med makroer som gir grafisk fremstilling av oversiktsdata med diagram fra eksemplet vist i Appendiks A.4.4.

K2, informasjonsbehandling og representasjon av strategier for optimering basert på spillteori innebærer den største utfordring for modellen. En videreutvikling for å kunne representere ulike strategier på en hensiktsmessig måte vil derfor være ønskelig. En mulig utvidelse av den begrensede strategitypen til å angi f eks en andel av ulike styrkekomponenter som tillates å utnytte et gitt nodesett kan gi ytterligere fleksibilitet for å tilnærme resultat fra andre studier og kartspill.

For ytterligere grafisk fremstilling av situasjonsoversikter vil også resultatfilene kunne viderebehandles som input for en 'plug-in' funksjon direkte i STRIDSLOGG for å presentere løsningen dynamisk sfa tid, område (med assosiert UTM-koordinat), oppdragstype og komponenttype. Antallet av en komponenttype kan aggregeres i enheter av f eks 3 for å tilnærme en vanlig hierarkisk organisatorisk oppbygging i presentasjonene ved et enkelt filter. Dette vil gi en dokumentasjon av analysemodellens resultater som tilsvarer den form som ellers benyttes ved bruk av STRIDSLOGG (39) og vil kunne være en del av videre arbeid med analysemodellen MSO².1

FORKORTELSER

AA	Anti-Air
AMPL	A Mathematical Programming Language
C2	Command and Control
C2W	Command and Control Warfare
CJTF	Combined Joint Task Force (Multinasjonal Felles Task-styrke)
DCA	Defensive Counter Air
HOTDOG	Heuristisk Oppskrift Til å Determinere Optimum Globalt (Heuristic On The Determination of Optimum Globally)
IRS	Intelligence, Reconnaissance and Surveillance
JSTARS	Joint Surveillance Targeting And Reconnaissance System
K2	Kommando og Kontroll
MICIO	MInimum Computation by Iterative Optimization
MIKS	Modell for Innfasing av Kosteffektiv Struktur
MSO ²	Modell for StrukturOptimering i Multi-Scenario Operasjoner (Model for Structure Optimization in Multi-Scenario Operations)
OCA	Offensive Counter Air
OOB	Order Of Battle
ROE	Rules Of Engagement
SEAD	Suppression of Enemy Air Defence
SPD	Strid På Dypet
UCAV/UAV	Unmanned (Combat) Air Vehicle

APPENDIKS

A MODELL I AMPL

Det følgende er en oversikt over sannsynlighetsmodell for målfunksjonen $Prob(\text{egen vinn})$ og en detaljert listing av analysemodellen i AMPL. En beskrivelsen av løsningsverktøyet HOTDOG/MICIO er også gitt med en mal for modellutvikling ved bruk av dette verktøyet. Til slutt gis et eksempel på bruk av analysemodellen med data basert på et scenario i Finnmark.

A.1 Sannsynlighetsmodell for målfunksjon

Det er utviklet stokastiske ligninger for å estimere $Prob(\text{vinn} | \text{styrkeforhold})$, men disse er ofte svært vanskelige å løse (4)(5)(7). En forenklet betraktning basert på en Markov-prosess er beskrevet i (32)(33) som følger der:

$$\begin{aligned} X_0, Y_0 &= \text{initielle egne og motstander-styrker} \\ \text{eff}_x, \text{eff}_y, (r = \sqrt{\text{eff}_x / \text{eff}_y}) &= \text{egen og motstanders effektivitet (} r = \text{relativ effekt) beregnet} \\ &\text{som vist til slutt i dette kapitlet} \end{aligned}$$

Sannsynligheten for egen vinn (utfra binomiske forsøk) kan da tilnærmes ved integralet av Normalfordelingen (når total styrke > 4):

$$\begin{aligned} \Pr(X_0 \text{ vinn} | X_0, Y_0) &\cong \frac{1}{\sqrt{2\pi}} \int_{-\infty}^w e^{-t^2/2} \cdot dt \\ w &= \frac{(rX_0 - Y_0)\sqrt{3}}{\sqrt{r(X_0 + Y_0)}} \end{aligned} \tag{A.1}$$

der w er bestemt utfra Lanchesters 2. lov (1. lov kan tilpasses ved å fjerne $\sqrt{3}$). Dette uttrykket for w gjelder for slutt-tilstand $\rightarrow 0$ og er benyttet i målfunksjonen for analysemodellen MSO².1 i ligning (2.1) foran.

En approksimasjon for direkte beregning av sannsynligheten for å vinne $\Pr(X_0 \text{ vinn} | X_0, Y_0) = \Pr(w)$ kan utvikles basert på en tilnærmelse av den kumulative standard Normal-fordelingen:

$$\Pr(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^w e^{-t^2/2} \cdot dt \cong \frac{1}{2} \left[1 + \text{sign}(w) \left[1 - e^{-2w^2/\pi} \right]^{\frac{1}{2}} \right] \tag{A.2}$$

Denne tilnærmelsen er også benyttet i målfunksjonen i ligning (2.1) foran.

Dersom man ønsker å utvide kriteriet for å vinne til at f eks gjenværende styrker skal være minst X_s, Y_s , kan en korreksjon for w bestemmes fra to deterministiske approksimasjoner (33). Ligningen for w over gir middelerdi $X_M = \frac{Y_0}{r}$ som initialverdi for $\Pr_{0.50}$, mens Lanchesters 2. lov gir $X_L = \sqrt{X_s^2 + \frac{Y_0^2 - Y_s^2}{r^2}}$. For $X_s, Y_s = 0$ er disse like, og differensen $\Delta X = X_M - X_L$ kan være et

korreksjonsledd til w i ligningen over i det generelle tilfelle der ønsket nivå av gjenværende styrker X_s, Y_s er spesifisert tilsvarende et avbruddskriterium med tapsfraksjon hhv p, q . Da kan den modifiserte verdi for w bestemmes ved følgende ligning:

$$w = \frac{(rX_p - Y_0)\sqrt{3}}{\sqrt{r(X_p + Y_0)}}$$

der

$$X_p = X_0 + Y_0 / r - \sqrt{X_s^2 + \frac{Y_0^2 - Y_s^2}{r^2}}$$

$$= X_0 + Y_0 / r - \sqrt{(1-p)^2 X_0^2 + \frac{q(2-q)Y_0^2}{r^2}}$$
(A.3)

Dette uttrykket gir mulighet for å beregne på en enkel måte den nødvendige egne initialstyrke X_0 som må til for å vinne med gitt sannsynlighet, f eks $\text{Pr}_{0,90}$ eller $\text{Pr}_{0,95}$, når kriteriet er ønskede gjenværende styrker = X_s, Y_s (evt i % av initielle styrker, f eks 80 % egne og 50 % motstander) :

$$X_0 = X_p - X_M + X_L$$

der X_p finnes av 2. grads - ligningen:

$$(rX_p - Y_0)^2 - w_p^2 \cdot \frac{r}{3} \cdot (X_p + Y_0) = 0$$

og w_p er gitt av kumulativ std Normal - fordeling ($w_{0,90} = 1.29, w_{0,95} = 1.65$) :

$$w(P) \cong \sqrt{2\pi} \cdot \text{sign}(P - 0.5) \cdot \ln \left[\frac{0.5}{\sqrt{P(1-P)}} \right]^{\frac{1}{2}}$$

der $w(P)$ er en approksimasjon for den inverse funksjon av $\text{Pr}(w)$. Settes denne inn i 2.grads-ligningen for X_0 over, fås følgende tilnærmede uttrykk for nødvendig styrkebehov X_0 (etter en del Mathematica operasjoner):

$$X_0(y, r, p, q, w) \cong \left[w^2 + w\sqrt{w^2 + 12y(1+r)} + \sqrt{2\sqrt{(pw)^2(w^2 + 6y(1+r)) + 18y^2(1-p)(1+p)(1-q)(1+q)} + w(pw)^2\sqrt{w^2 + 12y(1+r)}} \right] / (6r(1-p)(1+p))$$

der $y, r = \sqrt{\text{eff}_x / \text{eff}_y}, 100 \cdot p, 100 \cdot q, w(\text{Pr})$ er hhv gitt motstanderstyrke, relativ effektivitet, egen %-vis reststyrke og motstandertap, samt ønsket vinn-sannsynlighet Pr . Uttrykket er kanskje litt komplisert, men bør vel være grei som formel betraktet. (Som et eksempel fås at for like effektive styrker, avbruddskriterium 20 % tap for egne og 50 % tap for motstanderen, vil et styrkeforhold på minst 3 gi vinn-sannsynlighet 0.9 for $y=10$, avtagende til nødvendig styrkeforhold 2 for $y=40$).

Til slutt vises hvordan effektivitetskoeffisientene $\text{eff}_x, \text{eff}_y, (r = \sqrt{\text{eff}_x / \text{eff}_y})$ kan beregnes utfra dynamisk styrkeutvikling fra lavere nivå modeller. Det antas at X_0, Y_0, X_T, Y_T er styrkene ved hhv start og avbrudd fra en slik modell etter en tid T . Med Lanchester-ligninger som aggregert modell kan effektivitetskoeffisientene beregnes :

$$eff_x = \frac{Y_0}{X_0} \cdot \frac{\gamma}{\rho \cdot T} \quad , \quad eff_y = \frac{X_0}{Y_0} \cdot \frac{\gamma \cdot \rho}{T}$$

der

$$\rho = \frac{Y_0}{X_0} \sqrt{\frac{X_T^2 - X_0^2}{Y_T^2 - Y_0^2}} \quad , \quad \gamma = \text{Ln}[\beta + \sqrt{1 + \beta^2}] \quad (\text{A.4})$$

$$\beta = \frac{\rho}{1 - \rho^2} \left(\frac{X_T}{X_0} - \frac{Y_T}{Y_0} \right)$$

Disse betraktningene er gjort for homogene styrker, og en sammensatt styrke må evt vektas (ved aggregering) f eks basert på egenverdier fra lavere nivå modeller til et samlet totalt styrkemål som i analysemodellens målfunksjonen i ligning (2.1). Ved å skalere styrkemålene til det ekvivalente antallet komponenter ved start NX_0, NY_0 og bruke de samme relative tap $X_T / X_0, Y_T / Y_0$ som gir et tilsvarende antall komponenter ved avbrudd lik $NX_T = (X_T / X_0)NX_0, NY_T = (Y_T / Y_0)NY_0$ i ligning (A.4), blir den relative utviklingen den samme og et uttrykk for komponentenes aggregerte effektivitetskoeffisienter.

I uttrykket for styrkeforholdet w i ligning (A.1) kan en ytterligere skalafaktor d settes inn for å kalibrere sannsynligheten for vinn $\Pr(NX_0 \text{ vinn} | NX_0, NY_0) = \Pr(w)$ til f eks anslag fra erfarne offiserer utfra bedømmelse av alle relevante faktorer. En slik skalafaktor d er også benyttet i målfunksjonen i ligning (2.1) for analysemodellen (parameter *scale* i datafilen i eksemplet i Appendiks A.4.1).

A.2 Modell i AMPL

En listing av analysemodellen MSO².1 skrevet i AMPL (38) viser oppbygging og innhold som beskrevet foran i kapittel 2.2 med modelldefinisjon og relasjoner (*.mod*), beskrankninger (*.con*) og kjørefil (*.run*).

Deretter følger i neste kapittel A.3 en beskrivelse av mal for modellutvikling og av løsningsverktøyet HOTDOG/MICIO som er benyttet ved kjøring av analysemodellen.

Til slutt vises eksempel på kjøring med gjennomgang av datafil (*.dat*) og resultatfiler basert på et scenario i Finnmark. En mer detaljert beskrivelse av datafil og kjøreoppsett er gitt i en egen brukerveiledning for modellen (37).

A.2.1 Modelldefinisjon: Mengder, variable og relasjoner

En modellfil gjenkjennes av AMPL ved filtype *.mod*. Denne kan plasseres i valgfri katalog og vil inneholde den generelle modellbeskrivelsen. Her deklarerer blant annet alle mengder og variable. Ligningene og ulikhetene som utgjør den matematiske beskrivelsen av problemet, er også formulert med generelle mengdeuttrykk. Denne filen er det normalt ikke nødvendig å endre fra kjøring til kjøring.

Modellfilen bør ellers i størst mulig grad benytte definerte variable der det er mulig, istedenfor å innføre nye ligninger i modellen. Dette er gjort i utstrakt grad for analysemodellen.

Det følgende er en listing av modellfilen (*.mod*) for analysemodellen MSO².1. Den er i hovedsak bygd opp som vanlig for AMPL (38). Imidlertid er det en rekke variable som man naturlig ville deklarerert med forskjellige navn, som er blitt samlet under navnet "calc" som beskrevet foran i kapittel 2.2.4.5. Det gjelder følgende variable i listingen nedenfor:

'din'	styrker som kommer inn i en node (tilsvarer 'tilført' i kapittel 2.2.4.5)
'pol'	hjelpvariable; tapspåførende styrke
'TARGET'	hjelpvariable; nevner i tapsvariablene 'lossf' ... etc
'ISR'	antall oppklaringskomponenter innen dekning
'C2'	antall C2-komponenter innen dekning
'lossf'	tap til artilleri (tilsvarer det generelle 'tap' i kapittel 2.2.4.5)
'lossac'	tap til fly (tilsvarer også som for 'lossf')
'lossm'	tap til manøverstyrker (tilsvarer også som for 'lossf')
'lossaac'	tap av fly til andre fly og luftvern (tilsvarer også som for 'lossf')
'disp'	disponible styrker
'ratio'	forhold mellom egne og motstanders manøverstyrker
'dforce'	styrker som flyttes (tilsvarer 'avgitt' i kapittel 2.2.4.5)
'force'	styrker som står i en node (tilsvarer 'styrke' i kapittel 2.2.4.5)

Grunnen til at disse er deklarerert sammen, er at det gjør det mulig å definere dem. Dermed oppfattes de ikke som variabler av løsningsalgoritmen, men som interne definisjoner (tilsvarende makroer). Dette gir et redusert antall variable og derved en mindre problemstørrelse. Alternativt måtte noen av dem vært frie variable, og man måtte brukt likhetsbeskravninger for å gi disse riktig verdi. Man kan nemlig ikke definere dem hver for seg fordi de er avhengige av hverandre (f eks 'disp' og 'force').

Modellfilen er inndelt i en deklarasjonsdel for mengder, parametre og de frie variable, samt en definisjonsdel for de variable som er definert, inklusive *f* som er det generiske (reserverte) navnet på objektivfunksjonen (målfunksjonen) for HOTDOG/MICIO løsningsalgoritme (se nedenfor).

```
# MSO2.1 : Model for StructureOptimization in Multi-Scenario Operations
# CHAPTER 1: MULTISIDE, MULTISCENARIO AND MIXED STRATEGY PARAMETERS

param dam {x in 1..s,y in 1..s} >=0 default if x=y then 0 else 1;
# Damage distribution between the sides
param res {x in 1..s,y in 1..s} default if x=y then 1 else -1;
# The linear weights of each side used as coefficients in linear objective

param n integer >=0 default 0;
param weight {0..n} >0 default 1;
# Number of scenarios (n+1) and their respective weights

set parts {0..n} within {1..s} default {1..s};
# The sides that take part in each scenario

param m {u in 0..n, parts[u]} in {1..9} default 1;
# Number of strategies on each side
set cum {u in 0..n, x in parts[u]} within 1..m[u,x] default 1..m[u,x];

print 'set comb {u in 0..n} := setof {{1}', {x in 1..s} ', a' & x & ' in cum[u,' & x &
'],' , ' ) 0', {x in 1..s} '+a' & x & '*10^' & x-1 & '* (if ' & x & ' in parts[u] then
1)', ';' > comb;
print 'set com {u in 0..n} := setof {{1}', {x in 1..s} ', a' & x & ' in 1..m[u,' & x &
'],' , ' ) 0', {x in 1..s} '+a' & x & '*10^' & x-1 & '* (if ' & x & ' in parts[u] then
1)', ';' > comb;
```

```

include comb;
# The set of all possible combinations of strategies, represented by numbers where
each digit represents the strategy of a side

param str {(x,c) in setof {u in 0..n, x in parts[u], c in com[u]} (x,c)} :=
floor(c/10^(x-1))-10*floor(c/10^x);
# The strategy of a particular side in a given combination

param stype {u in 0..n, x in 1..s, 1..m[u,x]} integer in {0..2} default 0;
# Strategy type:      0=Free 1=Limited 2=Fixed

param min_prob {u in 0..n, x in parts[u]} <= 0.7/m[u,x] default 0.4/m[u,x];

param info {x in 0..s} binary default if x=0 then 1 else 0;
# Decides whether or not information aspects are included in the model

param def {0..n} in {0..s} default 0;
# Decides which side is allowed to start in oar

param lino 'nonlinear objective' binary default if s<3 then 1 else 0;      # lino=0 is
linear obj., else nonlinear
param all_count binary default lino;                                     # all_count=0 means
only last time period counts, else all periods count

param eps 'low limit' := 1e-3 ; # To avoid div by zero in ratios

# CHAPTER 2: COMPONENT SETS AND PARAMETERS

set ty 'components';          # Members of the components
param si {ty} in {1..s};     # The side each component belongs to

set ta 'tasks' {ty} ordered;          # Members of the tasks
set isr 'isr-class' within {ty};     # ISR classes of the component groups
set c2 'c2-class' within {ty};       # C2 classes of the component groups
set fir 'fire-class' within {ty};    # FIRE classes of the component groups
set man 'man-class' within {ty};     # MANEUVER classes of the component groups
set ac within {ty};                # AIRCRAFT classes of the component groups
set aac within {ty};              # ANTI-AIRCRAFT classes of the component groups

set now := {i in ty :si[i]=cu};      # The components belonging to that side

param cost 'component cost' {ty} > 0 default 1e3;
# Nominal cost per own component, used
# to compute total structure costs

param budsj {1..s} >=0 default 0; # Total budget constraint

param arv {u in 0..n, i in ty} default if u=0 then 0 else arv[0,i]; # Initial
components

param scale >=0 default 1;
param w {u in 0..n, i in ty} default (if u=0 then cost[i] else w[0,i]);
# Ideal linear weights = eigenvalues for components (relative)

param frt 'fire-rate' {i in fir union ac union aac} >0 default Infinity;
# Mission-rate pr hour for FIRE class components (art,ac,heli)

param max_r 'max range' {fir union c2 union isr union ac} > 0 default 2;
# Max Range for fir+c2+isr+ac types (art,ac,hq,recce etc.)

param c2_cap 'span of control capacity pr c2-component' {i in c2} >=1 default 3;
param max_eff 'max effectiveness factor for c2+isr' {u in 0..n} > 0
default (if u=0 then 1.3 else
max_eff[0]) ;
param min_eff 'min effectiveness factor for c2+isr' {u in 0..n} > 0
default (if u=0 then 0.7 else
min_eff[0]) ;
param rate_eff 'effectiveness increase rate for c2+isr' {u in 0..n} > 0
default (if u=0 then 0.7 else
rate_eff[0]) ;
# Parameters to include effect of c2+isr as increased/decreased force effectiveness

param low 'effectiveness factor without artillery fire' {i in man} >=1 default 1;
param high 'effectiveness factor with heavy artillery fire' {i in man} <=1 default 1;
param norm 'normal losses to artillery' {i in man} >0 default 0.01;

param col 'comp. column length' {ty} > 0;          # Component column length on
axis

```



```
### CHAPTER 3: TERRAIN GRAPH + PARAMETERS ###
```

```

set ar 'areas' ordered;                # Nodes of the terrain graph
set base {ac} within ar default ar;    # Airfields
set sar {0..n} within ar default ar;
set ad 'adjacents' {ar} within ar;     # Adjacents of nodes
set trty 'terraintypes';              # Terraintypes

set iar 'initial area' {u in 0..n, x in parts[u]} within sar[u] default if u=0 then {}
else iar[0,x];                        # Initial deployment area

set oar {u in 0..n} := sar[u] diff (union {x in parts[u]} iar[u,x]);          #
Operational area = Total area different from initial opponent area

set lar {x in 1..s, u in 0..n, ø in cum[u,x] : stype[u,x,ø]=1} within iar[u,x] union
oar[u] default iar[u,x] union oar[u];

set tyo {u in 0..n, c in comb[u], o in ar} := {i in ty : si[i] in parts[u] and (if i
in ac then (if o in base[i] then 1 else 0) else 1) and o in oar[u] union iar[u,si[i]]
and if stype[u,si[i],str[si[i],c]]=1 then (if o in lar[si[i],u,str[si[i],c]] then 1)
else 1};
set be {u in 0..n, i in ty, st in cum[u,si[i]]} := {o in ar : si[i] in parts[u] and (if
i in ac then (if o in base[i] then 1 else 0) else 1) and o in oar[u] union
iar[u,si[i]] and if stype[u,si[i],st]=1 then (if o in lar[si[i],u,st] then 1) else 1};

param terr {ar} in {trty} ;           # Terraintypes in nodes

param max_in_node 'max no in area' {ar} >= 0 default Infinity;
# Max nos of xty+yty per node

param axis_cap 'axis capacity' {o in ar, ad[o]} >= 0, default 1 ;
# Max no of lanes for axis from node o to its adjacents

param max_speed 'max speed for terraintype' {ty, trty} >= 0 ;
# Max speed of xty+yty for terrain

param r 'distances in graph' {setof {u in 0..n, o in sar[u], oo in sar[u]} (o, oo)} ;
# Distance (km) between nodes

param pri 'priority of a node' {ar} >= 0, default 1; # Priority weight of each node

```

```
### CHAPTER 4: TIME PERIOD SETS AND PARAMETERS ###
```

```

param las {u in 0..n} default if u=0 then 20 else las[0], >2, integer;      #
Index of last period to be modeled

set time 'scenario horizon' {u in 0..n} := 1..las[u];

set otime 'operational time' {u in 0..n} := time[u] diff {1};

param L {u in 0..n} >0 default if u=0 then 1 else L[0] ;                    #
Time-period (hrs) corresp. with xeff/yeff Lanch-coeffs.

```

```
### CHAPTER 5: DEFINITIONS OF SETS ###
```

```

set Per {u in 0..n, c in comb[u]} := setof {i in ty, a in ta[i], o in if i in ac then
be[u,i,str[si[i],c]] else oar[u] inter be[u,i,str[si[i],c]]} (i,a,o);
set Roma := setof {u in 0..n, i in ty, st in cum[u,si[i]], o in be[u,i,st], t in
time[u] : si[i] in parts[u]} (u,i,o,t,st);
set Styrke := setof {(u,i,o,t,st) in Roma, c in comb[u] : t>1 and str[si[i],c]=st}
(u,i,o,t,c);
set Tap := setof {(u,i,o,t,c) in Styrke, a in ta[i] : o in oar[u] or i in ac}
(u,i,a,o,t,c);
set Flytt := setof {(u,i,o,t,c) in Styrke, oo in be[u,i,str[si[i],c]] inter (if o in
oar[u] then oar[u] else oar[u] union iar[u,si[i]]) inter if i in ac then base[i] diff
{o} else ad[o]} (u,i,o,oo,t,str[si[i],c]);

```

CHAPTER 6: EFFECTIVENESS AND ENGAGE-FRACTION PARAMETERS

```

param eff 'Lanch. eff. coeff' {u in 0..n, i in man union fir union ac union aac, j in
ty, k in trty, a in ta[i] union {0}, b in ta[j] union {0}} >=0
  default (if b!=0 then eff[u,i,j,k,a,0] else if u=0 and a=0 then 0 else if u=0
then eff[u,i,j,k,0,b] else eff[0,i,j,k,a,b]) ;
  # Effectiveness pr engagement of ty vs ty in terr for task ta

param eng 'Lanch. eng. coeff' {u in 0..n, i in man union fir union ac union aac, j in
ty, k in trty, a in ta[i] union {0}, b in ta[j] union {0}} >=0
  default (if b!=0 then eng[u,i,j,k,a,0] else if u=0 and a=0 then 1 else if u=0
then eng[u,i,j,k,0,b] else eng[0,i,j,k,a,b]) ;
  # Engagemtrate of ty vs ty in terr for task ta

param task_creator {i in man union fir union ac union aac, ta[i], ty} >=0 default 1;

param reff 'relative effectiveness' {u in 0..n, i in man union fir union ac union aac,
j in ty, k in trty, a in ta[i], b in ta[j]} :=
  task_creator[i,a,j]*
  if i in ac union aac then w[u,j]*eff[u,i,j,k,a,b]*eng[u,i,j,k,a,b]
  else eff[u,i,j,k,a,b]*eng[u,i,j,k,a,b]/(1e-8+sum {ii in fir union man, aa in
ta[ii] :si[i]=si[ii]} (eff[u,ii,j,k,aa,b]*eng[u,ii,j,k,aa,b]
/(1e-8+card(ta[ii])*sum {jj in ty, bb in ta[jj]}
w[u,jj]*eff[u,ii,jj,k,aa,bb]*eng[u,ii,jj,k,aa,bb]/card(ta[jj]))) ;

param lowest_eff {i in 1..2} default if i=1 then 0.004 else 0.0003;

```

CHAPTER 7: VARIABLES DECLARATIONS AND DEFINITIONS

```

var p2 {u in 0..n, x in parts[u], st in 2..m[u,x]} >=(if card(cum[u,x])=1 then (if st
in cum[u,x] then 0.2) else min_prob[u,x]) <=(if card(cum[u,x])=1 then (if st in
cum[u,x] then 1 else 0.8) else 1-(m[u,x]-1)*min_prob[u,x]) default 1/m[u,x];
var p1 {u in 0..n, x in parts[u]} = 1-sum {st in 2..m[u,x]} p2[u,x,st];
var p {u in 0..n, x in parts[u], st in 1..m[u,x]} = if st=1 then p1[u,x] else
p2[u,x,st];
# Probability of choosing each strategy

var start {u in 0..n,i in ty,o in sar[u], st in cum[u,si[i]} :si[i] in parts[u] and o
in (if si[i]=def[u] then be[u,i,st] else be[u,i,st] inter iar[u,si[i]])} >=(if i='E'
then 2 else 0) <=(if si[i]=2 then 30 else if i='E' then 12 else 8);
var task1 {(u,i,o,t,st) in Roma, ta[i] diff {last(ta[i])}} >=0 <=1 default
1/card(ta[i]);
var task2 {(u,i,o,t,st) in Roma} = 1-sum {a in ta[i] diff {last(ta[i])}}
task1[u,i,o,t,st,a];
var task {(u,i,o,t,st) in Roma, a in ta[i]} = if a=last(ta[i]) then task2[u,i,o,t,st]
else task1[u,i,o,t,st,a];
var dffrac {(u,i,o,oo,t,st) in Flytt} <=1 >=0;
var acfrac {(u,i,o,t,st) in Roma, oar[u] :i in ac} >=0 <=1;

set calcvars :=
{'din','pol','TARGET','ISR','C2','lossf','lossac','lossm','lossaac','disp','ratio','df
orce','force'};
set calcindx := {u in 0..n, c in comb[u], t in 1..las[u], na in calcvars, o in sar[u],
i in tyo[u,c,o] union if na='TARGET' then ac else {0}, a in if i=0 then {0} else ta[i]
union {0}, x in 0..s, oo in if na='dforce' then {(if o in oar[u] then oar[u] else
oar[u] union iar[u,si[i]}) inter be[u,i,str[si[i],c]} inter if i in ac then base[i]
diff {o} else ad[o]) else {0}
:(s=2 or na!='ratio') and info[x]=1 and (t>1 or na='force') and (na not in {'TARGET'
or i in man union fir union ac union aac) and (na!='pol' or i in ac or o in oar[u])
and (i in ac or na!='lossaac') and (x in parts[u] or na not in {'ISR','C2'}) and (x=0
or na in {'ISR','C2'}) and (x!=0 or na not in {'ISR','C2'}) and (a=0 or na in
{'pol','TARGET','lossf','lossac','lossm','lossaac'}) and (a!=0 or na not in
{'pol','TARGET','lossf','lossac','lossm','lossaac'}) and (na not in
{'ISR','C2','ratio'} or i=0) and (na in {'ISR','C2','ratio'} or i!=0) and (o in oar[u]
or na not in {'TARGET','ISR','C2','lossf','lossac','lossm'})};
var calc {(u,c,t,na,o,i,a,x,oo) in calcindx} = (

# Dynamic equations w/movement (for each area diff iar):
# Force now = Force previous - loss + force_into - force_out

if na='disp' then max(0,
  calc[u,c,t-1,'force',o,i,0,0,0]
+ calc[u,c,t,'din',o,i,0,0,0]
- sum {aa in ta[i] :o in oar[u] or i in ac} (sum {{1} :o in oar[u])

```

```

(calc[u,c,t,'lossm',o,i,aa,0,0]+calc[u,c,t,'lossf',o,i,aa,0,0]+calc[u,c,t,'lossac',o,i
,aa,0,0])+sum {{1} :i in ac} calc[u,c,t,'lossaac',o,i,aa,0,0])
)
)+(
if na='force' then (if t=1 then (if (o in iar[u,si[i]] or def[u]=si[i]) then
start[u,i,o,str[si[i],c]]) else max(0,calc[u,c,t,'disp',o,i,0,0,0] - sum {ooo in
be[u,i,str[si[i],c]] inter (if o in oar[u] then oar[u] else oar[u] union iar[u,si[i]])
inter if i in ac then base[i] diff {o} else ad[o]) calc[u,c,t,'dforce',o,i,0,0,ooo]))
)
)+(
if na='ratio' then (sum {ii in tyo[u,c,o] inter man :si[ii]=cu}
calc[u,c,t,'disp',o,ii,0,0,0]*w[u,ii]+1e-6)/(sum {ii in tyo[u,c,o] inter man
:si[ii]!=cu} calc[u,c,t,'disp',o,ii,0,0,0]*w[u,ii]+1e-6)
)
)+(
if na='dforce' then calc[u,c,t,'disp',o,i,0,0,0]*if s!=2 or o not in oar[u] then
dfrac[u,i,o,oo,t,str[si[i],c]] else min( dfrac[u,i,o,oo,t,str[si[i],c]], max(0, (if
si[i]=cu then calc[u,c,t,'ratio',o,0,0,0,0] else 1/calc[u,c,t,'ratio',o,0,0,0,0])*0.6-
0.3 , (if si[i]=cu then calc[u,c,t,'ratio',oo,0,0,0,0] else
1/calc[u,c,t,'ratio',oo,0,0,0,0])*1.5-0.5 ) )
# Force in node o moved to adj. node at time
)
)+(
if na='din' then
if i in ac then
sum {ooo in (if o in oar[u] then be[u,i,str[si[i],c]] else
be[u,i,str[si[i],c]] inter iar[u,si[i]]) diff {o}, tt in otime[u]
: t=tt+max(1,trunc((r[o,ooo]+col[i])/(max_speed[i,terr[ooo]]*L[u]))))
calc[u,c,tt,'dforce',ooo,i,0,0,o]
else
sum {ooo in ad[o] inter (if o in oar[u] then be[u,i,str[si[i],c]] else
be[u,i,str[si[i],c]] inter iar[u,si[i]]) , tt in otime[u]
: t=tt+max(1,trunc((r[o,ooo]+col[i])/(max_speed[i,terr[ooo]]*L[u]))))
calc[u,c,tt,'dforce',ooo,i,0,0,o]
# Forces into node o at time t from previous moves
)
)+(
if na='pol' then (calc[u,c,t-1,'force',o,i,0,0,0]
+calc[u,c,t,'din',o,i,0,0,0])*task[u,i,o,t-1,str[si[i],c],a]
)
)+(
if na='TARGET' then sum {(ii,aa,ooo) in Per[u,c] : (if i in man union ac then (if ooo=o
then 1) else 1) and (if i in fir then (if max_r[i]>=r[o,ooo] and ooo in oar[u] then 1)
else 1) and (i in ty diff ac or ii in ty diff ac)}
w[u,ii]*(calc[u,c,t,'pol',ooo,ii,aa,0,0]+eps)*reff[u,i,ii,terr[ooo],a,aa]*dam[si[i],si
[ii]]
)
)+(
if na='ISR' then 2*eps + sum {ii in isr, ooo in oar[u] : si[ii]=x and
max_r[ii]>=r[o,ooo]} calc[u,c,t-1,'force',ooo,ii,0,0,0]
)
)+(
if na='C2' then 2*eps + sum {ii in c2, ooo in oar[u] : si[ii]=x and
max_r[ii]>=r[o,ooo]} calc[u,c,t-1,'force',ooo,ii,0,0,0]
)
)+(
if na='lossf' then L[u]*w[u,i]*(calc[u,c,t,'pol',o,i,a,0,0]+eps)*sum {(j,b,0) in
Per[u,c] : j in fir and max_r[j]>=r[o,0] and o in oar[u]} (if info[si[j]]=0 then 1 else
(max_eff[u]-(max_eff[u]-min_eff[u])*exp(-
rate_eff[u]*sqrt(eps+calc[u,c,t,'C2',o,0,0,si[j],0]*calc[u,c,t,'ISR',o,0,0,si[j],0])))
)*calc[u,c,t,'pol',o,j,b,0,0]*eff[u,j,i,terr[o],b,a]*eng[u,j,i,terr[o],b,a]*reff[u,j,i
,terr[o],b,a]*dam[si[j],si[i]]/calc[u,c,t,'TARGET',o,j,b,0,0]
)
)+(
if na='lossac' then
if i in ac then 0 else
w[u,i]*(calc[u,c,t,'pol',o,i,a,0,0]+eps)*sum {(j,b,0) in Per[u,c] :
L[u]>=2*r[o,0]/max_speed[j,terr[0]] and j in ac} ((L[u]-
2*r[o,0]/max_speed[j,terr[0]])*calc[u,c,t,'pol',o,j,b,0,0]*acfrac[u,j,0,t,str[si[j],c]
,o]*eff[u,j,i,terr[o],b,a]*eng[u,j,i,terr[o],b,a]*reff[u,j,i,terr[o],b,a]*dam[si[j],si
[i]]/calc[u,c,t,'TARGET',o,j,b,0,0]
)

```

```

    *if info[si[j]]=0 then 1 else (max_eff[u]-(max_eff[u]-min_eff[u])*exp(-
rate_eff[u]*calc[u,c,t,'ISR',o,0,0,si[j],0]))
)
)+(
if na='lossm' then
    L[u]*w[u,i]*(calc[u,c,t,'pol',o,i,a,0,0]+eps)*sum {b in union {tull in ty}
ta[tull], j in man inter tyo[u,c,o] : b in ta[j] and si[j] in parts[u] and o in
oar[u]}
calc[u,c,t,'pol',o,j,b,0,0]*eff[u,j,i,terr[o],b,a]*eng[u,j,i,terr[o],b,a]*reff[u,j,i,t
err[o],b,a]*dam[si[j],si[i]]/calc[u,c,t,'TARGET',o,j,b,0,0]
    *if info[si[j]]=0 then 1 else (max_eff[u]-(max_eff[u]-min_eff[u])*exp(-
rate_eff[u]*sqrt(eps+calc[u,c,t,'C2',o,0,0,si[j],0]*calc[u,c,t,'ISR',o,0,0,si[j],0]))
    *(high[j]-(high[j]-low[j])*((1-high[j]+eps)/(low[j]-
high[j]+eps)))^((calc[u,c,t,'lossf',o,j,b,0,0]+calc[u,c,t,'lossac',o,j,b,0,0])/(L[u]*no
rm[j]*calc[u,c,t,'pol',o,j,b,0,0]+eps)))
)+(
if na='lossaac' then
    calc[u,c,t-1,'force',o,i,0,0,0]*task[u,i,o,t-1,str[si[i],c],a]*(sum {O in
oar[u] : L[u]>=2*r[o,0]/max_speed[i,terr[O]]} acfrac[u,i,o,t,str[si[i],c],O]*(1-exp(-
1.3355*(
    w[u,i]*L[u]*sum {(j,b,O) in Per[u,c] : j in ac} calc[u,c,t-
1,'force',o,j,0,0,0]*task[u,j,O,t-1,str[si[j],c],b]*(sum {ooo in oar[u] :
L[u]>=2*r[ooo,0]/max_speed[i,terr[ooo]]}
acfrac[u,j,0,t,str[si[j],c],ooo]*eff[u,j,i,terr[O],b,a]*eng[u,j,i,terr[O],b,a]*reff[u
,j,i,terr[O],b,a]*dam[si[j],si[i]]/(eps+sum {(ii,aa,ooo) in Per[u,c] : ii in ac}
calc[u,c,t-1,'force',ooo,ii,0,0,0]*task[u,ii,ooo,t-1,str[si[ii],c],aa]*w[u,ii]*(sum
{oooo in oar[u] : L[u]>=2*r[ooo,ooo]/max_speed[i,terr[O]]}
acfrac[u,ii,ooo,t,str[si[ii],c],oooo]*reff[u,j,ii,terr[O],b,aa]*dam[si[j],si[ii]]}
+((sum {O in oar[u] : L[u]>=2*r[o,0]/max_speed[i,terr[O]]}
acfrac[u,i,o,t,str[si[i],c],O]*(1-exp(-1.3355*((L[u]-
2*r[o,0]/max_speed[i,terr[O]])*w[u,i]*sum {j in aac inter tyo[u,c,O], b in ta[j]
:si[j] in parts[u]}
calc[u,c,t,'pol',o,j,b,0,0]*eff[u,j,i,terr[O],b,a]*eng[u,j,i,terr[O],b,a]*reff[u,j,i,t
err[O],b,a]*dam[si[j],si[i]]/(sum {ii in ac, aa in ta[ii], ooo in
be[u,ii,str[si[ii],c]} : L[u]>=2*r[ooo,0]/max_speed[ii,terr[O]]} (L[u]-
2*r[ooo,0]/max_speed[i,terr[O]])*w[u,ii]*calc[u,c,t-
1,'force',ooo,ii,0,0,0]*task[u,ii,ooo,t-
1,str[si[ii],c],aa]*acfrac[u,ii,ooo,t,str[si[ii],c],O]*reff[u,j,ii,terr[O],b,aa]*dam[s
i[j],si[ii]]+1e-3))))))
    /(eps/10+sum {O in oar[u] : L[u]>=2*r[o,0]/max_speed[i,terr[O]]}
acfrac[u,i,o,t,str[si[i],c],O]))
);
# Two alternatives for Prob(win) calculation :
# 1) Start/End-Forces if all-count = 0 : Only first and last forces determine
forceratio and probwin
var start1 {u in 0..n, c in comb[u] :s=2 and all_count=0} = sum {i in ty,o in sar[u]
:si[i]=1 and o in (if si[i]=def[u] then be[u,i,str[si[i],c]] else be[u,i,str[si[i],c]]
inter iar[u,si[i]])} start[u,i,o,str[si[i],c]]*w[u,i];
var start2 {u in 0..n, c in comb[u] :s=2 and all_count=0} = sum {i in ty,o in sar[u]
:si[i]=2 and o in (if si[i]=def[u] then be[u,i,str[si[i],c]] else be[u,i,str[si[i],c]]
inter iar[u,si[i]])} start[u,i,o,str[si[i],c]]*w[u,i];
var end1 {u in 0..n, c in comb[u] :s=2 and all_count=0} = sum {i in ty,o in (if i in
ac then oar[u] union iar[u,si[i]] else oar[u]) inter be[u,i,str[si[i],c]] :si[i]=1}
calc[u,c,las[u],'force',o,i,0,0,0]*w[u,i];
var end2 {u in 0..n, c in comb[u] :s=2 and all_count=0} = sum {i in ty,o in (if i in
ac then oar[u] union iar[u,si[i]] else oar[u]) inter be[u,i,str[si[i],c]] :si[i]=2}
calc[u,c,las[u],'force',o,i,0,0,0]*w[u,i];
var fratio {u in 0..n, c in comb[u] :s=2 and all_count=0} = 0.35 *
sqrt(3*scale)*(sqrt(max(1e-6,(start2[u,c]^2-end2[u,c]^2)/
(start1[u,c]^2-end1[u,c]^2+1e-8)))*start1[u,c]-start2[u,c])/
sqrt(sqrt(max(1e-6,(start2[u,c]^2-end2[u,c]^2)/
(start1[u,c]^2-end1[u,c]^2+1e-8)))*(start1[u,c]+start2[u,c]));
var probwin {u in 0..n, c in comb[u] :s=2 and all_count=0} =
0.5*(1 +
(if fratio[u,c] >= 0 then 1 else -1) *
(sqrt(1-exp(-2*(fratio[u,c]^2)/3.14)+1e-8)-1e-4)
);
# 2) All Forces count if all-count = 1 : All nodes and time forces determine
forceratio and probwin (default)

```

```

var COEFF {u in 0..n, c in comb[u], o in oar[u], t in otime[u] :s=2 and all_count=1} =
(eps*lowest_eff[1]+sum {j in tyo[u,c,o], b in ta[j] :si[j]=2}
(calc[u,c,t,'lossm',o,j,b,0,0]+calc[u,c,t,'lossf',o,j,b,0,0]+calc[u,c,t,'lossac',o,j,b,0,0]+sum {{1} :j in ac} calc[u,c,t,'lossaac',o,j,b,0,0])*w[u,j])
*(eps+sum {j in tyo[u,c,o] :si[j]=2} calc[u,c,t-1,'force',o,j,0,0,0]*w[u,j])
/(eps+sum {i in tyo[u,c,o] :si[i]=1} calc[u,c,t-1,'force',o,i,0,0,0]*w[u,i])
/(eps*lowest_eff[2]+sum {i in tyo[u,c,o], a in ta[i] :si[i]=1}
(calc[u,c,t,'lossm',o,i,a,0,0]+calc[u,c,t,'lossf',o,i,a,0,0]
+calc[u,c,t,'lossac',o,i,a,0,0]+sum {{1} :i in ac}
calc[u,c,t,'lossaac',o,i,a,0,0])*w[u,i])
;

```

```

var fratiol 'force ratio' {u in 0..n, c in comb[u], o in oar[u], t in time[u] :s=2 and
all_count=1}=if t=1 then (if def[u]=1 then 1 else if def[u]=2 then -1) else
if COEFF[u,c,o,t]<=0 then (cu-1.5)*20 else
sqrt(3*scale)/sqrt(eps+ sqrt(COEFF[u,c,o,t])*sum {i in tyo[u,c,o]} calc[u,c,t-1,'force',o,i,0,0,0]*w[u,i] )*(sqrt(COEFF[u,c,o,t])*
sum {i in tyo[u,c,o] :si[i]=1} calc[u,c,t-1,'force',o,i,0,0,0]*w[u,i] -
sum {j in tyo[u,c,o] :si[j]=2} calc[u,c,t-1,'force',o,j,0,0,0]*w[u,j] )
; # Weighted forceratio x/y distributed approx. as a std. normal stochastic
variate

```

```

param use {x in 1..2,u in 0..n, c in comb[u],o in oar[u] :s=2 and all_count=1} := if
stype[u,x,str[x,c]]=1 then (if o in lar[x,u,str[x,c]] then 1) else 1;

```

```

var probwin1 'Prob. side 1 win' {u in 0..n, c in comb[u], o in oar[u], t in otime[u]
:s=2 and all_count=1} =
if use[1,u,c,o]=0 and use[2,u,c,o]=0 then 0.5 else if use[1,u,c,o]=0 then 0
else if use[2,u,c,o]=0 then 1 else 0.5*( 1 +
(if fratiol[u,c,o,t] >= 0 then 1 else -1) *
(sqrt(eps+1 - exp(-2*(fratiol[u,c,o,t]^2)/3.14))-sqrt(eps)) );

```

```

# Probability of x win as approximate cumulative normal distribution
# Maybe concave approx. for negative fratio can be used if difficulties.

```

```

param M2 'values for locked strategies' {u in 0..n, com[u], 1..s} default 0;

```

```

var M 'maximin force_diff' {u in 0..n, c in com[u], x in 1..s} =
if str[cu,c] in cum[u,cu] then

if lino=1 then (if all_count=0 then probwin[u,c]*(3-2*x) else
sum {o in oar[u], t in 3..las[u]} probwin[u,c,o,t]*pri[o]*(3-2*x) /
((las[u]-2)*sum {oo in oar[u]} pri[oo]) )

else
(if all_count=1 then sum {o in sar[u], i in tyo[u,c,o], t in 3..las[u] :si[i]
in parts[u] and lino=0 and (i in ac or o in oar[u])}
calc[u,c,t,'force',o,i,0,0,0]*w[u,i]*res[x,si[i]]*pri[o]/(las[u]-2)
else sum {o in sar[u], i in tyo[u,c,o] :si[i] in parts[u] and lino=0 and (i in
ac or o in oar[u])} calc[u,c,las[u],'force',o,i,0,0,0]*w[u,i]*res[x,si[i]]*pri[o])
*scale/sum {o in oar[u]} pri[o]
# Linear relation = Weighted difference of total forces
# Might influence the solution towards the components that have the
highest ratio between weight and cost

else M2[u,c,x];

```

```

var value {u in 0..n :cu in parts[u]} = min {c in setof {c in comb[u]} (c-
str[cu,c]*10^(cu-1))} sum {st in 1..m[u,cu]} p[u,cu,st]*M[u,c+st*10^(cu-1),cu];

```

```

### CHAPTER 8: OBJECTIVE FUNCTION(S) TO BE OPTIMIZED ###

```

```

# Force effectiveness = Diff x - y ; Ratio x/y or P(x win | x,y) later

```

```

var f = -(sum {u in 0..n :cu in parts[u]} weight[u]*value[u])/sum {u in 0..n :cu in
parts[u]} weight[u];

```

A.2.2 Beskrankninger

Det er også hensiktsmessig å samle alle beskrankninger i en fil med filtype *.con*. Dette er ikke generelt nødvendig i AMPL, men for å benytte det heuristiske løsningsverktøyet HOTDOG/MICIO er det laget en mal for hvordan beskrankninger uttrykkes for å tilpasses løsningsalgoritmen. Denne malen er vist i Appendiks A.3.2 og gir en veiledning til konstruksjon av beskrankninger.

Malen benytter AMPL's 'meta'-funksjoner der hver variabel gis et nummer (*j*) som kan refereres generisk ved en indeks (*_var[j]*) med korrespondanse til variabelnavnet (*_varname[j]*) for generell bruk. Dette utnyttes av løsningsalgoritmen som på denne måten tilordner beskrankninger til variable for å lage straffe-ledd i optimeringen. Dette gir mulighet for å separere beskrankningsvariable fra de deklarete variable som skal optimeres i modellen og teste gyldighet av beskrankninger separat.

Hver beskrankning må gis en egen variabel som angitt i Appendiks A.3.2, og AMPL genererer så de aktuelle instansvariablene på grunnlag av verdiene i datafilen.

Beskrankningsfilen vil normalt heller ikke endres fra kjøring til kjøring.

```
# MSO2.1 : Constraints
### CHAPTER 9: CONSTRAINTS DEFINITIONS ###

var Total_cost = sum {i in now} max(0,max {u in 0..n, st in cum[u,si[i]] : si[i] in
parts[u]} (sum {o in (if def[u]=si[i] then oar[u] union iar[u,si[i]] else
iar[u,si[i]]) inter if i in ac then base[i] else ar} start[u,i,o,st]-
arv[u,i]))*cost[i] - budsj[cu];
      # Initial force cost < budsj

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 1e-1;
let {i in QD..QM} Pen[i] := 1e3;
let QD:=QM+1;

# TERRAIN-CONSTRAINTS

var Move_cap {u in 0..n, c in comb[u], o in oar[u] union iar[u,cu], oo in (if o in
oar[u] then oar[u] else oar[u] union iar[u,cu]) inter ad[o], t in otime[u] :cu in
parts[u]} =
      sum {i in (now diff ac) inter tyo[u,c,o] inter tyo[u,c,oo]}
(calc[u,c,t,'dforce',o,i,0,0,oo] + if (o in oar[u] or oo in iar[u,si[i]]) then
calc[u,c,t,'dforce',oo,i,0,0,o])*col[i]
      - r[o,oo]*axis_cap[o,oo] ;
      # Current forces column length on axis <= axis_capacity length

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 2e-1;
let {i in QD..QM} Pen[i] := 0.3;
let QD:=QM+1;

var Max_depl {u in 0..n, c in comb[u], o in oar[u] union iar[u,cu], t in time[u] :cu
in parts[u]} =
      sum {i in tyo[u,c,o] inter man inter now} calc[u,c,t,'force',o,i,0,0,0] -
max_in_node[o] ;
      # Sum of depl. maneuver components <= Max in each node always

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 2e-1;
let {i in QD..QM} Pen[i] := 0.4;
let QD:=QM+1;
```

```

var task_limit {(u,i,o,t,æ) in Roma :si[i]=cu} =
  -task2[u,i,o,t,æ];
#   abs(sum {a in ta[i]} task[u,i,o,t,æ,a]-1);

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 3e-2;
let {i in QD..QM} Pen[i] := 2;
let QD:=QM+1;

var dfrac_limit {(u,i,o,t,æ) in Roma :t>1 and si[i]=cu} =
  sum {oo in be[u,i,æ] inter (if o in oar[u] then oar[u] else oar[u] union
iar[u,si[i]]) inter if i in ac then ar diff {o} else ad{o}) dfrac[u,i,o,oo,t,æ] - 1;

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 3e-2;
let {i in QD..QM} Pen[i] := 5;
let QD:=QM+1;

var acfrac_limit {(u,i,o,t,æ) in Roma :i in ac and si[i]=cu} =
  sum {oo in oar[u]} acfrac[u,i,o,t,æ,oo] - 1;

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 1e-1;
let {i in QD..QM} Pen[i] := 1;
let QD:=QM+1;

# CONSTRAINTS RELATED TO C2 AND INFORMATION

var C2move {u in 0..n, c in comb[u], o in oar[u], t in otime[u] : info[cu]=1 and cu in
parts[u]} =
  sum {i in man union fir union aac, oo in ad{o} inter oar[u] :si[i]=cu}
  calc[u,c,t,'dforce',o,i,0,0,oo]
  - sum {ii in c2, ooo in oar[u] :si[ii]=cu and max_r[ii]>r[o,ooo]}
  c2_cap[ii]*calc[u,c,t,'force',ooo,ii,0,0,0] ;
  # Moves from node o in time period [t, t+L] <= Sum of C2 span of
  control capacity near o

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 2e-1;
let {i in QD..QM} Pen[i] := 0.1;
let QD:=QM+1;

var Fire_ac {u in 0..n, c in comb[u], i in ac, t in otime[u] :si[i]=cu and cu in
parts[u]} =
  (sum {o in base[i] inter oar[u], oo in oar[u]} (L[u]-
2*r[o,oo]/max_speed[i,terr[oo]])*acfrac[u,i,o,t,str[si[i],c],oo]*calc[u,c,t,'force',o,
i,0,0,0]*sum {a in ta[i], j in ty, b in ta[j] :si[j] in parts[u]}
eng[u,i,j,terr[oo],a,b]*dam[si[i],si[j]])/(card[ta[i], j in ty, ta[j] :si[j] in
parts[u]}*sum {x in parts[u]} dam[si[i],x])
  -L[u]*frt[i];

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 10;
let {i in QD..QM} Pen[i] := 5e-3;
let QD:=QM+1;

var Aircraft_control {u in 0..n, c in comb[u], o in oar[u] inter (union {i in ac
:si[i]=cu} base[i]), t in otime[u] : info[cu]=1 and cu in parts[u]} =
  sum {oo in oar[u]} (1.2*((10*eps+sum {i in ac inter tyo[u,c,o] :si[i]=cu}
acfrac[u,i,o,t,str[si[i],c],oo]*calc[u,c,t,'force',o,i,0,0,0])^0.1-(10*eps)^0.1)
  + sum {i in ac inter tyo[u,c,o] :si[i]=cu}
acfrac[u,i,o,t,str[si[i],c],oo]*calc[u,c,t,'force',o,i,0,0,0])
  -sum {i in c2 :si[i]=cu} c2_cap[i]*calc[u,c,t,'force',o,i,0,0,0];

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 2e-1;
let {i in QD..QM} Pen[i] := 0.1;
let QD:=QM+1;

var Min_prob {u in 0..n :cu in parts[u]} = (if card(cum[u,1])=1 then (if 1 in cum[u,1]
then 0.2) else min_prob[u,1])-pl[u,cu];

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 1e-2;
let {i in QD..QM} Pen[i] := 1;
let QD:=QM+1;

```

A.2.3 Kjørefil

Filer med filtype `.run` inneholder kommandoer i AMPL som laster inn modellfilen, datafilen og beskrankningsfilen, starter løsningsalgoritmen og skriver resultatene ut på filer. I denne filen setter man også opsjoner som påvirker hvordan løsningsalgoritmen arbeider.

```
# MSO2.1 : Kjørefil

# Nullstill

reset;
reset options;

option omit_zero_rows 0;
option display_eps 1e-5;

option show_stats 0;
option times 0;

option linelim 0;
option substout 0;
option pl_linearize 0;
option presolve 0;
option solver minos;

param QC;
param QD;
param QM;
param FeasTol {QC+1..QM};
param Pen {i in QC+1..QM};

# Innstillinger

option bane ''; # Angir katalogen hvor resultatfilene skal
                # skrives, '' innebærer gjeldende katalog
param file symbolic default 'hotdog.tmp'; # Angir navnet på filen hvor HotDog
param output default 2; # Angir mengden av informasjon som ønskes om
                        # søkeprosessen; 0 er ingenting, 1 er sammendrag, 2 er sammendrag av
                        # hvert lokalsøk, 3 er utviklingen i hvert enkelt lokalsøk
param StartDef default 1; # Angir hvorvidt første lokalsøk skal startes
                        # fra variablenes default-verdier
param OptTol default 0.03; # Angir hvor liten gradientnormen skal være
                        # for at et punkt blir godtatt som lokalt minimum
param rep default 10; # Angir antallet lokalsøk som skal påbegynnes
param max_red default 19; # Angir maksimalt antall ganger søkealgoritmen
                        # skal redusere steplengdene
param max_iters default 300; # Angir maksimalt antall iterasjoner som skal
                        # utføres på et lokalsøk
param max_step default 0.1; # Angir den maksimale endringen i en variabel
                        # i en iterasjon. For eksempel betyr 0.1 at hver variabel kan endres
                        # maksimalt en tiendedel av differansen mellom øvre og nedre grenser
                        # for den enkelte variabelen.
param max_acc default 1e-16; # Angir den maksimale presisjon i beregninger
param Qr default 1; # Angir radius i "pølsene" som HotDog lager,
                    # relativt til standardverdien, som er  $0.05\sqrt{N}$ , hvor N er
                    # antallet variabler. Denne parameteren har ingen funksjon når Micio
                    # brukes alene, uten HotDog

# Problem

param LOOP default 1;
param cu default 1;
param cu_str default 0;
param s:=2;
param vars;

include MSO.mod # Last inn modell-filen listet over
data MSO.dat # Last inn datafil for scenari(ene)
include LOOP.tmp # Last inn data fra forrige iterasjon for MaxiMin algoritme
```



```

if cu=2 then let {u in 0..n} cum[u,2]:={cu_str};;
check s=2 or lino=0;
let rep      := if cu=1 then 15 else 7;
fix start;
fix task1;
fix dfrac;
fix acfrac;
#fix p2;
unfix {(u,i,o,t,æ) in Roma, a in ta[i] diff {last(ta[i])} :si[i]=cu and
(stype[u,si[i],æ]=0 or stype[u,si[i],æ]=1)} task1[u,i,o,t,æ,a];
unfix {(u,i,o,oo,t,æ) in Flytt :si[i]=cu and (stype[u,si[i],æ]=0 or
stype[u,si[i],æ]=1)} dfrac[u,i,o,oo,t,æ];
unfix {(u,i,o,t,æ) in Roma, oo in oar[u] :i in ac and si[i]=cu and (stype[u,si[i],æ]=0
or stype[u,si[i],æ]=1)} acfrac[u,i,o,t,æ,oo];
unfix {u in 0..n,i in ty,o in (if si[i]=def[u] then oar[u] union iar[u,si[i]] else
iar[u,si[i]]) inter (if i in ac then base[i] else ar), st in cum[u,si[i]] :si[i] in
parts[u] inter {cu} and (stype[u,si[i],st]=0 or stype[u,si[i],st]=1)} start[u,i,o,st];

let {u in 0..n, i in ty, st in cum[u,si[i]], o in iar[u,si[i]] :budsj[si[i]]=0 and
def[u]!=si[i] and card(iar[u,si[i]])=1} start[u,i,o,st] := arv[u,i];
fix {u in 0..n, i in ty, st in cum[u,si[i]], o in iar[u,si[i]] :budsj[si[i]]=0 and
def[u]!=si[i] and card(iar[u,si[i]])=1} start[u,i,o,st];

let QC:=_nvars;
let QD:=QC+1;

include MSO.con      # Last inn beskrankningsfilen listet over

if lino=0 then let {i in QC+1..QM} Pen[i]:=2*Pen[i];

# Søk

if LOOP=1 then {let StartDef:=0; let rep:=1+(cu-1)*3};

let file := ($bane & 'zHot' & substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) &
substr('0123456789',LOOP-10*floor(LLOOP/10)+1,1) & '-' & cu & '-' & cu_str & '.tmp');

include micio.txt    # Last inn og start HOTDOG/MICIO løsningsalgoritme

close;

### CHAPTER 10: SETS RELATED TO DISPLAYING RESULTS ###

set exprs;
set printvars within calcvars default {};

data;

set exprs :=
"Artilleriets nedholdende effekt på manøverstyrkene:"
"({(u,j,b,o,t,c) in Tap :j in man} (high[j]-(high[j]-low[j])*((1-high[j]+eps)/(low[j]-
high[j]+eps))^(calc[u,c,t,'lossf',o,j,b,0,0]+calc[u,c,t,'lossac',o,j,b,0,0])/(L[u]*no
rm[j]*calc[u,c,t,'pol',o,j,b,0,0]+eps))))"
"Informasjonens effektivitetsfaktor (manøverstyrkene):"
"{u in 0..n, x in parts[u], c in comb[u], o in oar[u], t in otime[u]} if info[x]=0
then 1 else (max_eff[u]-(max_eff[u]-min_eff[u])*exp(-
rate_eff[u]*sqrt(eps+calc[u,c,t,'C2',o,0,0,x,0]*calc[u,c,t,'ISR',o,0,0,x,0])))"
"Fordeling av komponentene på oppdrag:"
"{u in 0..n, c in comb[u], i in ty, a in ta[i] :si[i] in parts[u]} (sum {o in
be[u,i,str[si[i],c]}, t in time[u]}
calc[u,c,t,'force',o,i,0,0,0]*task[u,i,o,t,str[si[i],c],a])/(eps+sum {o in
be[u,i,str[si[i],c]}, t in time[u]} calc[u,c,t,'force',o,i,0,0,0])"
"Fordeling av tap på komponenttyper:"
"{u in 0..n,c in comb[u],i in ty,loss in {'man','fir','ac','aac'} :si[i] in parts[u]
and (loss!='aac' or i in ac)} sum {(u,i,a,o,t,c) in Tap :{o in oar[u] or i in ac}}
((if loss='man' and o in oar[u] then calc[u,c,t,'lossm',o,i,a,0,0])+(if loss='fir'
and o in oar[u] then calc[u,c,t,'lossf',o,i,a,0,0])+(if loss='ac' and o in oar[u] then
calc[u,c,t,'lossac',o,i,a,0,0])+(if loss='aac' then
calc[u,c,t,'lossaac',o,i,a,0,0]))*min(1,calc[u,c,t,'pol',o,i,a,0,0]*(1-exp(1-
abs(1+1.00587* ((sum {{1} :o in oar[u]}
(calc[u,c,t,'lossm',o,i,a,0,0]+calc[u,c,t,'lossf',o,i,a,0,0]+calc[u,c,t,'lossac',o,i,a
,0,0])+sum {{1} :i in ac}
calc[u,c,t,'lossaac',o,i,a,0,0])/(calc[u,c,t,'pol',o,i,a,0,0]+3*eps))+0.432961*((sum
{{1} :o in oar[u]}

```

```

(calc[u,c,t,'lossm',o,i,a,0,0]+calc[u,c,t,'lossf',o,i,a,0,0]+calc[u,c,t,'lossac',o,i,a
,0,0])+sum {{1} :i in ac}
calc[u,c,t,'lossaac',o,i,a,0,0]/(calc[u,c,t,'pol',o,i,a,0,0]+3*eps))^2+2.93395* ((sum
{{1} :o in oar[u]}
(calc[u,c,t,'lossm',o,i,a,0,0]+calc[u,c,t,'lossf',o,i,a,0,0]+calc[u,c,t,'lossac',o,i,a
,0,0])+sum {{1} :i in ac}
calc[u,c,t,'lossaac',o,i,a,0,0]/(calc[u,c,t,'pol',o,i,a,0,0]+3*eps))^5))/ (eps+(if o
in oar[u] then
calc[u,c,t,'lossm',o,i,a,0,0]+calc[u,c,t,'lossf',o,i,a,0,0]+calc[u,c,t,'lossac',o,i,a
,0,0])+(if i in ac then calc[u,c,t,'lossaac',o,i,a,0,0])))"
"Totale styrker og tap på hver side:"
"{u in 0..n, c in comb[u], x in parts[u], o in oar[u], t in otime[u]} (scale*sum {i in
ty :si[i]=x and i in tyo[u,c,o]} (calc[u,c,t-
1,'force',o,i,0,0,0]+calc[u,c,t,'din',o,i,0,0,0])*w[u,i], scale*sum {i in ty :si[i]=x
and i in tyo[u,c,o]} (calc[u,c,t-1,'force',o,i,0,0,0]+calc[u,c,t,'din',o,i,0,0,0]-
calc[u,c,t,'disp',o,i,0,0,0])*w[u,i])"
"Flyangrep; flypakker som angriper og normerte styrker som tapes:"
"{u in 0..n, c in comb[u], x in parts[u], o in oar[u], t in otime[u]} (sum {i in ac,
oo in be[u,i,str[si[i],c]} :si[i]=x} (calc[u,c,t-
1,'force',oo,i,0,0,0]+calc[u,c,t,'din',oo,i,0,0,0])*acfrac[u,i,oo,t,str[si[i],c],o],
scale*sum {i in ty diff ac, a in ta[i]} :si[i]!=x and i in tyo[u,c,o]}
calc[u,c,t,'lossac',o,i,a,0,0]*w[u,i])"
;

set printvars := disp;

model;

if cu_str=0 or cu_str=m[0,cu] then {

display LOOP,cu > ($bane & 'zVar' &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp');
display LOOP,cu > ($bane & 'zInf' &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp');
display LOOP,cu > ($bane & 'zExp' &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp');

print {i in QC+1..QM : _var[i] > FeasTol[i]/2 } : _var[i]/FeasTol[i],_varname[i],
_var[i] >> ($bane & 'zInf' & substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) &
substr('0123456789',LOOP-10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp');

};

if cu_str>0 then {
let vars:=_snvars;
print {i in 1..vars}: "let", _svarname[i], ":", _svar[i], ";" >> ($bane & 'zSol' &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp');
};

if cu_str=m[0,cu] then {
let {u in 0..n} cum[u,cu]:=1..m[u,cu];
close;
print 'in' & 'clude ' & $bane & 'zSol' &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp' > LOOP.tmp;
close;
};

if cu_str=m[0,cu] then {
include LOOP.tmp
};

if cu_str=0 or cu_str=m[0,cu] then {

print {i in _VARS diff {'calc'}} ('display ' & i & ', ' & i & '.rc, ' & i & '.dual' &
" >> " & $bane & "zVar" & substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) &
substr('0123456789',LOOP-10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp;') > display.tmp;
print {i in exprs} ('display ' & i & " > " & $bane & "zExp" &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp;') > display.tmp;
print {i in printvars} ('display {(u,c,t,"' & i & "',o,i,a,x,oo) in calcindx}
calc[u,c,t,"' & i & "',o,i,a,x,oo] >> ' & $bane & "zVar" &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(LOOP/10)+1,1) & '-' & cu & '.tmp;') > display.tmp;

} else print ' ' > display.tmp;

```

```

close;

include display.tmp

close;

if cu_str=0 then {
unfix task1;
unfix dfrac;
unfix acfrac;
unfix start;
unfix p2;
let vars:=_snvars;
print {i in 1..vars}: "let", _svarname[i], ":", _svar[i], ";" >> ($bane & 'zSol' &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(L00P/10)+1,1) & '-' & cu & '.tmp');
};

if cu=1 then shell ('copy zSol' & substr('0123456789x',min(ceil(0.001+LOOP/10),11),1)
& substr('0123456789',LOOP-10*floor(L00P/10)+1,1) & '-1.tmp zSol' &
substr('0123456789x',min(ceil(0.001+LOOP/10),11),1) & substr('0123456789',LOOP-
10*floor(L00P/10)+1,1) & '-2.tmp');

param Minos {u in 0..n, com[u]};
let {u in 0..n, c in com[u]} Minos[u,c] := M[u,c,cu];
var prob {u in 0..n,1..m[u,cu]} >=0 <=1;
var Max {0..n};
Bigger {u in 0..n, c in setof {c in com[u]} (c-str[cu,c]*10^(cu-1))} : sum {st in
1..m[u,cu]} prob[u,st]*Minos[u,c+st*10^(cu-1)] >= Max[u];
Total_prob {u in 0..n}: sum {st in 1..m[u,cu]} prob[u,st]=1;
minimize V: -(sum {u in 0..n :cu in parts[u]} weight[u]*Max[u])/sum {u in 0..n :cu in
parts[u]} weight[u];

if cu_str=0 or cu_str=m[0,cu] then {

objective V;
solve;

option display_lcol 3;
display LOOP,cu,V,f,Obj,{u in 0..n,st in 1..m[u,cu],c in (setof {c in com[u]} (c-
str[cu,c]*10^(cu-1))) union {'prb'}} if c='prb' then prob[u,st] else M[u,c+st*10^(cu-
1),cu] >> ($bane & 'zSummary.tmp');
print {i in now}: i,max(0,max {u in 0..n, st in 1..m[u,si[i]} : si[i] in parts[u]}
(sum {o in (if def[u]=si[i] then oar[u] union iar[u,si[i]] else iar[u,si[i]]) inter if
i in ac then base[i] else ar} start[u,i,o,st]-arv[u,i]),max(0,max {u in 0..n, st in
1..m[u,si[i]} : si[i] in parts[u]} (sum {o in (if def[u]=si[i] then oar[u] union
iar[u,si[i]] else iar[u,si[i]]) inter if i in ac then base[i] else ar}
start[u,i,o,st]-arv[u,i]))*cost[i] >> ($bane & 'zSummary.tmp');
print "" >> ($bane & 'zSummary.tmp');

};

# Lagre data på fil LOOP.tmp som input til neste iterasjon for MaxiMin algoritme

print 'let LOOP :=' & LOOP+(if cu=s and cu_str=m[0,s] then 1) & '; let cu :=' & (if
cu<s then cu+1 else if cu_str=m[0,cu] then 1 else cu) & '; let cu_str:= ' & (if cu=1
then 1 else if cu_str=m[0,2] then 0 else cu_str+1) & ';' > LOOP.tmp;
print 'in' & 'clude ' & $bane & 'zSol' & (if 0<cu_str<m[0,cu] and cu=1 then
substr('0123456789x',min(ceil(L00P/10-0.999),11),1) & substr('0123456789',L00P-
10*floor(L00P/10-0.1),1) else substr('0123456789x',min(ceil(0.001+L00P/10),11),1) &
substr('0123456789',L00P-10*floor(L00P/10)+1,1)) & '-' & (if cu_str=0 or
cu_str=m[0,cu] then cu else if cu=1 then s else cu-1) & '.tmp' >> LOOP.tmp;
print {u in 0..n, c in com[u],x in 1..s} "let M2[" & u & "," & c & "," & x & "]:=" &
M[u,c,x] & ";" >> LOOP.tmp;
close;

```

A.3 HOTDOG/MICIO løsningsverktøy

Interaktiv versjon og den frittstående kommando-linje versjonen av AMPL (37) er begge lisensiert med løsningsverktøyet *MINOS5.5* for både LP og NLP problemer. Dette verktøyet aktiveres ved en opsjon '*option solver minos*' som input til AMPL i hver kjørefil.

Det er også utviklet en egen heuristisk løsningsalgoritme kalt *HOTDOG* i AMPL som også kan benyttes. Denne kan aktiveres ved å inkludere en filpeker til *HOTDOG* ('*include hotdog.txt;*') i

kjørefilen istedet for opsjonen til MINOS5.5. En forenklet versjon av HOTDOG, kalt MICIO, er også utviklet som ikke lagrer søkeveier under optimering, og som derfor er hurtigere når det antas at antallet lokale optima kan være stort. Denne versjonen er brukt i eksemplet i Appendiks A.4.

A.3.1 Algoritme for HOTDOG/MICIO

Løsningsverktøyet HOTDOG ("Heuristisk Oppskrift Til å Determinere Optimum Globalt" / "Heuristic On The Determination of Optimum Globally") er en algoritme som er utviklet for å effektivisere søket etter det globale minimum i problemer med et rimelig antall lokale minimalpunkter

For å finne det absolutte minimum til en funksjon, er det en vanlig fremgangsmåte å foreta flere søk etter lokale minima fra forskjellige startpunkter for til slutt å velge det lokale minimumet med lavest verdi. Startpunktene blir som regel generert mer eller mindre tilfeldig. Det er imidlertid lite gunstig å søke etter lokalpunkter i samme område flere ganger. Derfor er det utviklet diverse metoder for å velge ut et delsett av punkter hvorfra det skal igangsettes søk, fra et sett av genererte punkter. Hovedideen i disse metodene er at det ikke skal igangsettes søk fra punkter som ligger nær hverandre. Noen av metodene bruker dessuten kriterier knyttet til for eksempel funksjonsverdiene til startpunktene eller gradientretningene i dem (35). HOTDOG prøver å oppnå en enda kraftigere ytelsesforbedring ved også å avbryte noen av søkene etter at de er igangsatt. Hvis antallet lokale optimalpunkter er lite i forhold til antallet lokale søk som igangsettes, kan man på denne måten oppnå en betydelig besparelse i regnetid. Hovedidéen er å avbryte de lokale søkene så fort som mulig hvis de er på vei mot et lokalpunkt man allerede har oppdaget. Det gjøres i HOTDOG ved at man representerer den omtrentlige ruten hvert enkelt lokalsøk følger, med en rekke sammenhengende linjestykker. Dermed kan man underveis i et lokalt søk teste om man er i nærheten av ruten til et tidligere lokalt søk. Hvis det er tilfellet, vil trolig det lokale søket man holder på med, ende i samme lokale optimalpunkt. Derfor blir søket avbrutt. Hvis man visualiserer problemet som en høy-dimensjonal topografi med "bakkar og berg", vil trolig de lokale søkene raskt komme seg ned i "dalbunnene". Siden det området hvor søkene blir avbrutt, har en viss utstrekning, blir området et nettverk av "pølser", derav navnet på algoritmen. Konsentrasjonen av disse blir størst i dalbunnene, hvor de fleste søkene raskt vil havne og bli avbrutt.

Når antallet lokale optimalpunkter er stort i forhold til antallet lokale søk som ønskes utført, har det liten hensikt å bruke mer raffinerte metoder enn å la samtlige av de lokale søkene fullføres. HOTDOG vil nemlig da kunne avbryte bare en liten andel av søkene, og effektivitetsforbedringen blir marginal. I slike tilfeller hjelper det enda mindre å bruke de nevnte metodene som kun ser på startpunktene. Man kan da bare håpe at det beste lokalpunktet man finner, har en verdi som ligger nært opptil den globale løsningen. For disse tilfelle er algoritmen MICIO ("Minimum Computation by Iterative Optimization") også utviklet for lokalsøk spesielt velegnet på store problemer.

MICIO er en søkealgoritme for optimeringsproblemer og baserer seg hovedsakelig på gradienter. Ved hjelp av disse beregnes det dessuten en vektor som er relatert til den andrederiverte, og som kan gi noen av de samme gevinstene som ved å bruke den fullstendige Hessiske matrisen, men uten at man må ofre store mengder beregningstid (36). Algoritmen er

derfor spesielt velegnet på store, ulineære problemer hvor en ren gradientmetode ikke fører frem og det er for kostbart å beregne andrederivert-matrisen.

Algoritmen opererer på en funksjon hvor hver av variablene kan ta alle verdiene i et fast intervall. Beskrankninger gjøres automatisk om til et ekstra straffeledt i funksjonen som skal optimeres (se mal for modellutvikling i kapittel A.3.2). Den opprinnelige funksjonen må derfor defineres slik at den returnerer et reelt tall uansett hvilken kombinasjon av verdier variablene innehar, selv om beskrankningene er brutt. Beskrankningene skal være formulert som uttrykk som skal være ikke-positive når beskrankningene er tilfredsstilt. Det vil si at en beskrankning av typen $f(x) \leq g(x)$ blir til uttrykket $f(x) - g(x)$, og $f(x) = g(x)$ blir til $|f(x) - g(x)|$. Dessuten skal man for hver beskrankning fastsette en toleransegrense Tol og en størrelse Pen på straffeledet. Straffeledet er en kombinasjon av en kvadratisk funksjon innenfor toleransegrensen og en lineær funksjon utenfor. Hvis uttrykket er $v(x)$, blir straffeledet seende slik ut:

$$\frac{Pen}{2 * Tol} * \min(Tol, \max(0, v(x)))^2 + Pen * \max(0, v(x) - Tol)$$

Den deriverte av denne funksjonen med hensyn på x er kontinuerlig over alt såfremt $f(x) - g(x)$ er det. Det gjør at funksjonen er lett å arbeide med for søkealgoritmen. Hensikten med å formulere straffeledet på denne måten er at man skal oppnå like rask progresjon mot det tilfredsstilte området som med en ren lineær funksjon, samtidig som man unngår de problemene en lineær funksjon skaper når beskrankningen nesten er tilfredsstilt. Pen bør settes til en verdi som er stor nok til at objektivverdien forringes når beskrankningen brytes.

Selve det lokale søket i HOTDOG/MICIO kan grovt sammenfattes slik:

1. Sett $dx := 0$, trekk random punkt := x
2. Sett $f1 := F(x)$, sett $point1 := x$, sett $rc2 := rc$, sett $rc := \nabla F(x)$
3. Hvis $|rc| \leq OptTol$, stopp
4. Sett $dx :=$ hvis $dx = 0$ så (hvis $rc < 0$ så $0,03$ ellers hvis $rc > 0$ så $-0,03$)
ellers $\frac{rc \times dx}{rc2 - rc}$ (= min. av lokal 2.ordens kurve)

(uttrykket må dessuten modifiseres for å hindre at variablene kommer utenfor grensene, at endringene blir for store og at endringen blir feil vei, se listing nedenfor)

5. Sett $x := point1 + dx$
6. Hvis $F(x) < f1$, gå til punkt 2
7. La $dx := dx * (\text{hvis } dx * \nabla F(x) \leq 0 \text{ så } 1 \text{ ellers } \frac{rc}{rc - \nabla F(x)})$
8. Gå til punkt 5

Hvis ønskelig, foretas deretter et ordinært gradientretning-søk for om mulig å finpusse resultatet.

Likheter er generelt noe vanskeligere å håndtere enn ulikheter da likhetene alltid er aktive. Likheter bør derfor i størst mulig grad brukes til å eliminere variabler ved å benytte definerte variable i AMPL (38) som beskrevet foran i kapittel 2.2.4.5. Dermed reduserer man både antallet variabler og antallet beskrankninger. De likhetene man velger å beholde, bør gis størst mulig toleransegrense.

Størrelsen på straffeledene Pen kan man i første omgang fastsette etter skjønn ut i fra at

straffen for å bryte en beskrankning bør være en del større enn den reduksjonen som derved kan oppnås i funksjonen som skal minimeres. På den måten vil gradientene som regel peke mot det mulige området. Likevel kan man selvsagt risikere at noen av lokalpunktene algoritmen stopper ved, er umulige. Hvis det bare gjelder et mindretall, kan man klare seg bra ved bare å eliminere de umulige lokalpunktene og velge det beste av de mulige. Hvis for mange av lokalpunktene er umulige, må man øke straffeleddene på de beskrankningene som blir brutt. Visse typer beskrankninger kan skape problemer for søkealgoritmen hvis straffeleddene er store. Man bør derfor ikke sette overdrevent store straffeledd.

HOTDOG i kombinasjon med MICIO er realisert i AMPL (38). Den er ikke en egen prosess, men opererer innenfor kommandospråket til AMPL og benytter dennes innebygde støtte for gradientberegninger av en funksjon f ($=f.rc$) basert på den effektive og elegante metoden med automatisk differensiering i AMPL (36)(38). Det gjør det lett å tilpasse algoritmen til egne behov. Algoritmen trekker et antall tilfeldige punkter innenfor de øvre og nedre grensene ($.ub$ og $.lb$) og starter lokale søk fra dem med MICIO. HOTDOG lagrer rutene som MICIO følger og sørger for å avbryte søket når det kommer i nærheten av en gammel rute.

Dersom man har mindre enn 300 variabler i problemet, klarer man seg med studentversjonen av AMPL. Ellers kan man faktisk gjøre det samme hvis man enten lager et analytisk uttrykk for den deriverte av objektivfunksjonen eller selv beregner en tilnærmet gradient. Da slipper man å kalle AMPLs interne gradientfunksjoner $.rc$. Dermed kan man unnsnippe "presolve" i AMPL siden man ikke sender problemet til en ekstern solver, og da slipper man samtidig kontrollen av antallet variabler.

Listing av HOTDOG :

```
# Problem

if QM=QC then print "param feas := 1; minimize Obj: f;" > objective.tmp;
if QM>QC then print "var feas = if 1", {i in QC+1..QM} ("and max(0,",
_varname[i], "-FeasTol[" , i, "]]=0)", "then 1 else 0; minimize Obj: f", {i in
QC+1..QM} ("+Pen[" , i, "]/(2*FeasTol[" , i, "])*min(FeasTol[" , i, "],max(0,",
_varname[i], "))^2 + Pen[" , i, "]*max(0,", _varname[i], "-FeasTol[" , i,
"])]", ";") > objective.tmp;

close;

include objective.tmp

param QN;
let QN:=_snvars;

# Opsjoner

option randseed '';
option display_lcol 11;
option display_width 119;
option display_eps 1e-6;
option display_precision 5;
option print_precision 4;
option print_separator " " ;
```

```

print ' ' > options;

if {'file'} not within _PARS then print "param file symbolic default
'HotDog.tmp';" >> options;
if {'output'} not within _PARS then print "param output default 2;" >>
options;
if {'StartDef'} not within _PARS then print "param StartDef default 1;" >>
options;
if {'OptTol'} not within _PARS then print "param OptTol default 0.03;" >>
options;
if {'rep'} not within _PARS then print "param rep default 10;" >> options;
if {'max_red'} not within _PARS then print "param max_red integer default
13;" >> options;
if {'max_iters'} not within _PARS then print "param max_iters default 250;"
>> options;
if {'max_step'} not within _PARS then print "param max_step default 0.1;" >>
options;
if {'max_acc'} not within _PARS then print "param max_acc default 1e-16;" >>
options;
if {'Qr'} not within _PARS then print "param Qr default 1;" >> options;

include options

# Søk

let Qr:=Qr*sqrt(QN)*0.05;

param Qc default 0;
param Objs {1..Qc};
param Xs {1..Qc, 1..QN};
param best_rep;
param Qcancel {1..Qc} default 0;
param Qf;
param Qfinish binary default 0;
param Qiters binary default 0;
param Qstop binary default 0;
param Qhotdog binary;

param Qline {1..Qc} default 0;
param Qpoint {i in 1..Qc, 0..Qline[i], 1..QN};
param Qpoint1 {1..QN} default 0;
param Qdist {i in 1..Qc, 1..Qline[i]};
param Qorig {i in 1..Qc, 0..Qline[i]};
param Qorigo;
param Qmove1 default 0;
param Qtest;
param Qdistance;
param Qminor default 0;
param Qnumber default 0;
param Qtotal default 0;
param Grad_evals default 0;

param Qrc {1..QN} default 0;
param Qrc2 {1..QN};
param Qdx {1..QN};
param Qgrad;
param Qn_red;

param Qa {i in 1..QN};
for {i in 1..QN} {
let Qa[i] := _svar[i].ub-_svar[i].lb;
if Qa[i]<=0 then print "Invalid bounds on variable", _svarname[i], ". Upper
bound must be greater than lower bound.";

```

```

);

if output>=1 then print "Variabler:",QN > (file);
if output>=1 then print "Beskrankninger:",QM-QC >> (file);
if output>=1 then print "" >> (file);
if output>=2 then print
"Draw", "Iters", "Total", "Grad", "Feas", "Penalty", "Function", "End" >> (file);

# Random draws for global search : finish if Qc > rep
repeat {
let Qc:=Qc+1;
if Qc>1 or StartDef=0 then let {i in 1..QN}
_svar[i]:=Uniform(_svar[i].lb,_svar[i].ub);
let {i in 1..QN} Qpoint[Qc,0,i]:=_svar[i]/Qa[i];
let {i in 1..QN} Qpoint1[i]:=Qpoint[Qc,0,i];
let Qorig[Qc,0]:=sqrt(sum {i in 1..QN} ((_svar[i]-_svar[i].lb)/Qa[i])^2);
let Qhotdog:=0;
let Qdistance:=0;
let Qtest:=Qr;
let Qiters:=0;
let Qfinish:=0;
let Qstop:=0;
let Qnumber:=0;
let Qminor:=0;
let {i in 1..QN} Qdx[i]:=0;
let {i in 1..QN} Qrc[i]:=0;
if output>=3 then print "" >> (file);
if output>=3 then print "", "Iters", "Dist", "Grad" >> (file);

# Major iterations : terminate if 1) within "Hotdog" tolerance, 2) Qgrad <=
OptTol or 3) max_iters reached
repeat {
repeat {

if Qtest >= Qr/2 then {
let Qtest:=0;
let Qorigo := sqrt(sum {i in 1..QN} ((_svar[i]-_svar[i].lb)/Qa[i])^2);
if min {Qct in 1..Qc-1, l in 1..Qline[Qct] :Qorigo+Qr>=min(Qorig[Qct,l-
1],Qorig[Qct,l]) and Qorigo-Qr<=max(Qorig[Qct,l-1],Qorig[Qct,l]) and
Obj>=Obj[Qct]} {
if sum {i in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l,i])^2 >= Qdist[Qct,l]+sum {i
in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l-1,i])^2 then sum {i in 1..QN}
(_svar[i]/Qa[i]-Qpoint[Qct,l-1,i])^2
else if sum {i in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l-1,i])^2 >=
Qdist[Qct,l]+sum {i in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l,i])^2 then sum {i
in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l,i])^2
else 0.25*(2*sum {i in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l-1,i])^2+2*sum {i
in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l,i])^2-Qdist[Qct,l]-((sum {i in 1..QN}
(_svar[i]/Qa[i]-Qpoint[Qct,l-1,i])^2)^2+(sum {i in 1..QN} (_svar[i]/Qa[i]-
Qpoint[Qct,l,i])^2)^2-2*(sum {i in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l-
1,i])^2)*(sum {i in 1..QN} (_svar[i]/Qa[i]-Qpoint[Qct,l,i])^2))/Qdist[Qct,l])
) <= Qr^2 then {let Qhotdog:=1; let Qcancel[Qc]:=1; let Qfinish:=1; break;};
};

let Qf:=Obj;
let {i in 1..QN} Qpoint1[i] := _svar[i];
let {i in 1..QN} Qrc2[i]:= Qrc[i];
let {i in 1..QN} Qrc[i] := _svar[i].rc*Qa[i];
let Grad_evals:=Grad_evals+1;

let Qgrad := sqrt(sum {i in 1..QN :if Qrc[i]>0 then (if _svar[i]>_svar[i].lb
then 1) else (if _svar[i]<_svar[i].ub then 1)} Qrc[i]^2);
if Qgrad <= OptTol then {let Qfinish:=1; break;};

if Qnumber >= max_iters then {let Qiters:=1; let Qfinish:=1; break;};

```



```

let Qnumber:=Qnumber+1;

# Step based upon local 2.order approximation
let {i in 1..QN} Qdx[i] := if Qrc[i]=0 then 0 else if abs(Qdx[i])<=1e-8 then
((if Qrc[i]>0 then (_svar[i].lb-_svar[i])*max_step/10) + if Qrc[i]<0 then
(_svar[i].ub-_svar[i])*max_step/10)/Qa[i] else (max(min(_svar[i]+Qa[i]*(if
Qrc[i]<0 and Qdx[i]>0 and Qrc2[i]>=(1+Qdx[i]/max_step)*Qrc[i] then max_step
else if Qrc[i]>0 and Qdx[i]<0 and Qrc2[i]<=(1-Qdx[i]/max_step)*Qrc[i] then -
max_step else Qrc[i]*Qdx[i]/(Qrc2[i]-Qrc[i])),_svar[i].ub),_svar[i].lb)-
_svar[i])/Qa[i];
let Qn_red:=0;
let Qstop:=1;

# Minor iterations - stop : 1) diff < max_acc or 2) max_red reached
repeat {
let {i in 1..QN} _svar[i] := Qpoint1[i] + Qa[i]*Qdx[i];
if 0<Qf-Obj then {let Qstop:=0; break;};
if Qn_red>=max_red then break;
let Qn_red:=Qn_red+1;
if Qn_red=round(Qn_red/3)*3 then {let {i in 1..QN} Qdx[i] := Qdx[i]/2;} else
{
let {i in 1..QN} Qrc2[i]:=_svar[i].rc*Qa[i];
let {i in 1..QN} Qdx[i] := Qdx[i]*if Qdx[i]*Qrc2[i]<=max_acc then 1 else
max(0.1,Qrc[i]/(Qrc[i]-Qrc2[i]));
let Grad_evals:=Grad_evals+1;};
};

if Qstop=1 then {let {i in 1..QN} _svar[i] := Qpoint1[i]; let Qfinish:=1;
break;};

let Qmove1 := sqrt(sum {i in 1..QN} Qdx[i]^2);
let Qtest := Qtest + Qmove1;
let Qdistance := Qdistance + Qmove1;
if Qdistance >= 2*Qr then {let Qdistance:=0; break;};

};

if Qline[Qc]=0 and (Qnumber>1 or Qhotdog=0) then let Qline[Qc]:=1;
if Qline[Qc]>0 then {
let {i in 1..QN} Qpoint[Qc,Qline[Qc],i]:=_svar[i]/Qa[i];
let Qorig[Qc,Qline[Qc]]:=sqrt(sum {i in 1..QN} ((_svar[i]-
_svar[i].lb)/Qa[i])^2);
let Qdist[Qc,Qline[Qc]]:=sum {i in 1..QN} (Qpoint[Qc,Qline[Qc],i]-
Qpoint[Qc,Qline[Qc]-1,i])^2;
if output>=3 then print "",Qnumber-Qminor,sqrt(Qdist[Qc,Qline[Qc]]),Qgrad >>
(file);
};

let Qminor:=Qnumber;

if Qfinish=1 then {if feas=0 then let Qcancel[Qc]:=1; break;} else let
Qline[Qc]:=Qline[Qc]+1;

};

let Qtotal:=Qtotal+Qnumber;
let {i in 1..QN} Xs[Qc,i] := _svar[i];
let Objs[Qc]:=Obj;
if feas=0 then let Qcancel[Qc]:=1;

if output>=3 then print "", "....." >> (file);
if output>=2 then print Qc,Qnumber,Qtotal,if Qgrad<0.01 then (if Qgrad>=0.001
then precision(Qgrad,3) else precision(Qgrad,2)) else Qgrad,feas,if Obj-
f<0.01 then (if Obj-f>=0.001 then precision(Obj-f,3) else precision(Obj-f,2))
else Obj-f,f,"",if Qstop=1 then "Kan ikke forbedres" else if Qhotdog=0 then

```

```
(if Qiters=1 then "For mange iterasjoner" else "Lokalpunkt") else "Traff
pølse" >> (file);
#print Qc,Qline[Qc],sum {i in 1..Qc} Qline[i],feas,f,"",Obj,"",if Qstop=1
then "Kan ikke forbedres" else if Qhotdog=0 then (if Qiters=1 then "For mange
iterasjoner" else "Lokalpunkt") else "Traff pølse" >> (file);
if output>=3 then print
"_____ " >> (file);
```

```
include HTdisplay.tmp
```

```
if Qc>=rep then break;
};
```

```
let best_rep := min {i in 1..Qc :Objs[i]=min {j in 1..Qc} Objs[j]} i;
let {i in 1..QN} _svar[i]:=Xs[best_rep,i];
```

```
if output>=1 then {
print "" >> (file);
print "Gradientberegninger:",Grad_evals >> (file);
print "" >> (file);
print "Lokale maksimalpunkter:" >> (file);
if QN<=10 then display {j in 1..Qc, i in 1..QN :Qcancel[j]=0} Xs[j,i] >>
(file);
display {j in 1..Qc :Qcancel[j]=0} Objs[j] >> (file);
print "Globalt maksimalpunkt: (?)" >> (file);
display best_rep, f, Obj, feas >> (file); #{i in 1..QN} _svar[i],
display sum {i in 1..Qc} Qline[i], (sum {i in 1..Qc} Qline[i])/Qc >> (file);
};
```

Listing av MICIO (forenklet versjon av HOTDOG):

```
# Problem
```

```
if QM=QC then print "param feas := 1; minimize Obj: f;" > objective.tmp;
if QM>QC then print "var feas = if 1, {i in QC+1..QM} ("and max(0,",
_varname[i], "-FeasTol[" , i, "])=0)", "then 1 else 0; minimize Obj: f", {i in
QC+1..QM} ("+Pen[" , i, "]/(2*FeasTol[" , i, "])*min(FeasTol[" , i, "],max(0,",
_varname[i], "))^2 + Pen[" , i, "]*max(0,", _varname[i], "-FeasTol[" , i,
"])",",";" > objective.tmp;
```

```
close;
```

```
include objective.tmp
```

```
param QN;
let QN:=_snvars;
```

```
# Opsjoner
```

```
option randseed '';
option display_1col 11;
option display_width 119;
option display_eps 1e-6;
option display_precision 5;
option print_precision 4;
option print_separator "      ";
```

```
print ' ' > options;
```

```
if {'file'} not within _PARS then print "param file symbolic default
'HotDog.tmp';" >> options;
```

```

if {'output'} not within _PARS then print "param output default 2;" >>
options;
if {'StartDef'} not within _PARS then print "param StartDef default 1;" >>
options;
if {'OptTol'} not within _PARS then print "param OptTol default 0.03;" >>
options;
if {'rep'} not within _PARS then print "param rep default 10;" >> options;
if {'max_red'} not within _PARS then print "param max_red integer default
13;" >> options;
if {'max_iters'} not within _PARS then print "param max_iters default 250;"
>> options;
if {'max_step'} not within _PARS then print "param max_step default 0.1;" >>
options;
if {'max_acc'} not within _PARS then print "param max_acc default 1e-16;" >>
options;

```

```
include options
```

```
# Søk
```

```

param Qc default 0;
param Objs {1..Qc};
param Xs {1..Qc, 1..QN};
param best_rep;
param Qcancel {1..Qc} default 0;
param Qf;
param Qiters binary default 0;
param Qstop binary default 0;

```

```

param Qpoint1 {1..QN} default 0;
param Qnumber default 0;
param Qtotal default 0;
param Grad_evals default 0;

```

```

param Qrc {1..QN} default 0;
param Qrc2 {1..QN};
param Qdx {1..QN};
param Qgrad;
param Qn_red;

```

```

param Qa {i in 1..QN};
for {i in 1..QN} {
let Qa[i] := _svar[i].ub-_svar[i].lb;
if Qa[i]<=0 then print "Invalid bounds on variable", _svarname[i], ". Upper
bound must be greater than lower bound.";
};

```

```

if output>=1 then print "Variabler:",QN > (file);
if output>=1 then print "Beskrankninger:",QM-QC >> (file);
if output>=1 then print "" >> (file);
if output>=2 then print
"Draw", "Iters", "Total", "Grad", "Feas", "Penalty", "Function", "End" >> (file);

```

```

# Random draws for global search : finish if Qc > rep
repeat {
let Qc:=Qc+1;
if Qc>1 or StartDef=0 then let {i in 1..QN}
_svar[i]:=Uniform(_svar[i].lb,_svar[i].ub);
let {i in 1..QN} Qpoint1[i]:=_svar[i]/Qa[i];
let Qiters:=0;
let Qstop:=0;
let Qnumber:=0;

```

```

let {i in 1..QN} Qdx[i]:=0;
let {i in 1..QN} Qrc[i]:=0;

# Major iterations : terminate if 1) Qgrad <= OptTol or 2) max_iters reached
repeat {
let Qf:=Obj;
let {i in 1..QN} Qpoint1[i] := _svar[i];
let {i in 1..QN} Qrc2[i]:= Qrc[i];
let {i in 1..QN} Qrc[i] := _svar[i].rc*Qa[i];
let Grad_evals:=Grad_evals+1;

let Qgrad := sqrt(sum {i in 1..QN :if Qrc[i]>0 then (if _svar[i]>_svar[i].lb
then 1) else (if _svar[i]<_svar[i].ub then 1)} Qrc[i]^2);
if Qgrad <= OptTol then break;

if Qnumber >= max_iters then {let Qiters:=1; break;};
let Qnumber:=Qnumber+1;

# Step based upon local 2.order approximation
let {i in 1..QN} Qdx[i] := if Qrc[i]=0 then 0 else if Qdx[i]=0 or
abs(Qrc2[i]-Qrc[i])<1e-16 then ((if Qrc[i]>0 then (_svar[i].lb-
_svar[i])*max_step/10) + if Qrc[i]<0 then (_svar[i].ub-
_svar[i])*max_step/10)/Qa[i] else (max(min(_svar[i]+Qa[i]*(if Qrc[i]<0 and
Qdx[i]>0 and Qrc2[i]>=(1+Qdx[i]/max_step)*Qrc[i] then max_step else if
Qrc[i]>0 and Qdx[i]<0 and Qrc2[i]<=(1-Qdx[i]/max_step)*Qrc[i] then -max_step
else Qrc[i]*Qdx[i]/(Qrc2[i]-Qrc[i])),_svar[i].ub),_svar[i].lb)-
_svar[i])/Qa[i];
let Qn_red:=0;
let Qstop:=1;

# Minor iterations - stop : 1) diff < max_acc or 2) max_red reached
repeat {
let {i in 1..QN} _svar[i] := Qpoint1[i] + Qa[i]*Qdx[i];
if 0<Qf-Obj then {let Qstop:=0; break;};
if Qn_red>=max_red then break;
let Qn_red:=Qn_red+1;
if Qn_red=round(Qn_red/3)*3 then {let {i in 1..QN} Qdx[i] := Qdx[i]/2;} else
{
let {i in 1..QN} Qrc2[i]:=_svar[i].rc*Qa[i];
let {i in 1..QN} Qdx[i] := Qdx[i]*if Qdx[i]*Qrc2[i]<=max_acc then 1 else
max(0.1,Qrc[i]/(Qrc[i]-Qrc2[i]));
let Grad_evals:=Grad_evals+1;};
};

if Qstop=1 then {let {i in 1..QN} _svar[i] := Qpoint1[i]; break;};

};

let Qtotal:=Qtotal+Qnumber;
let {i in 1..QN} Xs[Qc,i] := _svar[i];
let Objs[Qc]:=Obj;
if feas=0 then let Qcancel[Qc]:=1;

if output>=2 then print Qc,Qnumber,Qtotal,if Qgrad<0.01 then (if Qgrad>=0.001
then precision(Qgrad,3) else precision(Qgrad,2)) else Qgrad,feas,if Obj-
f<0.01 then (if Obj-f>=0.001 then precision(Obj-f,3) else precision(Obj-f,2))
else Obj-f,f,"",if Qstop=1 then "Kan ikke forbedres" else if Qiters=1 then
"For mange iterasjoner" else "Lokalpunkt" >> (file);

include HTdisplay.tmp

if Qc>=rep then break;
};

let best_rep := min {i in 1..Qc :Objs[i]=min {j in 1..Qc} Objs[j]} i;

```

```

let {i in 1..QN} _svar[i]:=Xs[best_rep,i];

if output>=1 then {
print "" >> (file);
print "Gradientberegninger:",Grad_evals >> (file);
print "" >> (file);
print "Lokale maksimalpunkter:" >> (file);
if QN<=10 then display {j in 1..Qc, i in 1..QN :Qcancel[j]=0} Xs[j,i] >>
(file);
display {j in 1..Qc :Qcancel[j]=0} Objs[j] >> (file);
print "Globalt maksimalpunkt: (?)" >> (file);
display best_rep, f, Obj, feas >> (file); #{i in 1..QN} _svar[i],
};

```

A.3.2 HOTDOG mal for modellutvikling

For å benytte det heuristiske løsningsverktøyet HOTDOG/MICIO er det laget en mal for modellutvikling tilpasset denne løsningsalgoritmen. Denne malen gir en veiledning bl a til konstruksjon av beskrankninger.

Malen benytter AMPL's 'meta'-funksjoner der hver variabel gis et nummer (*j*) som kan refereres generisk ved en indeks (*_var[j]*) med korrespondanse til variabelnavnet (*_varname[j]*). for generell bruk Dette utnyttes av løsningsalgoritmen som på denne måten tilordner beskrankninger til variable for å lage straffe-ledd i optimeringen som beskrevet over. Dette gir mulighet for å separere beskrankningsvariable fra de deklarererte variable som skal optimeres i modellen og teste gyldighet av beskrankninger med variabelen *feas* separat (se listing av HOTDOG/MICIO over).

Hver beskrankning må gis en egen variabel som angitt i malen og AMPL genererer så de aktuelle instansvariablene på grunnlag av verdiene i datafilen.

```

# Nullstill

reset;
reset options;

option omit_zero_rows 0;
option display_eps 1e-5;

option show_stats 0;
option times 0;

option linelim 0;
option substout 0;
option pl_linearize 0;
option presolve 0;

# Deklarasjon av parametere HotDog trenger som input

param QC;
param QD;
param QM;
param FeasTol {QC+1..QM};      # Deklarasjon av beskrankningstoleranser
param Pen {i in QC+1..QM};     # Deklarasjon av beskrankningsledd straff

```

```

# Innstillinger av HotDog

option bane ''; # Angir katalogen hvor resultatfilene skal skrives, ''
                innebærer gjeldende katalog
param file symbolic default 'hotdog.tmp'; # Angir navnet på filen hvor
                HotDog fortløpende skriver informasjon om løsningsprosessen
param output default 2; # Angir mengden av informasjon som ønskes om
                søkeprosessen; 0 er ingenting, 1 er sammendrag, 2 er sammendrag
                av hvert lokalsøk, 3 er utviklingen i hvert enkelt lokalsøk
param StartDef default 1; # Angir hvorvidt første lokalsøk skal startes
                fra variablenes default-verdier
param OptTol default 0.03; # Angir hvor liten gradientnormen skal være for
                at et punkt blir godtatt som lokalt minimum
param rep default 10; # Angir antallet lokalsøk som skal påbegynnes
param max_red default 13; # Angir maksimalt antall ganger søkealgoritmen
                skal redusere steplengdene
param max_iters default 250; # Angir maksimalt antall iterasjoner som skal
                utføres på et lokalsøk
param max_step default 0.1; # Angir den maksimale endringen i en variabel i
                en iterasjon. For eksempel betyr 0.1 at hver variabel kan endres
                maksimalt en tiendedel av differansen mellom øvre og nedre
                grenser for den enkelte variabelen.
param max_acc default 1e-16; # Angir den maksimale presisjon i beregninger
param Qr default 1; # Angir radius i "pølsene" som HotDog lager,
                relativt til standardverdien, som er 0.05*sqrt(N), hvor N er
                antallet variabler. Denne parameteren har ingen funksjon når
                Micio brukes alene, uten HotDog

let file := ($bane & file); # Legger banereferansen til filnavnet

# Problem

# Her skal alle deklarasjoner av mengder, parametere og variabler i problemet
# være.
# Eventuelle datafiler må også lastes inn her. AMPL vil da generere den
# aktuelle modellen for videre tilpasning til HOTDOG.
#
# Dette kan ligge på hhv modell- og data-filer som kan inkluderes for å gi
# bedre oversikt over samlet kjøring, f eks slik :
#
# include <mymodel.mod>;
# data <mydatafile.dat>;

var f = <funksjon som skal minimeres> ;

let QC:=_nvars; # Antall variable generert hittil, dvs før beskrankninger.
let QD:=QC+1; # Benyttes for å kunne referere til beskrankningsvariable

# Her skal alle beskrankninger legges inn.
# Man skal ha en "let-blokk" mellom beskrankninger med ulike
# toleransegrenser eller penalty-parametere. Verdiene som legges inn
# i en "let-blokk", gjelder for alle beskrankninger som er blitt definert
# siden forrige "let-blokk". Enten man har beskrankninger eller ikke,
# skal det være en "let-blokk" til slutt.
#
# Beskrankninger lages ved å definere variabler som skal være null eller
# negative når beskrankningene er oppfylt.
#
# Eksempel på hvordan man legger inn beskrankningen Kåre,  $f(x) \leq g(x)$ :
# var Kåre = f(x)-g(x);

```

```

#
# Eksempel på hvordan man legger inn beskrankningen Olav,  $f(x)=g(x)$ :
#   var Olav = abs(f(x)-g(x));
#
# Som for modell-variable kan også alle beskrankninger samles i en fil som
#   kan inkluderes for å gi bedre oversikt slik :
#
#   include <myconstraints.con>;

let QM:=_nvars;
let {i in QD..QM} FeasTol[i] := 1e-1;
let {i in QD..QM} Pen[i] := 3;          # Beskrankningsvariable gis toleranse- og
                                       straffe-verdier i samlet målfunksjon i Hotdog
let QD:=QM+1;
# Dette er en "let-blokk".

# Søk og finn løsning(er) med Hotdog / Micio

include solve.txt # Filnavnet må tilhøre den versjonen av HotDog/Micio som
                  skal brukes

# Utskrift av resultatet
# Følgende filer genereres automatisk:
# "zInf" med alle beskrankninger hvor slakken er større en halvparten av
   toleransegrensen
# "zVar" med verdien og gradienten til alle variablene samt slakken og
   dualverdien til alle beskrankningene
# "zSol" som kan brukes til å hente inn løsningen i AMPL igjen på et senere
   tidspunkt
# (etter å ha lastet inn modellen og dataene skriver man "include zSol" )

print {i in 1.._snvars}: "let", _svarname[i], ":", _svar[i], ";" > ($bane &
'zSol' & '' & '.tmp');
print {i in QC+1..QM : _var[i] > 1e-4 } : _varname[i], _var[i] > ($bane &
'zInf' & '' & '.tmp');
print {i in _VARS} ('display ' & i & ', ' & i & '.rc, ' & i & '.dual' & " >
('' & $bane & "zVar' & '' & '.tmp');") > display.tmp;

close;

include display.tmp

close;

# Dette avslutter Hotdog/Micio-mal for modell-utvikling og -kjøring i AMPL.
# Ved bruk av andre løsningsverktøy kan man benytte ordinær AMPL metode for
# modell-utvikling som beskrevet i (38).

```

A.4 Eksempel

Et eksempel på kjøring av analysemodellen MSO².1 med datafil, kjøreoppsett, resultatfiler og Excel resultatbehandling er vist nedenfor for å illustrere bruk av modellen

A.4.1 Datafil

Data til analysemodellen gis i form av parametere og mengder. Disse spesifiseres i en datafil, hvis filnavn *.dat* må angis i kjøre-filen som beskrevet i kapittel A.2.3. En detaljert beskrivelse av dataspesifikasjon for modellen er gitt i (37)(38)

Det følgende er et eksempel på data for et scenario i Finnmark med 8 områder (A0,..., A8) i grafen og med komponenttyper fra alle kategorier (man, fir, c2, irs, ac, aac). Tiden går over 10 perioder med 6 timers tidssteg, dvs totalt 2.5 døgn. Det er angitt 1 fri strategi for side 1 og 2 strategier for side 2 hvorav 1 er fri og 1 begrenset.

Data for effektiviteter er fra tidligere MINOS database omsatt til aggregerte komponentdata og med anslåtte effektivitetsfaktorer for K2 og IRS (default-verdier) for illustrasjonsformål.

```
### Datafile

# Multi-scenario : n = 0 is single, n > 0 is multi
param n:=0; # no of scenarios - 1

param weight:= # pri of scenario
0 2
;

set parts[0] := 1 2; # two parts
param m := 0 1 1, 0 2 2; # side 1: 1 strat. , side 2: 2 strat.
param stype := [0,2,*] 1 0 2 1; # side 2: strat. 1=free, strat. 2= limited

# Component data
set ty := I P F C R E N M B S L; # I=inf,P=pn/pb,F=fire/art,C=K2,R=recon,E=ac,N=Lv
# M=mek, B=fire/art, S = aircraft, L=airdefence
param si := I 1 P 1 F 1 C 1 R 1 E 1 N 1 M 2 B 2 S 2 L 2; # 1=side 1, 2=side 2
param info := 1 1 2 0;
param def:= 0 1;

set ta[I] := S A; # S=sink/kontroll, A=avskjær/del, L=luftvern
set ta[P] := S A;
set ta[F] := S A ;
set ta[C] := CR ;
set ta[R] := CR ;
set ta[E] := AA AN LA ; # ac: xA=advanced amm, xN=ordinary amm, LA=SEAD
set ta[N] := L ;
set ta[M] := A S C; # C= C2W
set ta[B] := A S C;
set ta[S] := AA AN LA ;
set ta[L] := L ;

set isr := R ; # Intel/surv/recon comp.
set c2 := C ; # C2 comp.
set fir := F B ; # fire comp.
set man := I P M ; # maneuver comp.
set ac := E S ; # aircraft comp.
set aac := N L ; # antiaircraft comp.

# Area graph data (A0 = opp. start node/NEast, A8 = SWest node )
set ar := A0 A1 A2 A3 A4 A5 A6 A7 A8;
set base[E] := A8; # ac base node
set base[S] := A0;
set iar[0,2] := A0 ; # side 2 initial depl. node
set lar[2,0,2] := A0 A1 A2 A5 A6 A8; # side 2: limited strat. 2 nodes

set ad[A0] := A1 ; # Defines adj. connections in graph
set ad[A1] := A0 A2 A3 ;
set ad[A2] := A1 A3 A5 ;
set ad[A3] := A1 A2 A4 A6 ;
```



```

set ad[A4] := A3 A6 A7 ;
set ad[A5] := A2 A6 A8 ;
set ad[A6] := A3 A4 A5 A7 A8 ;
set ad[A7] := A4 A6 A8 ;
set ad[A8] := A5 A6 A7 ;

set trty := 1 2 ;          # Terraintypes: 1=defile. , 2=partly open

param terr :=             # Terraintype in nodes
A0 2
A1 2
A2 1
A3 2
A4 2
A5 1
A6 2
A7 2
A8 1;

# Time , Phase data

param las := 0 10 ;      # 20 periods = 5 days
param L := 0 6 ;        # 6 hrs pr period

#param lino := 0;        # lino=1: nonlinear, lino=0: linear objective

# Parameters Components

param cost :=
  I      0.0548
  P      0.1272
  F      0.0882
  C      0.0150
  R      0.0170
  E      0.1207
  N      0.0275 ;# Yearly LTC(mrd 2000-kr) = (Invest.-cost + MLU)/LT +
                O&M-cost/20 + ammo-cost
                # No Div+Brig overhead costs included : else + ca 50% increase
                # Component  Inv    MLU=Inv/3    O&M(20 years) LT(years)
                # I          0.67  0.22          0.080      20
                # Pn         3.00  1.00          0.282      30
                # Pb         1.10  0.36          0.260      30
                # P          2.40  0.80          0.275      30 P=(2Pn+Pb)/3 )
                # F          1.45  0.48          0.170      30
                # C          0.15  0.05          0.045      15
                # R          0.24  0.07          0.035      20 R= Oppkl-bn/6
                # E          1.8   0.6           0.040      30 E= 6 fly*0.3 mrd
                # N          0.34  0.11          0.040      20 N= 3 RB70enheter

                # I-amm0    0.5/4  0.0          0.0         20 fordelt på 4 enh
                # P-amm0    0.75/4 0.0          0.0         30 -----"-----
                # F-amm0    2.0/4  0.0          0.0         30 -----"-----
                # E-amm0    2.0/4  0.0          0.0         30 -----"-----

param budsj := 1 1.3 ;# 1.3 Total yearly LTC 2000-budget for material-related costs

param arv := [0,*] M 17 B 8 S 10 L 10 E 2 ;# Initial component types & sizes:
                # M(1 MID=9 MIB + 2 MBRIG=8 MIB),
                # B(RakBn), L(=3 lv-enh), E/S (=6 ac pr E/S)

# Parameters terrain

param max_in_node :=
  A0 200
  A1 6
  A2 3
  A3 6
  A4 6
  A5 3
  A6 6
  A7 6
  A8 3
;
# Max comp. pr node, A0 = initial source for opp. OOB

param max_speed :
  1 2 :=
  I 5 10
  P 10 20
  F 8 15
  C 10 20

```

```

R      5      10
E     500     500
N      8      15
M      8      15
B      8      15
S     500     500
L      8      15
;
# Comp. max-speed km/hr on axis

param col :=
  I 2
  P 1.5
  F 1.5
  C 1.0
  R 1.0
  E 1.0
  N 1.5
  M 1.5
  B 1.5
  S 1.0
  L 1.5
;
# Comp. col. length in km on axis

param r :
  A0 A1 A2 A3 A4 A5 A6 A7 A8 :=
  A0  2  20  70  80  110 150 150 160 230
  A1  20  2   50  70  90  130 135 145 210
  A2  70  50  2   70  110 150 125 140 200
  A3  80  70  70  2   60  80  85  90  160
  A4  110 90  110 60  2   60  50  50  130
  A5  150 130 150 80  60  2   30  75  170
  A6  150 135 125 85  50  30  2   45  85
  A7  160 145 140 90  50  75  45  2   170
  A8  230 210 200 160 130 170 85  170 2 ; # Node distances in km

param axis_cap :=
  [A1,*] A3 2
  [A3,*] A1 2
  [A3,*] A4 2
  [A4,*] A3 2
  [A4,*] A7 2
  [A7,*] A4 2;
# Axis capacity = no of lanes (default = 1)

param pri :=
  A1  3
  A2  2
  A3  3
  A4  6
  A5  8
  A6  6
  A7  6
  A8  10
;
# Parameters Effectiveness (eff/hr)

param eff :=
# Comp.= bn-size eff.

[0,I,M,*,*,0] : S A :=
  1 .040 .070
  2 .044 .090
[0,P,M,*,*,0] : S A :=
  1 .050 .130
  2 .054 .133
[0,F,M,*,*,0] : S A := # S=cgo , A=smart
  1 .00022 .0056 # Seffxengxtime=.00022x54x6=7%loss pr per.
  2 .00022 .0056
[0,E,M,*,*,0] : AN 0 := # Aircraft GA/AI (eff. pr amm-type : N=normal,
  1 .03 .3 # 0=other(advanced))
  2 .03 .3
[0,I,B,*,*,0] : S A :=
  1 .040 .070
  2 .044 .090
[0,P,B,*,*,0] : S A :=
  1 .050 .130
  2 .054 .133
[0,F,B,*,*,0] : S A := # KB-eff = Mek-eff
  1 .00022 .0056
  2 .00022 .0056
[0,E,B,*,*,0] : AN 0 :=

```

```

1      .03  .3
2      .03  .3
[0,I,L,*,*,0] : S      A      :=
1      .040 .070
2      .044 .090
[0,P,L,*,*,0] : S      A      :=
1      .050 .130
2      .054 .133
[0,F,L,*,*,0] : S      A      := # KB-eff = Mek-eff
1      .00022 .0056
2      .00022 .0056
[0,E,L,*,*,0] : AN     0      := # Suppr. of AD
1      .09   .67
2      .09   .67
[0,E,S,*,*,0] : 0      := # Air-air: own DCA (aggr. model)
1      .02
2      .02
[0,N,S,*,*,0] : L      := # Own AD : SAM
1      .00283
2      .00283

[0,M,I,*,*,0] : S      0      := # S=sink/kontroll, 0=all other tasks
1      .013 .0065
2      .050 .0082
[0,M,P,*,*,0] : S      0      :=
1      .062 .004
2      .065 .004
[0,M,F,*,*,0] : S      0      :=
1      .013 .0065
2      .050 .0082
[0,M,C,*,*,0] : S      0      :=
1      .062 .004
2      .065 .004
[0,M,R,*,*,0] : S      0      :=
1      .013 .0065
2      .050 .0082
[0,M,N,*,*,0] : S      0      :=
1      .013 .0065
2      .050 .0082
[0,B,I,*,*,0] : S      0      := # S=HE,A=cgo
1      .00002 .00006 # Seffxengxtime=.00002x36x6=0.4%loss pr per.
2      .00002 .00006
[0,B,P,*,*,0] : S      0      :=
1      .00006 .0002
2      .00006 .0002
[0,B,F,*,*,0] : S      0      :=
1      .00015 .0005
2      .00015 .0005
[0,B,C,*,*,0] : S      0      :=
1      .00015 .0005
2      .00015 .0005
[0,B,R,*,*,0] : S      0      :=
1      .00015 .0005
2      .00015 .0005
[0,B,N,*,*,0] : S      0      :=
1      .00015 .0005
2      .00015 .0005
[0,S,I,*,*,0] : AN     0      := # Aircraft GA/AI (eff. pr amm-type : N=normal,
                                0=other (advanced))
1      .06   .02
2      .06   .02
[0,S,P,*,*,0] : AN     0      :=
1      .0025 .01
2      .0025 .01
[0,S,F,*,*,0] : AN     0      :=
1      .0025 .01
2      .0025 .01
[0,S,C,*,*,0] : AN     0      := # Aircraft GA/AI : C2W (eff. as arty.)
1      .0025 .01
2      .0025 .01
[0,S,N,*,*,0] : AN     0      := # Suppr. of AD (3 x eff. as arty)
1      .0075 .03
2      .0075 .03
[0,S,E,*,*,0] : 0      := # Air-air : DCA (aggr. model)
1      .013
2      .013
[0,L,E,*,*,0] : L      := # SAM
1      .0044
2      .0044
;

```

```

# Parameters Fire Engagement rates      (eng/hr)

param frt :=
    F 200
    E 60
    N 30
    B 300
    S 60
    L 30
    ;

# Mean total-fire all fir-comp. pr hour
param max_r :=
    F 70
    C 90
    R 30
    E 500
    B 50
    S 500
    ;

# Max range for fir+C2 comp.

param eng :=
# Max-eng. pr comp. pr hour :
# F = No of shots pr hour pr comp.

[0,F,M,*,*,0] : S      A      :=
    1      54      9
    2      54      9
[0,F,B,*,*,0] : S      A      :=
    1      54      9
    2      54      9
[0,F,L,*,*,0] : S      A      :=
    1      54      9
    2      54      9
[0,E,M,*,*,0] : 0              := # Ac GA/AI : eng. rate = 4 x 6 amm pr 6 hours
    1      0.125
    2      0.125
[0,E,B,*,*,0] : 0              :=
    1      0.125
    2      0.125
[0,E,L,*,*,0] : 0              :=
    1      0.125
    2      0.125
[0,E,S,*,*,0] : 0              := # Air-air : (aggr. model)
    1      0.125
    2      0.125
[0,N,S,*,*,0] : L              := # Own SAM : 3 x 6 msl. fired pr 6 hours
    1      3
    2      3

[0,B,I,*,*,0] : 0              :=
    1      36
    2      36
[0,B,P,*,*,0] : 0              :=
    1      36
    2      36
[0,B,F,*,*,0] : 0              :=
    1      36
    2      36
[0,B,C,*,*,0] : 0              :=
    1      36
    2      36
[0,B,R,*,*,0] : 0              :=
    1      36
    2      36
[0,S,E,*,*,0] : 0              := # Air-air : (aggr. model)
    1      0.1
    2      0.1
[0,S,I,*,*,0] : 0              :=
    1      0.3
    2      0.3
[0,S,P,*,*,0] : 0              :=
    1      0.3
    2      0.3
[0,S,F,*,*,0] : 0              :=
    1      0.3
    2      0.3
[0,S,C,*,*,0] : 0              :=
    1      0.3
    2      0.3

```

```

[0,S,N,*,*,0] : 0 :=
  1 0.3
  2 0.3
[0,L,E,*,*,0] : L :=
  1 3
  2 3
[0,*,M,*,S,S] : 1 2 := # No S-S task combination : 0 eng. rate
  I 0 0
  P 0 0
[0,M,*,*,S,S] : 1 2 :=
  I 0 0
  P 0 0
;

param task_creator := # Allocate 10 x rel. ac-effort against L and N
[E,LA,L] 10
[S,LA,N] 10
[M,C,*] C 100 R 50
[B,C,*] C 100 R 50
;

param norm := I 0.00216 P 0.00702 M 0.03114 ;
#param low := I 1.2 P 1.2 M 1.4 ;
#param high := I 0.5 P 0.9 M 0.8 ;

param w := [0,*] M 0.0921 B 0.0485 S 0.2763 L 0.0485; # Comp. weight = cost
model;

let {i in ty} w[0,i] := 100*w[0,i]; # Normalize weights
let scale := 1/w[0,'P']; # Scale strength relative P

```

A.4.2 Kjøreoppsett

Når man skal foreta en kjøring, bør man opprette en ny mappe slik at man får samlet alle filene som genereres på en oversiktlig måte. Modellfilen, datafilen, beskrankningsfilen, kjørefilen og HOTDOG/MICIO løsningsalgoritme-fil må kopieres til denne mappen. Dessuten må man opprette filen *LOOP.tmp*, som skal være blank, og filen *HTdisplay.tmp*, som kan inneholde eventuelle kommandoer man vil utføre for hvert lokalpunkt HOTDOG finner.

I kjørefilen (se kapittel A.2.3) må man påse at det ligger korrekte referanser til modell-, data-, beskranknings- og solve-filene. Hvis AMPL kjøres fra en annen mappe enn den man skal ha filene i, må man spesifisere det fulle banenavnet til mappen med filene på denne måten "option bane 'C:\Hotdog\Kjøring3\' ". Man kan også endre på parametere som for eksempel antallet lokalsøk man vil utføre.

I analysemodellen gjennomføres gjentatte iterasjoner slik at man lar hver part i konflikten få flere sjanser til å forbedre strategien(e) sin(e) mot motstanderens. Derfor oppretter man en fil, f. eks *MSO2.run*, som skal inneholde kommandoen for å sette igang kjørefilen, "include *Myrun.run*" eller liknende, samme antallet ganger som man ønsker iterasjoner. Deretter må man gå inn i AMPL og inkludere denne filen.

En fullstendig kjøring gjennomføres slik :

- 1 Opprett ny mappe, f. eks : 'C:\Hotdog\Kjøring3\'
- 2 Kopier inn modell-, data-, beskrankning-, og HotDog/Micio
- 3 Opprett filene 'LOOP.tmp' (blank) og 'HTdisplay.tmp'
- 4 Opprett kjørefil 'MSO2.run' som består av:

```

include Myrun.run;
include Myrun.run;
include Myrun.run;
etc....          # Gjentatt ønsket antall iterasjoner
5 Gå til ny mappe: 'cd C:\Hotdog\Kjøring3'
6 Start AMPL kjøring: 'ampl MSO2.run'
7 Se på resultatfiler og lag Excel diagram (kfr kapittel A.4.4)

```

Hvis man ønsker å utføre flere slike kjøring etter hverandre, bør man lage en mappe til hver kjøring som beskrevet over. Deretter lager man en batch-fil (i MS-DOS vil det si at filnavnet slutter med *.bat*) som har følgende form:

```

cd C:\Hotdog\firstrun # går til mappen som inneholder kjøring 1
ampl MSO2.run         # MSO2.run er kjørefil i denne mappen
cd C:\Hotdog\nextrun # går til mappe for neste kjøring
ampl MSO2.run         # MSO2.run er kjørefil i denne mappen
etc....

```

Batch-kjøringene startes i et MS-DOS vindu ved å gi batch-fil som kommando.

A.4.3 Resultatfiler (utdrag)

zSummary.tmp (utdrag)

```

LOOP = 16
cu = 1
V = -0.64806
f = -0.64806
Obj = -0.64803

  if c == 'prb' then prob[u,st] else M[u,c + st*10^(cu - 1),cu] :=
0 1 10    0.64806
0 1 20    0.64822
0 1 prb   1
;

I      2.406  0.1319
P      3.436  0.437
F      2.297  0.2026
C      15.56  0.2334
R      5.186  0.08816
E      0.0337 0.00407
N      7.38  0.2029

LOOP = 16
cu = 2
V = 0.34593
f = 0.4001
Obj = 0.4001

  if c == 'prb' then prob[u,st] else M[u,c + st*10^(cu - 1),cu] [0,*,*]
:      1      prb      :=
1  -0.34593  1
2  -0.61675  0
;

M      0      0
B      0      0
S      0      0
L      0      0

LOOP = 17
cu = 1

```

V = -0.68229
 f = -0.68229
 Obj = -0.68217

```
if c == 'prb' then prob[u,st] else M[u,c + st*10^(cu - 1),cu] :=
0 1 10    0.68229
0 1 20    0.68622
0 1 prb   1
;
```

```
I    5.742  0.3147
P    1.6    0.2035
F    1.478  0.1303
C    15.26  0.2289
R    19.47  0.3311
E    0.1799 0.02172
N    2.544  0.06996
```

LOOP = 17
 cu = 2
 V = 0.4953
 f = 0.5288
 Obj = 0.5288

```
if c == 'prb' then prob[u,st] else M[u,c + st*10^(cu - 1),cu] [0,*,*]
:    1    prb    :=
1   -0.4953    1
2   -0.66282   0
;
```

```
M    0    0
B    0    0
S    0    0
L    0    0
```

LOOP = 18
 cu = 1
 V = -0.62784
 f = -0.62784
 Obj = -0.62784

```
if c == 'prb' then prob[u,st] else M[u,c + st*10^(cu - 1),cu] :=
0 1 10    0.62784
0 1 20    0.66916
0 1 prb   1
;
```

```
I    3.52   0.1929
P    2.926  0.3722
F    4.452  0.3927
C    10.56  0.1584
R    2.791  0.04745
E    0.4301 0.05192
N    3.071  0.08445
```

LOOP = 18
 cu = 2
 V = 0.45185
 f = 0.48771
 Obj = 0.48771

```
if c == 'prb' then prob[u,st] else M[u,c + st*10^(cu - 1),cu] [0,*,*]
:    1    prb    :=
1   -0.45185    1
2   -0.63114    0
;
```

```
M    0    0
B    0    0
S    0    0
L    0    0
```

zHot17-1-0.tmp

Variabler: 1793
 Beskrænkninger: 1751

Draw	Iters	Total	Grad	Feas	Penalty	Function	End
1	15	15	0.4307	1	6.4e-05	-0.5265	Kan ikke forbedres
2	147	162	3.301	0	0.09893	-0.6486	Kan ikke forbedres

3	125	287	0.3138	1	1e-05	-0.5998	Kan ikke forbedres
4	104	391	3.595	0	0.362	-0.6081	Kan ikke forbedres
5	126	517	1.619	1	3.6e-05	-0.6007	Kan ikke forbedres
6	165	682	0.3572	1	4.3e-07	-0.6213	Kan ikke forbedres
7	119	801	4.969	1	0.00017	-0.6797	Kan ikke forbedres
8	170	971	0.2171	1	3.2e-06	-0.6016	Kan ikke forbedres
9	169	1140	2.354	1	0.00013	-0.6823	Kan ikke forbedres
10	204	1344	0.27	1	1.3e-06	-0.625	Kan ikke forbedres

Gradientberegninger:4081

Lokale maksimalpunkter:

```

Objs[j] [*] :=
1 -0.52648
3 -0.59978
5 -0.60064
6 -0.62129
7 -0.67954
8 -0.60161
9 -0.68217
10 -0.625
;

```

Globalt maksimalpunkt: (?)

```

best_rep = 9
f = -0.68229
Obj = -0.68217
feas = 1

```

zHot17-2-1.tmp

Variabler: 638

Beskrænkninger: 715

Draw	Iters	Total	Grad	Feas	Penalty	Function	End
1	18	18	0.02534	1	1.3e-06	0.5293	Lokalpunkt
2	168	186	0.0268	1	9.9e-07	0.5293	Lokalpunkt
3	132	318	0.01346	1	6.4e-07	0.5317	Lokalpunkt
4	160	478	0.02398	1	1.2e-06	0.5303	Lokalpunkt
5	107	585	0.02363	1	1.4e-06	0.5313	Lokalpunkt
6	134	719	0.02191	1	2.5e-07	0.5314	Lokalpunkt
7	190	909	0.0125	1	2.1e-07	0.5265	Lokalpunkt

Gradientberegninger:2242

Lokale maksimalpunkter:

```

Objs[j] [*] :=
1 0.52932
2 0.52926
3 0.53165
4 0.5303
5 0.53131
6 0.53145
7 0.52647
;

```

Globalt maksimalpunkt: (?)

```

best_rep = 7
f = 0.52647
Obj = 0.52647
feas = 1

```

zVar17-2.tmp (uddrag)

LOOP = 17

cu = 2

:	start	start.rc	start.dual	:=
0 B A0 1	8	-0.0014059	0	
0 B A0 2	8	-0.00016904	0	
0 C A1 1	2.646	0.00013883	0	
0 C A2 1	0	0	0	
0 C A3 1	2.526	0.0004577	0	
0 C A4 1	1.772	0.00052244	0	
0 C A5 1	3.527	0.00028844	0	
0 C A6 1	2.163	0.0004834	0	
0 C A7 1	1.308	0.00048457	0	
0 C A8 1	1.317	0.0004482	0	


```

0 E A8 1      2.18      0.0045859      0
0 F A1 1      0          0.00010292     0
0 F A2 1      0          0              0
0 F A3 1      0.2815     0.0082593      0
0 F A4 1      0.1738     0.0084759      0
0 F A5 1      0.1959     0.0079158      0
0 F A6 1      0.4044     0.0083642      0
0 F A7 1      0.2521     0.0081471      0
0 F A8 1      0.17       0.0081808      0
0 I A1 1      0          0.00029602     0
0 I A2 1      0          0              0
0 I A3 1      0.0293     0.004225       0
0 I A4 1      2.144      0.0040909      0
0 I A5 1      0.0249     0.0016414      0
0 I A6 1      1.374      0.0043556      0
0 I A7 1      2.17       0.0041447      0
0 I A8 1      0          0.00023544     0
0 L A0 1      10         -0.00061606    0
0 L A0 2      10         -0.00013102    0
0 M A0 1      17         -0.0015192     0
0 M A0 2      17         -0.00016043    0
0 N A1 1      0          9.3183e-05     0
0 N A2 1      0          0              0
0 N A3 1      0.1384     0.0012148      0
0 N A4 1      0.09712    0.0011353      0
0 N A5 1      0.6746     0.0016102      0
0 N A6 1      0.1837     0.0015119      0
0 N A7 1      0.09677    0.0011398      0
0 N A8 1      1.353      0.0011824      0
0 P A1 1      0.3803     0.0067494      0
0 P A2 1      0          0              0
0 P A3 1      0.1737     0.0079581      0
0 P A4 1      0.1902     0.008197       0
0 P A5 1      0.2673     0.011422       0
0 P A6 1      0.267      0.010593       0
0 P A7 1      0.174      0.010017       0
0 P A8 1      0.147      0.010289       0
0 R A1 1      4.121      0.00025286     0
0 R A2 1      0          0              0
0 R A3 1      0          0              0
0 R A4 1      1.665      0.00044914     0
0 R A5 1      6.28       0.00050967     0
0 R A6 1      3.431      0.00048714     0
0 R A7 1      3.976      0.0004245      0
0 R A8 1      0          0              0
0 S A0 1      10         -0.001973      0
0 S A0 2      10         -0.00055031    0
;

```

```

:      probwin1 probwin1.rc probwin1.dual :=
0 11 A1 2      0.59617      0      0
0 11 A1 3      0.5          0      0.0068182
0 11 A1 4      0.28892      0      0.0068182
0 11 A1 5      0.38852      0      0.0068182
0 11 A1 6      0.32734      0      0.0068182
0 11 A1 7      0.38908      0      0.0068182
0 11 A1 8      0.38771      0      0.0068182
0 11 A1 9      0.39713      0      0.0068182
0 11 A1 10     0.33669      0      0.0068182
0 11 A2 2      0.5          0      0
0 11 A2 3      0.5          0      0.0045455
0 11 A2 4      0.50002      0      0.0045455
0 11 A2 5      0.44286      0      0.0045455
0 11 A2 6      0.4668       0      0.0045455
0 11 A2 7      0.28589      0      0.0045455
0 11 A2 8      0.29388      0      0.0045455
0 11 A2 9      0.28047      0      0.0045455
0 11 A2 10     0.072954     0      0.0045455
0 11 A3 2      0.60711      0      0
0 11 A3 3      0.53335      0      0.0068182
0 11 A3 4      0.53303      0      0.0068182
0 11 A3 5      0.16978      0      0.0068182
0 11 A3 6      0.42937      0      0.0068182
0 11 A3 7      0.26836      0      0.0068182
0 11 A3 8      0.042264     0      0.0068182
0 11 A3 9      0.078617     0      0.0068182
0 11 A3 10     0.1645       0      0.0068182
0 11 A4 2      0.60452      0      0
0 11 A4 3      0.6093       0      0.013636
0 11 A4 4      0.63326     0      0.013636
0 11 A4 5      0.646       0      0.013636
0 11 A4 6      0.59973     0      0.013636

```

```

0 11 A4 7      0.57679      0      0.013636
0 11 A4 8      0.55446      0      0.013636
0 11 A4 9      0.053323      0      0.013636
0 11 A4 10     0.037648      0      0.013636
0 11 A5 2      0.67537      0      0
0 11 A5 3      0.6396      0      0.018182
0 11 A5 4      0.64776      0      0.018182
0 11 A5 5      0.60221      0      0.018182
0 11 A5 6      0.60536      0      0.018182
0 11 A5 7      0.59937      0      0.018182
0 11 A5 8      0.57962      0      0.018182
0 11 A5 9      0.57321      0      0.018182
0 11 A5 10     0.55783      0      0.018182
0 11 A6 2      0.71045      0      0
0 11 A6 3      0.62868      0      0.013636
0 11 A6 4      0.64881      0      0.013636
0 11 A6 5      0.65638      0      0.013636
0 11 A6 6      0.56419      0      0.013636
0 11 A6 7      0.082803      0      0.013636
0 11 A6 8      0.20369      0      0.013636
0 11 A6 9      0.23703      0      0.013636
0 11 A6 10     0.053854      0      0.013636
0 11 A7 2      0.61383      0      0
0 11 A7 3      0.98444      0      0.013636
0 11 A7 4      0.59949      0      0.013636
0 11 A7 5      0.61461      0      0.013636
0 11 A7 6      0.52981      0      0.013636
0 11 A7 7      0.55629      0      0.013636
0 11 A7 8      0.56517      0      0.013636
0 11 A7 9      0.53462      0      0.013636
0 11 A7 10     0.030625      0      0.013636
0 11 A8 2      0.65917      0      0
0 11 A8 3      0.63109      0      0.022727
0 11 A8 4      0.6338      0      0.022727
0 11 A8 5      0.58336      0      0.022727
0 11 A8 6      0.56998      0      0.022727
0 11 A8 7      0.56659      0      0.022727
0 11 A8 8      0.57635      0      0.022727
0 11 A8 9      0.59017      0      0.022727
0 11 A8 10     0.57964      0      0.022727
;

```

```

:      M      M.rc M.dual      :=
0 11 1      0.4953      0      0
0 11 2      -0.4953      0      -0.8
0 21 1      0.66282      0      0
0 21 2      -0.66282      0      -0.2
;

```

```

:      value value.rc value.dual      :=
0      -0.5288      0      -1
;

```

```

f = 0.5288
f.rc = 0
f.dual = 1

```

```

Total_cost = 0
Total_cost.rc = 0
Total_cost.dual = 0

```

zInf17-2.tmp

```

LOOP = 17
cu = 2

```

(I denne kjøringen ble ingen av beskrankningene brutt med mer enn halvparten av toleransegrensen.)

zExp12-2.tmp (utdrag)

```

'Informasjonens effektivitetsfaktor (manøverstyrkene):'

```

```

if info[x] == 0 then 1 else max_eff[u] - (max_eff[u] - min_eff[u])*exp(-
(rate_eff[u]*sqrt(eps + calc[u,c,t,'C2',o,0,0
,x,0]*calc[u,c,t,'ISR',o,0,0,x,0]))) [0,1,11,*,*] (tr)

```

```

:      A1      A2      A3      A4      A5      A6      A7      A8      :=

```

2	1.2858	0.74311	0.7673	1.2795	1.2996	1.2997	1.2945	0.73633
3	0.78601	0.71541	0.75178	1.252	1.2974	1.2983	1.2686	0.73478
4	0.73642	0.7205	1.2162	1.2906	1.2985	1.2991	1.2811	1.2103
5	0.73458	0.71351	0.97817	1.2796	1.2919	1.2947	1.2516	1.1774
6	0.74325	0.71316	0.74906	1.295	1.2876	1.2912	1.2433	1.0735
7	0.74739	0.71446	0.90149	1.2977	1.2921	1.2945	1.2867	1.1023
8	0.74743	0.7223	1.2419	1.2969	1.2914	1.2939	1.286	1.0965
9	0.74225	0.72046	1.2123	1.2923	1.289	1.2925	1.1817	1.1034
10	0.7413	0.72168	1.1973	1.2943	1.2825	1.2878	1.1713	1.1424

```
[0,1,21,*,*] (tr)
:      A1      A2      A3      A4      A5      A6      A7      A8      :=
2      1.2858    0.74311  0.7673  1.2795  1.2996  1.2997  1.2945  0.73633
3      0.78602    0.71541  0.75178  1.252   1.2974  1.2983  1.2686  0.73478
4      0.73643    0.72051  1.2162   1.2906  1.2985  1.2991  1.2811  1.2102
5      0.73447    0.71316  0.74672  1.2794  1.2918  1.2947  1.2514  1.1773
6      0.74535    0.71316  0.81814  1.2963  1.2897  1.2926  1.2493  1.0733
7      0.74998    0.71328  0.8506   1.2986  1.2942  1.296   1.2899  1.1092
8      0.73918    0.76192  1.1753   1.2716  1.2867  1.291   1.2811  1.1039
9      0.95676    0.86431  1.2316   1.0284  1.2757  1.2854  1.1468  1.1314
10     0.93335    1.1087   1.2793   1.2376  1.2903  1.294   1.2001  1.2111
```

'Fordeling av komponentene på oppdrag:' = 'Fordeling av komponentene på oppdrag:'

```
(sum{o in be[u,i,str[si[i],c]], t in time[u]}
calc[u,c,t,'force',o,i,0,0,0]*task[u,i,o,t,str[si[i],c],a])/(eps + sum
{o in be[u,i,str[si[i],c]], t in time[u]} calc[u,c,t,'force',o,i,0,0,0])
[0,11,*,*]
:      A      AA      AN      C      CR      L      LA      S
:=
B      0.72116  .      .      0.13684  .      .      .      0.14197
C      .      .      .      .      0.99999  .      .      .
E      .      0.12249  0.11126  .      .      .      0.7662  .
F      0.95651  .      .      .      .      .      .      0.043355
I      0.84885  .      .      .      .      .      .      0.15111
L      .      .      .      .      .      0.99998  .      .
M      0.4505   .      .      0.073552  .      .      .      0.47594
N      .      .      .      .      .      0.99994  .      .
P      0.87698  .      .      .      .      .      .      0.12291
R      .      .      .      .      0.99999  .      .      .
S      .      0.485    0.4922  .      .      .      0.022794  .
```

```
[0,21,*,*]
:      A      AA      AN      C      CR      L      LA      S
:=
B      0.68403  .      .      0.20092  .      .      .      0.11502
C      .      .      .      .      0.99999  .      .      .
E      .      0.12132  0.11132  .      .      .      0.7673  .
F      0.9616   .      .      .      .      .      .      0.038268
I      0.79781  .      .      .      .      .      .      0.20215
L      .      .      .      .      .      0.99999  .      .
M      0.42912  .      .      0.07063  .      .      .      0.50024
N      .      .      .      .      .      0.99994  .      .
P      0.87904  .      .      .      .      .      .      0.12084
R      .      .      .      .      0.99999  .      .      .
S      .      0.43094  0.53841  .      .      .      0.030637  .
;
```

'Fordeling av tap på komponenttyper:' = 'Fordeling av tap på komponenttyper:'

```
sum((u,i,a,o,t,c) in Tap: o in oar[u] || i in ac) (( if loss == 'man' && o in oar[u]
then calc[u,c,t,'lossm',o,i,a,0
,0]) + ( if loss == 'fir' && o in oar[u] then calc[u,c,t,'lossf',o,i,a,0,0]) + ( if
loss == 'ac' && o in oar[u]
then calc[u,c,t,'lossac',o,i,a,0,0]) + ( if loss == 'aac' then
calc[u,c,t,'lossaac',o,i,a,0,0]))*( min(1,calc[u,c,t
,'pol',o,i,a,0,0]*(1 - exp(1 - abs(1 + 1.00587*((sum{ {1}: o in oar[u]}
(calc[u,c,t,'lossm',o,i,a,0,0] + calc[u,c,t
,'lossf',o,i,a,0,0] + calc[u,c,t,'lossac',o,i,a,0,0]) + sum{ {1}: i in ac}
calc[u,c,t,'lossaac',o,i,a,0,0]))/(
calc[u,c,t,'pol',o,i,a,0,0] + 3*eps)) + 0.432961*((sum{ {1}: o in oar[u]}
(calc[u,c,t,'lossm',o,i,a,0,0] + calc[u,c
,t,'lossf',o,i,a,0,0] + calc[u,c,t,'lossac',o,i,a,0,0]) + sum{ {1}: i in ac}
calc[u,c,t,'lossaac',o,i,a,0,0]))/(
calc[u,c,t,'pol',o,i,a,0,0] + 3*eps))^2 + 2.93395*((sum{ {1}: o in oar[u]}
(calc[u,c,t,'lossm',o,i,a,0,0] +
calc[u,c,t,'lossf',o,i,a,0,0] + calc[u,c,t,'lossac',o,i,a,0,0]) + sum{ {1}: i in ac}
calc[u,c,t,'lossaac',o,i,a,0,0]
)))/(calc[u,c,t,'pol',o,i,a,0,0] + 3*eps))^5)))/(eps + ( if o in oar[u] then
calc[u,c,t,'lossm',o,i,a,0,0] + calc[u,c
,t,'lossf',o,i,a,0,0] + calc[u,c,t,'lossac',o,i,a,0,0]) + ( if i in ac then
```

```

calc[u,c,t,'lossaac',o,i,a,0,0])))) [0,11,*,*]
:      aac      ac      fir      man      :=
B      .      0.076591  0.48371  1.2865
C      .      0.056733  1.2393  5.1463
E      1.0669  0      0      0
F      .      0.097217  0.54166  0.55243
I      .      2.1497  0.016945  3.2701
L      .      1.0523  0.08177  0.58054
M      .      0.15994  2.2782  3.6605
N      .      0.15062  0.00047967  1.0474
P      .      0.24213  0.12159  0.43443
R      .      0      1.8782  3.683
S      0.39264  0      0      0

```

```

[0,21,*,*]
:      aac      ac      fir      man      :=
B      .      0.054607  0.143  0.19708
C      .      0.025423  0.73417  2.7861
E      1.2337  0      0      0
F      .      0.081841  0.20203  0.50968
I      .      3.3145  0.01519  1.2184
L      .      0.90642  0.082553  0.21662
M      .      0.24174  1.3263  2.8438
N      .      0.27171  0.00082247  0.74169
P      .      0.18448  0.015841  0.7735
R      .      0      1.4063  1.4626
S      0.39844  0      0      0
;

```

'Totale styrker og tap på hver side:' = 'Totale styrker og tap på hver side:'

```

# $1 = scale*(sum{i in ty: si[i] == x && i in tyo[u,c,o]} (calc[u,c,t -
1,'force',o,i,0,0,0] + calc[u,c,t,'din',o,i,0,0,0])*
w[u,i])

```

```

# $2 = scale*(sum{i in ty: si[i] == x && i in tyo[u,c,o]} (calc[u,c,t -
1,'force',o,i,0,0,0] + calc[u,c,t,'din',o,i,0,0,0]
- calc[u,c,t,'disp',o,i,0,0,0])*w[u,i])

```

```

:      $1      $2      :=
0 11 1 A1 2      1.2431      0.014664
0 11 1 A1 3      0.0030843  0.0030843
0 11 1 A1 4      0.037989  0.037989
0 11 1 A1 5      0      0
0 11 1 A1 6      0      0
0 11 1 A1 7      0      0
0 11 1 A1 8      0      0
0 11 1 A1 9      0      0
0 11 1 A1 10     0.01637  0.00059568
0 11 1 A2 2      0      0
0 11 1 A2 3      0.34947  0.048146
0 11 1 A2 4      0.0012463  0.0012463
0 11 1 A2 5      0      0
0 11 1 A2 6      0.011416  0.0022341
0 11 1 A2 7      0.018116  0.0035352
0 11 1 A2 8      0.011586  0.0023708
0 11 1 A2 9      0.34938  0.059532
0 11 1 A2 10     0.22621  0.19194
0 11 1 A3 2      0.70931  0.0040738
0 11 1 A3 3      0.936  0.013752
0 11 1 A3 4      0.28734  0.20537
0 11 1 A3 5      0.043319  0.043319
0 11 1 A3 6      0.023546  0.0089603
0 11 1 A3 7      0.66319  0.34714
0 11 1 A3 8      0.31605  0.074386
0 11 1 A3 9      0.24166  0.046113
0 11 1 A3 10     0.42798  0.059252
0 11 1 A4 2      1.6869  0.12451
0 11 1 A4 3      2.1717  0.087524
0 11 1 A4 4      2.1454  0.013559
0 11 1 A4 5      3.0828  0.020189
0 11 1 A4 6      3.3505  0.11945
0 11 1 A4 7      1.8671  0.18892
0 11 1 A4 8      1.5356  0.33679
0 11 1 A4 9      1.7969  0.27191
0 11 1 A4 10     1.3864  0.19666
0 11 1 A5 2      1.8149  0.003966
0 11 1 A5 3      1.9433  0.015504
0 11 1 A5 4      1.7911  0.011783
0 11 1 A5 5      1.4106  0.035926
0 11 1 A5 6      1.481  0.042514
0 11 1 A5 7      1.4313  0.029556
0 11 1 A5 8      1.2065  0.043024

```

0 11 1 A5 9	1.1734	0.032039
0 11 1 A5 10	0.9288	0.044425
0 11 1 A6 2	1.8927	0.0079239
0 11 1 A6 3	2.2778	0.057737
0 11 1 A6 4	2.854	0.052027
0 11 1 A6 5	2.6663	0.038999
0 11 1 A6 6	1.7029	1.035
0 11 1 A6 7	0.716	0.26953
0 11 1 A6 8	0.049895	0.018291
0 11 1 A6 9	0.57908	0.2253
0 11 1 A6 10	0.54253	0.084167
0 11 1 A7 2	1.9902	0.08269
0 11 1 A7 3	0.88979	0
0 11 1 A7 4	1.0299	0.049549
0 11 1 A7 5	0.89341	0.017821
0 11 1 A7 6	1.3226	0.092693
0 11 1 A7 7	1.9421	0.12526
0 11 1 A7 8	1.9542	0.13679
0 11 1 A7 9	0.66241	0.35362
0 11 1 A7 10	0.30879	0.095143
0 11 1 A8 2	2.7813	0.0023638
0 11 1 A8 3	3.2929	0.015382
0 11 1 A8 4	3.2949	0.028753
0 11 1 A8 5	2.9913	0.26172
0 11 1 A8 6	2.8164	0.25984
0 11 1 A8 7	2.4765	0.20867
0 11 1 A8 8	2.6653	0.18539
0 11 1 A8 9	2.4236	0.13784
0 11 1 A8 10	2.324	0.27539
0 11 2 A1 2	0	0
0 11 2 A1 3	7.2207	3.7687e-05
0 11 2 A1 4	9.4396	0.013415
0 11 2 A1 5	6.7116	0.04466
0 11 2 A1 6	5.6996	0.081918
0 11 2 A1 7	1.8604	0.045269
0 11 2 A1 8	1.4892	0.030474
0 11 2 A1 9	1.2163	0.010542
0 11 2 A1 10	0.3297	0.00060906
0 11 2 A2 2	0	0
0 11 2 A2 3	0	0
0 11 2 A2 4	1.1047	0.00024099
0 11 2 A2 5	1.6119	0.072369
0 11 2 A2 6	1.6148	0.040778
0 11 2 A2 7	1.5551	0.032194
0 11 2 A2 8	1.6537	0.021039
0 11 2 A2 9	1.9801	0.028942
0 11 2 A2 10	2.8272	0.065532
0 11 2 A3 2	0	0
0 11 2 A3 3	0	0
0 11 2 A3 4	3.4164	0.31641
0 11 2 A3 5	10.519	0.2714
0 11 2 A3 6	1.2862	0.0067301
0 11 2 A3 7	5.182	0.19415
0 11 2 A3 8	2.2865	0.0099132
0 11 2 A3 9	1.403	0.011768
0 11 2 A3 10	1.3964	0.029516
0 11 2 A4 2	0	0
0 11 2 A4 3	0	0
0 11 2 A4 4	0	0
0 11 2 A4 5	0	0
0 11 2 A4 6	0.019354	0.019354
0 11 2 A4 7	0.79194	0.38257
0 11 2 A4 8	5.9099	0.28992
0 11 2 A4 9	3.565	0.10513
0 11 2 A4 10	3.4599	0.057151
0 11 2 A5 2	0	0
0 11 2 A5 3	0	0
0 11 2 A5 4	0	0
0 11 2 A5 5	0	0
0 11 2 A5 6	0	0
0 11 2 A5 7	0.023396	0.023396
0 11 2 A5 8	0.027069	0.027069
0 11 2 A5 9	0.037642	0.037642
0 11 2 A5 10	0.11267	0.085598
0 11 2 A6 2	0	0
0 11 2 A6 3	0	0
0 11 2 A6 4	0	0
0 11 2 A6 5	0	0
0 11 2 A6 6	9.8336	1.867
0 11 2 A6 7	6.9881	0.48867
0 11 2 A6 8	3.8063	0.0919
0 11 2 A6 9	4.0961	0.15656

0 11 2 A6 10	3.4593	0.026082
0 11 2 A7 2	0	0
0 11 2 A7 3	0	0
0 11 2 A7 4	0	0
0 11 2 A7 5	0	0
0 11 2 A7 6	0	0
0 11 2 A7 7	0.015068	0.015068
0 11 2 A7 8	0.034582	0.034582
0 11 2 A7 9	2.365	0.16891
0 11 2 A7 10	1.8783	0.011067
0 11 2 A8 2	0	0
0 11 2 A8 3	0	0
0 11 2 A8 4	0	0
0 11 2 A8 5	0	0
0 11 2 A8 6	0	0
0 11 2 A8 7	0.021848	0.021848
0 11 2 A8 8	0.02753	0.02753
0 11 2 A8 9	0.0392	0.0392
0 11 2 A8 10	0.68012	0.55209

zStructures17.tmp

Struktur fra trekning nummer 1

Komp.	Antall	Kost
I	2.403	0.1317
P	3.432	0.4365
F	2.295	0.2024
C	15.47	0.232
R	5.174	0.08795
E	2.045	0.005393
N	7.421	0.2041

Struktur fra trekning nummer 2

Komp.	Antall	Kost
I	2.89	0.1584
P	1.602	0.2038
F	5.711	0.5037
C	9.403	0.1411
R	6.926	0.1177
E	2.004	0.000499
N	6.358	0.1748

Struktur fra trekning nummer 3

Komp.	Antall	Kost
I	1.895	0.1038
P	2.886	0.3671
F	4.887	0.431
C	15.68	0.2352
R	0	0
E	3.086	0.1311
N	1.155	0.03176

Struktur fra trekning nummer 4

Komp.	Antall	Kost
I	1.782	0.09764
P	4.415	0.5615
F	2.324	0.205
C	12.11	0.1816
R	2.76	0.04693
E	2.975	0.1177
N	0.9498	0.02612

Struktur fra trekning nummer 5

Komp.	Antall	Kost
I	0.8155	0.04469
P	8.008	1.019
F	0.9455	0.08339
C	5.32	0.0798
R	0	0
E	2.594	0.07168
N	0.0678	0.001863

Struktur fra trekning nummer 6

Komp.	Antall	Kost
I	2.721	0.1491
P	3.697	0.4703
F	4.372	0.3856
C	8.164	0.1225
R	6.295	0.107
E	2.119	0.01441

N 1.859 0.05111

Struktur fra trekning nummer 7

Komp.	Antall	Kost
I	5.802	0.318
P	2.664	0.3389
F	1.249	0.1101
C	14.04	0.2105
R	7.747	0.1317
E	2.027	0.003304
N	6.821	0.1876

Struktur fra trekning nummer 8

Komp.	Antall	Kost
I	0.0066	0.0003627
P	5.257	0.6687
F	1.991	0.1756
C	12.42	0.1862
R	4.54	0.07719
E	3.585	0.1913
N	0.0231	0.0006362

Struktur fra trekning nummer 9

Komp.	Antall	Kost
I	5.742	0.3147
P	1.6	0.2035
F	1.478	0.1303
C	15.26	0.2289
R	19.47	0.3311
E	2.18	0.02172
N	2.544	0.06996

Struktur fra trekning nummer 10

Komp.	Antall	Kost
I	2.061	0.1129
P	3.887	0.4944
F	4.55	0.4013
C	15.12	0.2268
R	2.082	0.03539
E	2.028	0.003418
N	0.9366	0.02576

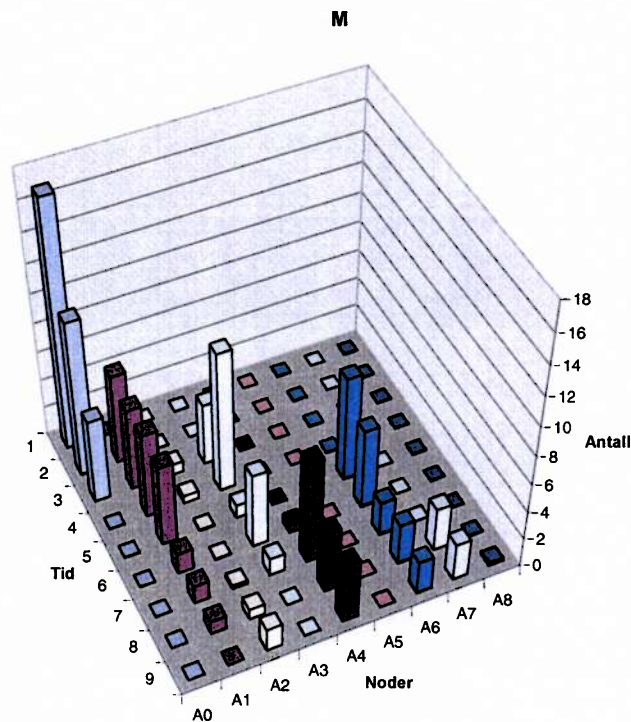
A.4.4 Excel resultatbehandling

Sammendragsfilen *zSummary.tmp* foran viser at det i iterasjon (*LOOP*) 17 for (*cu=current*) side 2 ble oppnådd en Maximin verdi for målfunksjonen (sannsynlighet for å lykkes, *f*) på 0.5288. Struktur for denne verdien var 5.7 'I' (kostnad 315 Mill kr årlig), 1.6 'P' (204 Mill kr), 1.5 'F' (130 Mill kr), 15 'C' (230 Mill kr), 19 'R' (330 Mill kr), 2.5 'N' (70 Mill kr) og 2 'E' (arv) som ble funnet å være best av ialt 10 lokalsøk ved optimering av side (*cu*) 1 som vist i filen *zStructures17.tmp* (*f*-verdi = 0.6823 i trekning nummer 9 fra filen *zHot17-1-0.tmp*). Dette må sies å være en 'moderne' struktur med stor vekt på ledelse og informasjon.

Maximin verdi for denne strukturen ble deretter funnet ved optimering av side (*cu*) 2 som vist i HOTDOG søkeresultater i filen *zHot17-2-1.tmp* (*f*-verdi = 0.5265 i trekning nummer 7). Dette er anslått beste (optimale) verdi for det antall iterasjoner som er foretatt, og vil gi det detaljerte strukturresultat som beskrives i det videre.

En fullstendig oversikt over detaljerte strukturdata for den optimale strukturen er gitt i filen *zVar17-2.tmp* med et utdrag vist i Appendiks A.4.3 over med bl a start-disponering av alle komponenter i noder i variabelen *start*. Ikke vist er tidsutviklingen av komponentene i noder i variabelen *disp*, men et eksempel på bruk av Excel resultatmakro **Komponentoversikt.xls** er gitt på Figur A.4.1 som viser variabelen *disp* for komponenten 'M' for side 2. Man ser at fra startposisjon i node 'A0' flytter komponentene via 'A1 A3 A4 A6' til 'A4 A6 A7 A8' i siste viste

periode 9. Antallet er da redusert fra 17 initielt til ca 10.9 pga 36 % tap (hovedsakelig til manøver- og ildstøtte-avdelinger, se filen *zExp17-2.tmp* over).

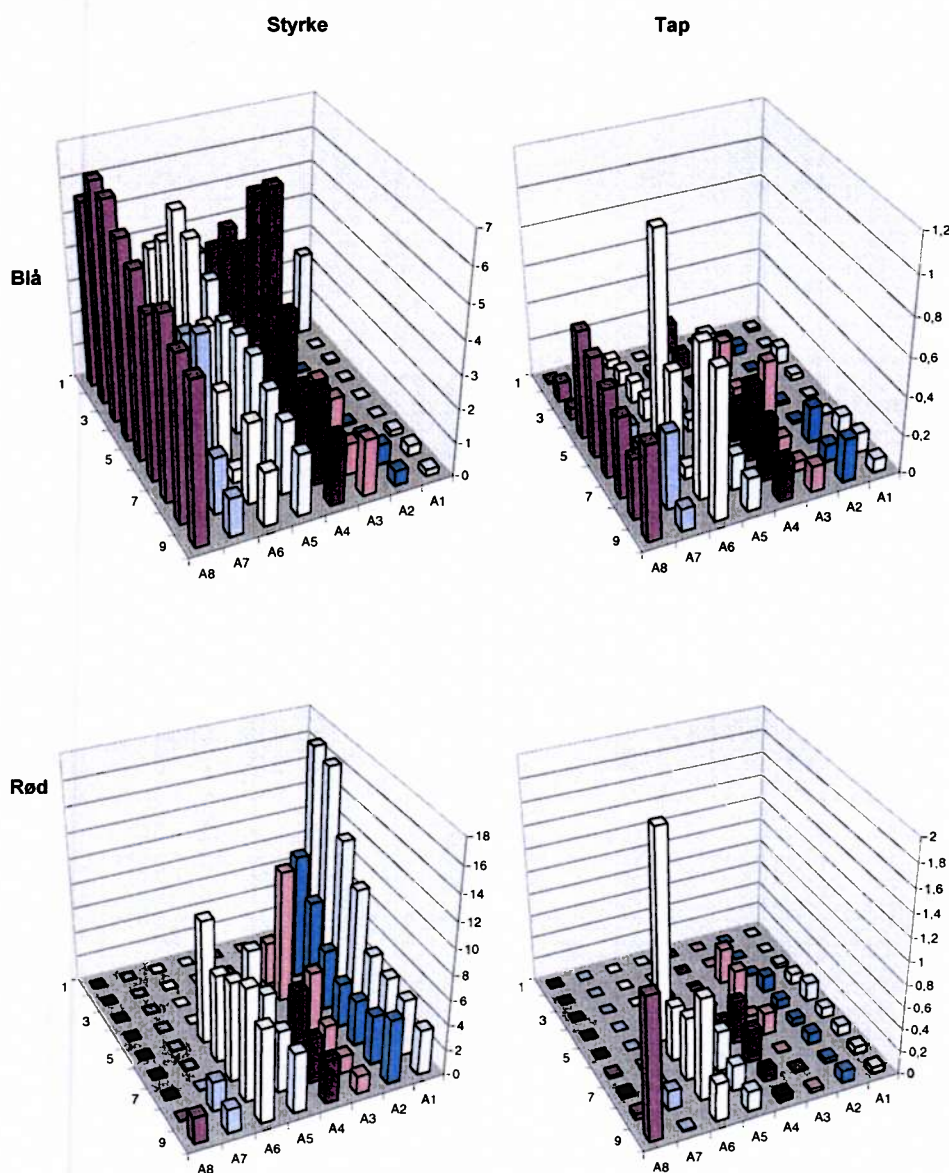


Figur A.4.1 Komponentutvikling i tid og rom i noder for komponenttype 'M' for side 2

En mer samlet oversikt over strukturutviklingen i stort kan også fås fra filen *zExp17-2.tmp* med et utdrag vist i Appendiks A.4.3. Der vises bl a informasjonseffektivitet pga K2 og ISR, fordeling av oppdrag og tap påført av ulike komponenttyper. En vektet totalstyrke og tap i tid og rom er også vist som et overordnet sammendrag av utviklingen i scenariet. Dette sammendraget lages med Excel resultatmakro **Sammendrag.xls** og er vist i Figur A.4.2 der det fremgår at side 1 (Blå) er disponert i de fleste noder initielt med en hovedstyrke i node 'A8' til slutt. Side 2 (Rød) er i node 'A1' i starten og flytter mot node 'A8' ved slutt med styrker også i de andre noder. Tapsutvikling skjer hovedsakelig i node 'A6' og 'A8'.

Maximin målfunksjon verdi (f -verdi) for strukturen med data som i eksemplet er 0.5288, noe som indikerer at styrkene er nær likeverdige uten at noen av sidene har et klart overtak. Dette avspeiles også delvis i de totale gjenværende styrker som vist ved slutt i Figur A.4.2.

Det skal også bemerkes at kjøringen er avsluttet etter 17 iterasjoner og at det ikke kan utelukkes at flere iterasjoner ville ha gitt bedre resultat sett fra side 1. Det er kun regnetid som vil bestemme en praktisk grense for antall iterasjoner som vil kunne gjennomføres for hver kjøring. Her vil teknologien bidra til en gunstig utvikling, både ved at hver enkelt datamaskin øker ytelsen og ved at iterasjonene kan kjøres i parallell på flere datamaskiner samtidig for å øke gjennomføringshastigheten ved bruk av analysemodellen.



Figur A.4.2 Sammendrag av styrke- og tapsutvikling i eksempel-scenariet for hver side

A.5 MIKS: Modell for innfasing av kosteffektiv struktur

Som beskrevet i kapittel 2.2.1.2 er det også utviklet en enkel lineær modell for minimering av total kostnad for innfasing over tid av den optimale struktur som er bestemt av analysemodellen $MSO^2.1$ under hensyn til (ulik) teknologisk fordyrelse av materiell. Denne modellen søker å fase inn hele eller deler av avdelinger (komponenter), slik at de til enhver tid er oppsatt med det materiell som inngår; derved vil materialet kunne bli faset inn i separate perioder og ikke som sammenhengende prosjekter.

Det er mulig å endre målfunksjonen som minimerer sum av avdelingsandel-kostnadene til å minimere totale materiellprosjekt-kostnader dersom man angir hvor mye materiellprisen påvirkes av "batch"-størrelsen på materialet.

Modellen MIKS illustrerer også en ny funksjon i AMPL som kobler modell- og dataspesifikasjon til 'Odbc'-kall for databaser (Access, Oracle, Tabeller, Ascii-filer) for MS WinNT. Derved kan en menybasert interaktiv applikasjon lages med direkte oppdatering av data fra databaser. En slik versjon for MS ACCESS er tilgjengelig ved SYS (kfr systemansvarlig) på katalog C:\HOTDOG\INNFAS\ INNFAS.MDB og inneholder menyfunksjoner som integrerer dataoppdatering, kjøring og resultatpresentasjon i en applikasjon.

Selve modellen MIKS skrevet i AMPL med deklarasjon av disse funksjonene er listet nedenfor og er rimelig selvforklarende. Merk at det er enkelt å beregne konsekvens av forsinkelse i innfasing av en avdeling (f eks nr 2) ved å ta med beskrankningen StrLimit (fjerne kommentartegnet #) og angi den tid forsinkelsen skal vare (f eks 5 år som vist). Derved kan betydningen av andre prioriteringer ved innfasingen vurderes opp mot økonomisk konsekvens direkte. Budsjett-endringer over tid er også antydnet, men ikke implementert.

```
# MIKS : Modell for Innfasing av Kost-effektiv Struktur

# Nullstill

reset;
reset options;

option omit_zero_rows 0;
option display_eps 1e-5;

option show_stats 0;
option times 0;

option linelim 0;
option substout 0;
option pl_linearize 0;
option presolve 0;
option solver minos;

# Deklarasjoner
set materiell;
set avdelinger;

param materiellnavn {materiell} symbolic;
param avdelingsnavn {avdelinger} symbolic;

param grunnpris {materiell};
param fordyrelse {materiell};
param avdelingsantall {avdelinger};
param materiellantall {avdelinger,materiell} default 0;
param budsjett;
param tid integer;
param Slakk integer;

var anskaffelse {avdelinger,1..tid} >=0 <=5;

# Minimer f= total innfasingskostnad inkludert teknologisk fordyrelse
minimize f: sum {a in avdelinger,m in materiell,t in 1..tid}
anskaffelse[a,t]*materiellantall[a,m]*grunnpris[m]*(1+fordyrelse[m]/100)^t;

# Beskrankninger i antall avdelinger i strukturen (NB! minst likhet søkes)
Struktur {a in avdelinger} : avdelingsantall[a]<=sum {t in 1..tid}
anskaffelse[a,t];
```

```

# Beskrankninger i årlig budsjett
Kostnad {t in 1..tid} : sum {a in avdelinger,m in materiell}
anskaffelse[a,t]*materiellantall[a,m]*grunnpris[m]*(1+fordyrelse[m]/100)^t <=
budsjett; # Budsjett kan evt også reduseres vha faktor : *(1-red/100)^t

# Konsekvens av utsatt investering i avdeling 4 i 5 år og i materielltype 10
i 6 år (fjern #-tegn i neste to uttrykk !):
#StrLimit {a in avdelinger:a=4} : 0 >=sum {t in 1..5} anskaffelse[a,t];
#MatrLimit {m in materiell:m=10} : 0 >=sum {a in avdelinger, t in 1..6}
anskaffelse[a,t]*materiellantall[a,m];

# Database deklarasjoner
table Materiell IN "ODBC" "C:\HOTDOG\INNFAS\Ny\Innfas2.mdb" "Materiell":
materiell <- [MatID], grunnpris ~ Grunnpris, fordyrelse ~ Fordyrelse ,
materiellnavn ~ MatNavn;
table Avdelinger IN "ODBC" "C:\HOTDOG\INNFAS\Ny\Innfas2.mdb" "Avdelinger":
avdelinger <- [AvdID], avdelingsantall ~ AvdAnt, avdelingsnavn ~ AvdNavn;
table Bruk IN "ODBC" "C:\HOTDOG\INNFAS\Ny\Innfas2.mdb" "Bruk":
[Avdeling,Materiell], materiellantall ~ Antall;
table Param "ODBC" "C:\HOTDOG\INNFAS\Ny\Innfas2.mdb" "Param": [], tid ~ Tid
IN, budsjett ~ Budsjett IN, Slakk OUT;
table Fasing INOUT "ODBC" "C:\HOTDOG\INNFAS\Ny\Innfas2.mdb" "Fasing":
{avdelinger,1..tid} -> [AvdID,År], anskaffelse;

read table Materiell;
read table Avdelinger;
read table Bruk;
read table Param;

# Skriv ut data
display tid, budsjett > C:\HOTDOG\INNFAS\Ny\Innfasing.dat;
display avdelingsnavn, avdelingsantall > C:\HOTDOG\INNFAS\Ny\Innfasing.dat;
display materiellnavn, grunnpris, fordyrelse >
C:\HOTDOG\INNFAS\Ny\Innfasing.dat;
display materiellantall > C:\HOTDOG\INNFAS\Ny\Innfasing.dat;
display {a in avdelinger,m in materiell} (avdelingsnavn[a],
materiellantall[a,m]) > C:\HOTDOG\INNFAS\Ny\Innfasing.dat;

# Finn optimal MIKS innfaset løsning = 'anskaffelse[a,t]' (LP med MINOS)
solve;

let {a in avdelinger, t in 1..tid} anskaffelse[a,t]:=if anskaffelse[a,t]<0
then 0 else anskaffelse[a,t];
let Slakk:=if min(min {a in avdelinger} Struktur[a].slack , min {t in 1..tid}
Kostnad[t].slack) <-1e-3 then 1;

# Skriv ut resultater til database
write table Fasing;
write table Param;

# Skriv ut resultater på fil: årlige avdelingsanskaffelser, materiell,
materiellkostnad
display anskaffelse,anskaffelse.rc > C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
display {m in materiell, t in 1..tid} sum {a in avdelinger}
anskaffelse[a,t]*materiellantall[a,m] > C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
display {m in materiell, t in 1..tid} sum {a in avdelinger}
anskaffelse[a,t]*materiellantall[a,m]*grunnpris[m]*(1+fordyrelse[m]/100)^t >
C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;

# Skriv ut resultater på fil: årlige (avdelingsnavn, avdelingsanskaffelser),
(materiellnavn, materiell, materiellkostnad)
display {a in avdelinger, t in 1..tid}
(avdelingsnavn[a],anskaffelse[a,t],anskaffelse[a,t].rc) >
C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;

```

```
display {m in materiell, t in 1..tid} (materiellnavn[m],sum {a in avdelinger}
anskaffelse[a,t]*materiellantall[a,m]) > C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
display {m in materiell, t in 1..tid} (materiellnavn[m],sum {a in avdelinger}
anskaffelse[a,t]*materiellantall[a,m]*grunnpris[m]*(1+fordyrelse[m]/100)^t )>
C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
```

```
# Skriv ut resultater på fil: (avdelingsnavn, gjenanskaffelsekostnad, midlere
innfaset avdelingskostnad)
```

```
display {a in avdelinger} (avdelingsnavn[a],sum {m in
materiell}materiellantall[a,m]*grunnpris[m],sum {m in materiell,t in
1..tid}((anskaffelse[a,t]*materiellantall[a,m]*grunnpris[m]*(1+fordyrelse[m]/
100)^t)/avdelingsantall[a]) ) > C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
```

```
# Skriv ut resultater på fil: total innfasingskostnad, evt Slack, + evt
varsel
```

```
display f > C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
display Struktur.slack,Struktur.dual > C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
display Kostnad.slack,Kostnad.dual > C:\HOTDOG\INNFAS\Ny\Innfasing.tmp;
```

```
if min(min {a in avdelinger} Struktur[a].slack , min {t in 1..tid}
Kostnad[t].slack) <-1e-3 then {print "Tiden eller budsjettet må økes!" >
Innfasing.tmp; print "Tiden eller budsjettet må økes!";};
```

```
close;
```

```
end;
```

A.5.1 Eksempel på MIKS resultatfil

Resultatfil fra modellen MIKS angir andel anskaffet av hver avdelingstype over tid, samt tilhørende materiellmengde og kostnad. Ved kopi til Excel kan grafiske fremstillinger av avdelingsvis og materiellvis investering over tid enkelt lages.

```
# Optimal innfasing fra MIKS (utdrag):
:   avdelingsnavn[a]  anskaffelse[a,t]  anskaffelse[a,t].rc   :=
1 1   Infanteri           0           2073.44
1 2   Infanteri           0           1782.24
1 3   Infanteri           0           1500.73
1 4   Infanteri           0           1228.64
1 5   Infanteri           0           965.691
1 6   Infanteri           0           711.611
1 7   Infanteri           0           466.14
1 8   Infanteri           0           229.02
1 9   Infanteri           0.30315         0
1 10  Infanteri           1.06511         0
1 11  Infanteri           1.05003         0
1 12  Infanteri           1.03503         0
1 13  Infanteri           1.0201          0
1 14  Infanteri           1.00526         0
1 15  Infanteri           0.99049         0
1 16  Infanteri           0.975807        0
1 17  Infanteri           0.555027        0
2 1   Panser             0.412984        0
2 2   Panser             0.39334         0
2 3   Panser             0.374558        0
2 4   Panser             0.356607        0
2 5   Panser             0.339452        0
2 6   Panser             0.323063        0
2 7   Panser             0.30741         0
2 8   Panser             0.292462        0
2 9   Panser             0.200124        0
2 10  Panser             0              890.366
2 11  Panser             0              1814.44
```

2 12	Panser	0	2773.45
2 13	Panser	0	3768.65
2 14	Panser	0	4801.36
2 15	Panser	0	5872.9
2 16	Panser	0	6984.68
2 17	Panser	0	8138.13

;

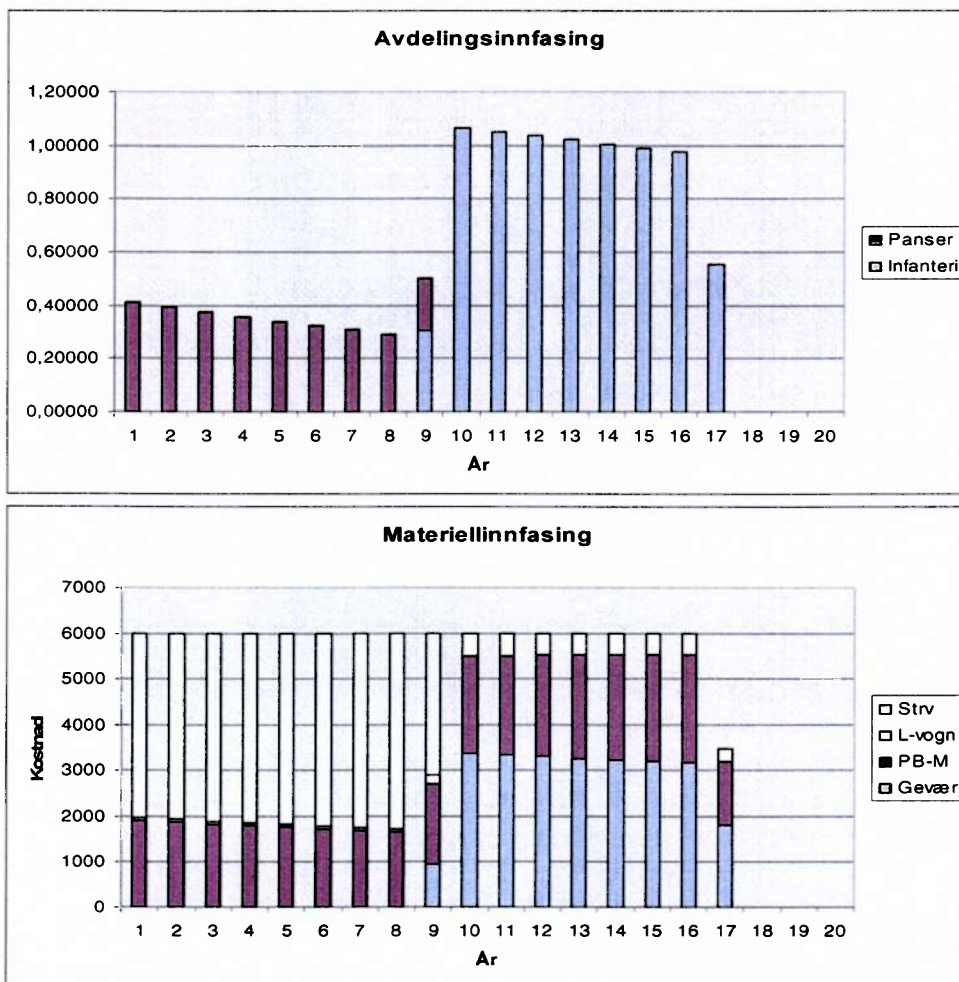
\$2 = sum{a in avdelinger} anskaffelse[a,t]*materiellantall[a,m]

: materiellnavn[m] \$2 :=

1 1	Gevær	0	
1 2	Gevær	0	
1 3	Gevær	0	
1 4	Gevær	0	
1 5	Gevær	0	
1 6	Gevær	0	
1 7	Gevær	0	
1 8	Gevær	0	
1 9	Gevær	30.315	
1 10	Gevær	106.511	
1 11	Gevær	105.003	
1 12	Gevær	103.503	
1 13	Gevær	102.01	
1 14	Gevær	100.526	
1 15	Gevær	99.049	
1 16	Gevær	97.5807	
1 17	Gevær	55.5027	
2 1	PB-M	1.23895	
2 2	PB-M	1.18002	
2 3	PB-M	1.12368	
2 4	PB-M	1.06982	
2 5	PB-M	1.01836	
2 6	PB-M	0.969189	
2 7	PB-M	0.922229	
2 8	PB-M	0.877387	
2 9	PB-M	0.903522	
2 10	PB-M	1.06511	
2 11	PB-M	1.05003	
2 12	PB-M	1.03503	
2 13	PB-M	1.0201	
2 14	PB-M	1.00526	
2 15	PB-M	0.99049	
2 16	PB-M	0.975807	
2 17	PB-M	0.555027	
3 1	L-vogn	0.825968	
3 2	L-vogn	0.78668	
3 3	L-vogn	0.749117	
3 4	L-vogn	0.713213	
3 5	L-vogn	0.678904	
3 6	L-vogn	0.646126	
3 7	L-vogn	0.614819	
3 8	L-vogn	0.584925	
3 9	L-vogn	2.21915	
3 10	L-vogn	6.39064	
3 11	L-vogn	6.30018	
3 12	L-vogn	6.21017	
3 13	L-vogn	6.12062	
3 14	L-vogn	6.03154	
3 15	L-vogn	5.94294	
3 16	L-vogn	5.85484	
3 17	L-vogn	3.33016	
4 1	Strv	1.65194	
4 2	Strv	1.57336	
4 3	Strv	1.49823	
4 4	Strv	1.42643	

4 5	Strv	1.35781
4 6	Strv	1.29225
4 7	Strv	1.22964
4 8	Strv	1.16985
4 9	Strv	0.800495
4 10	Strv	0
4 11	Strv	0
4 12	Strv	0
4 13	Strv	0
4 14	Strv	0
4 15	Strv	0
4 16	Strv	0
4 17	Strv	0;

Figuren nedenfor viser grafiske oversikter over optimal innfasing av avdelingsandeler og materiell fra modellen MIKS med den gitte struktur og under hensyn til materiellets teknologiske fordyrelse. Det ses at strukturen kan innfases i løpet av 17 år med den årlige budsjetttrammen på 6000 Mill kr som er forutsatt i eksemplet.



Figur A.4.3 Eksempel på optimal innfasing av avdelingsandeler og materiell

Litteratur



- (1) Perry W L, Moffat J (1997): Measuring the effects of knowledge in military campaigns, *Jnl of the Op. Res. Soc* **48**, 10, 965-972.
- (2) Gass N (1997): An analytical model for close combat dynamics, *Jnl of the Op Res Soc* **48**, 132-141.
- (3) Ng K Y K, Lam M N (1995): Force Deployment in a Conventional Theatre-level Military Engagement, *Jnl of the Op Res Soc* **46**, 1063-1072.
- (4) Jaiswal N K, Sangeeta Y, Gaur S C (1997): Stochastic modelling of combat with reinforcement, *Eur Jnl of Op Res (EJOPR)* **100**, 225-235.
- (5) Jaiswal N K, Sangeeta Y, Gaur S C (1995): Stochastic analysis of combat models under different termination decision rules, *Eur Jnl of Op Res (EJOPR)* **83**, 530-546.
- (6) Rowland D (1987): The Use of Historic Data in the Assessment of Combat Degradation, *Jnl of the Op Res Soc* **38**, 2, 149-162.
- (7) Williams T, Ancker C J (1981): Stochastic Duels with Displacements (Supression), , *Nav Res Log Quart* **28** , 3, September, 519-524.
- (8) Griggs B, Parnell G S, Lehmkuhl L J (1997): An Air Mission Planning Algorithm Using Decision Analysis and Mixed Integer Programming, *Op Res* **45**, 5, Sept-Oct, 662-676.
- (9) Pugh P G (1992): Lanchester Revisited, DOAE Memorandum 92104, UK..
- (10) Sundfør H (1998): Feltetterretning i et felles informasjonssystem på divisjonsnivå, FFI/RAPPORT-97/02365, Forsvarets Forskningsinstitutt, BEGRENSET
- (11) Sundfør H (1999): Mulige representasjoner av kommando og kontroll i MUNGO, FFI/NOTAT-99/03227, Forsvarets Forskningsinstitutt
- (12) Aronsveen L E (1982): Optimal allokering av flyressurser (Hovedoppgave ved NTH), FFI/NOTAT-82/5001, Forsvarets forskningsinstitutt , BEGRENSET
- (13) Weys R J-B (1989): The Aggregation Principle in Scenario Analysis and Stochastic Optimization In: *NATO ASI Series Algorithms and Model Formulations in Math Prog* (Eds S Wallace), Springer-Verlag , Berlin F51, 91-113.
- (14) Shubik M, Weber R J (1980): Systems Defense Games : Colonel Blotto, Command and Control, *Nav Res Log Quart*, 281-287.
- (15) Johnson C R, Loane E P (1975): Evaluation of Force Structures under Uncertainty, *Nav Res Log Quart*, 511-519.
- (16) Kleinrock L (1976): Queuing Systems Volume II: Computer Applications , Wiley, New York.

- (17) Gryffenberg I et al. (1997): Guns or Butter: Decision Support for Determining the Size and Shape of the South African National Defense Force, *Interfaces* **27**, 1 (Jan-Feb) , 7-28.
- (18) Ng K Y K, Lam M N (1994): Composition of the Canadian Total Force: Issues and Findings for the Air Components, *Jnl Opl Res Soc* **45**, 2, 151-164.
- (19) Dare D P (1975): The NATO Deployment Model In: *Military Strategy and Tactics* (Eds R K Huber, L F Jones, E Reine), Plenum Press, New York, 243-258.
- (20) Dirim Z U (1995): Land Forces Equipment and Munitions Expenditure Model, STC TM-926, Shape Techn. Centre, The Hague, NATO Unclassified..
- (21) Bock J C, Robinson S M (1995): Scenario analysis via bundle decomposition, *Ann. of Op. Res* **56**, 39-63.
- (22) Washburn A, Wood K (1995): Two-person Zero-sum Games for Network Interdiction, *Op. Res* **43**, 2, 243-251.
- (23) Sundfør H O (1997): Effektivitet av kommando- og kontrollsystem, FFI/RAPPORT-97/02366, Forsvarets forskningsinstitutt, Begrenset
- (24) Bertsimas D, Patterson S S (1998): The Air Traffic Flow Problem with Enroute Capacities, *Op. Res* **46**, 3, 406-422.
- (25) Mulvey J M, Vanderbei J, Zenios S A (1995): Robust optimization of large-scale systems, *Op Res* **43**, 2, 264-281.
- (26) Eppen G D, Martin R K, Schrage L (1989): A scenario approach to capacity planning, *Op Res* **37**, 4, 517-527.
- (27) Chun J B (1992): Scenario analysis modeling and Decomposition methods for Optimization under uncertainty, Ph. D. thesis, Univ of Wisconsin-Madison, Bell & Howell, Ann Arbor Mich, USA.
- (28) Bradley J M, Arntzen B C (1999): The simultaneous planning of production, capacity, and inventory in seasonal demand environments, *OP Res* **47**, 6, 795-806.
- (29) Darlington J, Pantelides C C, Rustem B, Tanyi B A (2000): Decreasing the sensitivity of open-loop optimal solutions in decision making under uncertainty, *Eur Jnl of Op Res (EJOPR)* **121**, 343-362.
- (30) Moffat J (2000): Representing the command and control process in simulation models of conflict, *Jnl of the Op Res Soc* **51**, 431-439.
- (31) Waltz E (1990): Multisensor Data Fusion, Artech House, Norwood, Ma, USA.
- (32) Brown R H (1963): Theory of Combat : The Probability of Winning, *Op Res* **11**, 418-425.
- (33) Dathe H M (1967): Some Extensions of Stochastic Combat Analysis , NATO Conf on OP Res - Recent Developments in Lanchester Theory, München, Germany, 3-7 July.
- (34) Fletcher R (1987): Practical Methods of Optimization (2nd ed), J Wiley, UK.

- (35) Schoen F (1999): Global optimization methods for high-dimensional problems, *Eur Jnl of Op Res (EJOPR)* 119, 345-352.
- (36) Nash S G, Sofer A (1996): *Linear and Nonlinear Programming*, McGraw-Hill, N.Y, USA.
- (37) Grotmol Ø, Braathen S (2000): **Modell for flerscenario strukturoptimering - Brukerveiledning**, FFI/NOTAT-2000/04743
- (38) Fourer R, Gay D M, Kernighan B W (1993): *AMPL - A Modeling Language for Mathematical Programming*, Boyd & Fraser, Mass. , USA.
- (39) Halck O M (2000): **STRIDSLOGG 2.0 : Et hjelpeverktøy for stridsanalyser og kartspill - Beskrivelse og brukerveiledning**, FFI/NOTAT-99/03062
- (40) Karlsen R, Steder F B, Korfu-Pedersen N (2000): **Kostnadsberegning av FS2000-strukturen**, FFI/RAPPORT-2000/03428, Begrenset

FORDELINGSLISTE

FFISYS
Dato: 30 november 2000

RAPPORTTYPE (KRYSS AV) <input checked="" type="checkbox"/> RAPP <input type="checkbox"/> NOTAT <input type="checkbox"/> RR		RAPPORT NR. 2000/04739	REFERANSE FFISYS/734-161.2	RAPPORTENS DATO 30 november 2000
RAPPORTENS BESKYTTELSESGRAD BEGRENSET		ANTALL EKS UTSTEDT 35	ANTALL SIDER 91	
RAPPORTENS TITTEL (U) MODELL FOR FLERSCENARIO STRUKTUROPTIMERING - Modellbeskrivelse		FORFATTER(E) BRAATHEN Sverre, GROTMOL Øyvind, LANGSÆTER Tor		
FORDELING GODKJENT AV FORSKNINGSSJEF: 		FORDELING GODKJENT AV AVDELINGSSJEF: 		

EKSTERN FORDELING
INTERN FORDELING

ANTALL	EKS NR	TIL	ANTALL	EKS NR	TIL
1		FO/SENST	2		FFI-Bibl
1		FO/HST	1		Adm direktør/stabssjef
1		FSTS	1		FFIE
1		FSTS/H	7		FFISYS
			1		FFIBM
			1		Ragnvald H Solstrand, FFISYS
			1		Bent Erik Bakken, FFISYS
			1		Jan Erik Torp, FFISYS
			1		Bjørn Tallak Bakken, FFISYS
			1		Sverre Braathen, FFISYS
			1		Fredrik Andreas Dahl, FFISYS
			1		Bård Eggereide, FFISYS
			1		Kristian Ekroll, FFISYS
			1		Geir Enemo, FFISYS
			1		Ole Martin Halck, FFISYS
			1		Marius Halsør, FFIBM
			1		Øyvind Karlsrud, FFISYS
			1		Tor Langsæter, FFISYS
			1		Terje Nilsen, FFISYS
			1		Jonny Otterlei, FFISYS
			1		Tor-Erik, Schjelderup, FFISYS
			1		Ole-Jakob Sendstad, FFISYS
			1		Hans Olav Sundfør, FFISYS
			1		Einar Østevold, FFIBM
					FFI-veven

FFI-K1

Retningslinjer for fordeling og forsendelse er gitt i Oraklet, Bind I, Bestemmelser om publikasjoner for Forsvarets forskningsinstitutt, pkt 2 og 5. Benytt ny side om nødvendig.

FORSVARETS FORSKNINGSINSTITUTT

Fakta om FFI:

FFI er et forvaltningsorgan under Forsvarsdepartementet og har til formål å drive forskning og utvikling for Forsvarets formål. Instituttet er rådgiver for Forsvarets politiske og militære ledelse i faglige spørsmål.

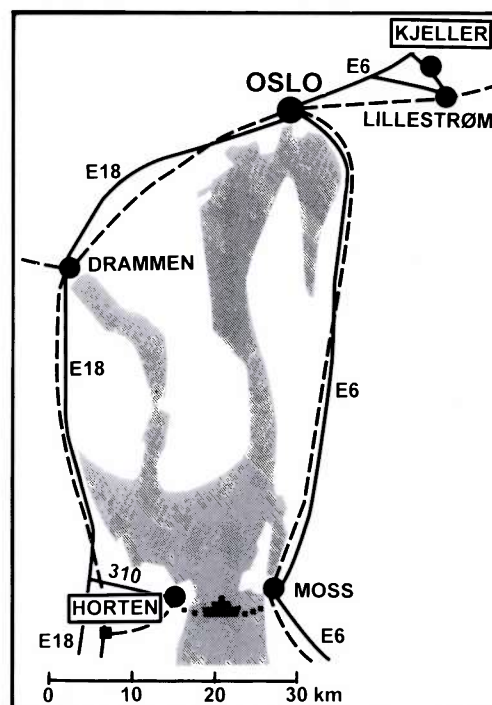
FFI er et tverrfaglig institutt som representerer de fleste ingeniørfag samt fysikk, biologi, medisin, økonomi og statsvitenskap. Instituttet er i aktivt samarbeid med ledende institusjoner og industri i inn- og utland. FFI har ialt ca. 500 medarbeidere og fem fagavdelinger.

Publikasjoner fra FFI:

FFI utgir hvert år en rekke teknisk/vitenskapelige publikasjoner som dekker ulike sider av virksomheten. Oversikt over ugraderete publikasjoner kan bestilles direkte fra FFIs bibliotek som også kan gi nærmere opplysninger.

Telefonnr.: 63 80 71 28

Telefaxnr.: 63 80 71 59



AVGRADERT
Dato: 2/3-16 Sign: SH

BEGRENSET
iht Sikkerhetsinstruksen

FORSVARETS FORSKNING SINSTITUTT

Instituttstaben
Avdeling for systemanalyse
Avdeling for beskyttelse og materiell
Avdeling for elektronikk

Postboks 25
2027 Kjeller

Telefon: 63 80 70 00
Telefax: 63 80 71 15