

Arbeidsdokument 51614

Oslo 23.06.2020

4885 Trenklin3.2

Aino Ukkonen, Stefan Flügel

Videreutvikling av Trenklin (versjon 3.2): Tilpasning av trengselsfunksjoner og valg av antall togsett per avgang

Innhold

1	Innledning	2
1.1	Bakgrunn.....	2
1.2	Om Trenklin.....	2
2	Tilpasning av trengselsfunksjoner	3
2.1	Trengselsfunksjoner i versjon 3.1	3
2.2	Nye trengselsfunksjoner	3
2.3	Implementering av nye trengselsfunksjoner.....	4
2.4	Uttesting.....	5
3	Valg av antall togsett per avgang	6
3.1	Togsett i versjon 3.1	6
3.2	Ny funksjonalitet	7
3.3	Nytt brukergrensesnitt.....	9
3.4	Endringer i VBA.....	11
3.5	Uttesting.....	12
3.6	Tips til brukere.....	16
4	Referanser	18
	Vedlegg: Oversikt over alle nye linjer i programkoden	19
	Modul: A__Parametere	19
	Modul: A__Variabler	20
	Modul: A_Redimensjonering.....	20
	Modul: A_MODELL.....	21
	Modul: B_Inndata	22
	Modul: C_RUTETABELLER.....	22
	Modul: E_NYE_ALTERNATIVER.....	25
	Modul: E1_OPPDATERER_ALTERNATIVER	26
	Modul: H6_Trengsel.....	35
	Modul: V_FUNKSJONER	39
	Modul: I__Resultatutskrift	39
	Modul: ZZZ_Settbruk.....	43

1 Innledning

1.1 Bakgrunn

Jernbanedirektoratet har bedt TØI om å gjennomføre en videreutvikling av Trenklin modellen. Prosjektet er en del av rammeavtalen Konsulentteneste innenfor samfunnsøkonomi, statistikk, transport- og markedsanalyse, nærmere bestemt modul 7, Utvikle og bruke andre/nye transportmodeller – persontrafikk.

Prosjektet går ut på å lage en ny versjon av Trenklin (versjon 3.2.) som skal inneholde følgende forbedringer:

- 1) Trengselsfunksjoner skal splittes opp i to segmenter: tjenestereiser og private reiser (arbeidsreiser og øvrige reiser). Parameterne til funksjonene skal være lik verdiene som anbefalt i den nye verdsetningsstudien (Flügel mfl 2020).
- 2) Det skal tillegges en opsjon i modellen, som aktivert utløser at Trenklin selv foreslår bruk av antall sett per avgang basert på predikert trengsel.

Oppdraget inneholder uttesting av den nye modellversjonen.

1.2 Om Trenklin

Trenklin er en persontransportmodell for togreiser som ble internt utviklet i Jernbanedirektoratet (på den tid Jernbaneverket) fra år 2013. Trenklin brukes i hovedsak for å analyse effekter av ruteplanendringer.

Styrken til modellen er den detaljerte modellering av rutevalget og trengselsnivå. Sentrale adferdsparametere inkluderer tidsverdier for ulike reisehensikter og trengselsfunksjoner. Trengselsfunksjoner definerer hvordan tidsverdien øker med økende belegg om bord. Belegg beregnes i Trenklin per avgang og snitt. Beleggsgraden vil også avhenge av togmateriell, som i dagens versjon bestemmes eksogent. Per i dag er Trenklin den eneste operative modellen i Norge som kan beregne trafikantnyttens av redusert trengsel ombord.

Første versjon er dokumentert i Aarhaug mfl. (2013) mens foreløpig siste offentlige tilgjengelige dokumentasjon omhandler versjon 3.0 (Ranheim 2017).¹

Vi fikk også tilsendt rapportutkast til versjon 3.1. (Ranheim og Høyem 2020). Versjon 3.1 inneholder en del ny funksjonalitet, i hovedsak rettet mot analyser for Flytoget. Det er også blitt endring i måten man spesifiserer parameterne på. I versjon 3.1 legges disse direkte i excel-arket (og ikke i VBA-koden som i tidligere versjoner).

Dette arbeidsnotatet forklarer de nye endringene og elementene i versjon 3.2. Vi går ut fra at leseren har en viss forståelse for hvordan Trenklin fungerer. For en utdypende modellbeskrivelse viser vi til rapportene nevnt over.

¹ Utover det er versjon 1.17 dokumentert i Caspersen mfl (2014) og versjon 2.7 er omtalt og diskutert i Flügel og Hulleberg (2017)

2 Tilpasning av trengselsfunksjoner

2.1 Trengselsfunksjoner i versjon 3.1

Trengselsfunksjoner defineres i versjon 3.1 basert på 6 parametere (Figur 1). I versjon 3.1 er disse felles for de tre reisehensiktene fritidsreise, arbeidsreise og tjenestereise.

19	Trengselsfunksjon			
20	F_sit	1	tall (single)	fastledd sittende
21	F_stå	1.646	tall (single)	fastledd stående
22	Mu_sit	0.5	tall (single)	beleggsgrad hvor trengsel begynner å inntre
23	F_sit_ingenledig	1.219	tall (single)	tidsverdivekt ved ingen ledige sitteplasser
24	Sit_u_stå	0.0991	tall (single)	påslag i tidsverdivekt per stående per kvm, sittende
25	Stå_u_stå	0.0769	tall (single)	påslag i tidsverdivekt per stående per kvm, sittende
26				

Figur 1: Utklipp fra arket «Parametere» i Versjon 3.1 der trengselsfunksjoner defineres

Merk at F_sit settes per definisjon til verdi 1. Denne relateres til tidsverdien uten trengsel. Tidsverdien spesifiseres i versjon 3.1. i arket «Segmentmodeller».

2.2 Nye trengselsfunksjoner

Den nye Verdsettingsstudien 2018/2019 har anbefalt nye trengselsfunksjoner (Flügel mfl 2020). Trengselsfunksjonene er estimert basert på valgekspesimenter fra kollektivbrukene i Oslo, Trondheim og Molde. Tabell 1 viser de anbefalte verdier.

Tabell 1: Anbefalte parametere for trengselsfunksjonen

Parameter i TRENKLIN		Fritid og arbeid	Tjenestereiser
F_sit	Faktor på tidsverdi uten trengsel	1	1
Mu_sit	Andel sitteplasser opptatt når trengsel inntre	0,5	0,5
F_sit_ingenledig	Faktor for sittende når 100 % av sitteplassene er opptatt	1,219	1,044
Sit_u_stå	Marginal effekt av én person mer per kvadratmeter for sittende	0,0769	0,0154
F_stå	Faktor for stående før det oppstår trengsel	1,646	1,646
Stå_u_stå	Marginal effekt av én person mer per kvadratmeter for stående	0,0991	0,0198

Parameterverdier for Sit_u_stå og Stå_u_stå er nedjustert for tjenestereiser. Bakgrunnen for denne nedjusteringen er at tidsverdien for tjenestereiser inneholder et betydelig bidrag fra arbeidsgiverne. Denne andelen forventes ikke å være påvirket av trengselsulempen (se mer detaljer i Flügel et al 2020).

Merk at $F_{sit} = 1$ tilsvarer tidsverdi uten trengsel. Tidsverdi uten trengsel er lavere enn enhetsverdien for tidsverdien, siden denne implisitt tilsvarer et gjennomsnittlig trengselsnivå.

Tabell 2 viser tidsverdi uten trengsel i 2018-kroner som anbefalt i verdsettingsstudien.

Tabell 2: Anbefalte tidsverdier for togreiser uten trengsel (2018-kroner)

Reiseformål	Under 70 km	70-200 km	Over 200 km
Tjenestereise	436	376	410
Til/fra arbeid	85	151	216
Fritidsreiser	81	105	138
Alle formål	89	139	179

De nye tidsverdiene har ikke vært implementert i versjon 3.1, men er nå med i versjon 3.2. I modelltestingen er det brukt tidsverdier for togreiser under 70 km.

2.3 Implementering av nye trengselsfunksjoner

I den nye versjonen 3.2 får tjenestereisene egne parametere, mens trengselsfunksjonene for fritids- og arbeidsreiser ikke endres fra versjon 3.1.

Parameterne i trengselsfunksjonen defineres i inndata-arket Parametere. I den nye modellversjonen er det lagt til et til sett med inndata for tjenestereiser. Disse er navngitt på samme måte som for fritid og arbeid, men markert med «_tje», som vist i Figur 2.

Trengselsfunksjon (fritid, arbeid)			
F_{sit}	1	tall (single)	fastledd sittende
$F_{stå}$	1,646	tall (single)	fastledd stående
Mu_{sit}	0,5	tall (single)	beleggsgrad hvor trengsel begynner å inntre
$F_{sit_ingenledig}$	1,219	tall (single)	tidsverdivekt ved ingen ledige sitteplasser
$Sit_u_stå$	0,0769	tall (single)	påslag i tidsverdivekt per stående per kvm, stående
$Stå_u_stå$	0,0991	tall (single)	påslag i tidsverdivekt per stående per kvm, sittende
Trengselsfunksjon (tjeneste)			
F_{sit_tje}	1	tall (single)	fastledd sittende
$F_{stå_tje}$	1,646	tall (single)	fastledd stående
Mu_{sit_tje}	0,5	tall (single)	beleggsgrad hvor trengsel begynner å inntre
$F_{sit_ingenledig_tje}$	1,044	tall (single)	tidsverdivekt ved ingen ledige sitteplasser
$Sit_u_stå_tje$	0,0154	tall (single)	påslag i tidsverdivekt per stående per kvm, stående
$Stå_u_stå_tje$	0,0198	tall (single)	påslag i tidsverdivekt per stående per kvm, sittende

Figur 2: Utklipp fra arket «Parametere» i Versjon 3.2 der trengselsfunksjoner defineres

For å kunne kjøre modellen med hensiktsspesifikke trengselsfunksjoner er noen deler av koden endret. I modulen H6_trengsel, der trengselsfunksjonene beregnes, er det lagt til en løkke over hensikter og to «if»-setninger sånn at trengselskostnaden beregnes for hver hensikt. Den totale trengselskostnaden blir beregnet som en vektet sum av trengselskostnadene for de forskjellige hensiktene. Vektene som blir brukt er belegg pr. hensikt, delt på det totale belegget. På den måten unngår man dobbelttelling i løkken.

I modulene E_NYE_ALTERNATIVER og E1_OPPDATER_ALTERNATIVER beregnes også trengselskostnader. I disse er det allerede kodet inn en løkke over hensiktene, og dermed er det kun lagt til hensiktsspesifikke verdier, sånn at trengselskostnadene blir beregnet med verdiene i Figur 2.

Vi viser til vedlegget for en fullstendig oversikt over kode-deler som er endret.

2.4 Uttesting

Endringene som er forklart i kapittel 2.3 ble først testet med opprinnelige tidsverdier som lå i versjon 3.1 (se avsnitt 2.1). Gjennom å bruke trengselsparameterne for arbeid og fritid også i boksen for tjenestereiser i arket Parametere, verifiserte vi at endringene i kode ga samme resultat som versjon 3.1.

Vi har også gjort noen tester for den store Østlandsmodellen til å se på effekten av å bruke de nye parameterne i trengselsfunksjoner for tjenestereiser. Tabell 3 viser resultat fra eksempelberegning med Østlandsmodellen, for referanse og tiltak. Beregningene er gjort med nye tidsverdier.

Tabell 3: Effekten av nedjustering av trengselskostnader for tjenestereiser. Kroner per virkedøgn.

Parametere i trengselsfunksjonen	Trengselskostnader referanse	Trengselskostnader tiltak	Trafikantnytte
Like for alle reisehensikter (tall fra avsnitt 2.1)	3 100 216	3 070 449	3 103 020
Med nedjustering for tjenestereiser (tall fra avsnitt 2.2)	2 042 797	1 996 766	3 026 515

Vi ser fra eksempelberegningen med Østlandsmodellen i tabell 3 at de nye trengselsparameterne gjør at trengselskostnadene reduseres med 34 prosent når man sammenligner med å bruke samme parametere for alle hensikter. Trafikantnyttet reduseres med 2 prosent når vi bruker nye hensiktsfordelte trengselsfunksjoner. Vi ser at det altså er en effekt i totale trengselskostnader og trafikantnytte når trengselskostnadene blir nedjustert.

Det er verdt å legge merke til at trengselskostnadene reduseres mer enn andelen til tjenestereiser skulle tilsi. Dette skyldes at tidsverdien for tjenestereiser (Tabell 2) er veldig høy i forhold til tidsverdien for de andre reisehensiktene.

3 Valg av antall togsett per avgang

I dette kapittel omtales implementering av opsjon for valg av antall togsett i versjon 3.2.

Formål med opsjonen er å tilpasse antall togsett for hver avgang, gitt beregnet belegg. Ved bruk av en slik opsjon kan man redusere trengsel ved å tilføre sett på avganger med høy trengsel og fjerne sett fra avganger med veldig lite trengsel.

3.1 Togsett i versjon 3.1

I versjon 3.1 er antall sett per avgang statiske inndata og forblir uendret for referanse og tiltak gjennom en gitt modellkjøring.

Togsettene er definert i arkfanen Togmateriell, kolonne D (se eksempel i figur 3). I tillegg til antall sett er også seteantall (kolonne B) og kvm ståareal (kolonne C) definert i samme arkfane. Kolonne B het før «Seter pr sett» men dette er rettet opp siden kolonnen egentlig inneholder seter pr tog.

	A	B	C	D	E	F
1		Seter	Kvm ståareal	Antall sett	Kolliplass	Diesel
2	69-3	300	68,1	1	40	0
3	69-4	408	82,8	1	40	0
4	69-5	504	109,5	1	40	0
5	69-6	600	165,6	2	40	0
6	69G-3	242	50	1	40	0
7	70-4	230	50,1	1	80	0
8	70-8	460	100,2	2	40	0
9	72-4	306	44,8	1	80	0
10	72-8	612	89,6	2	40	0
11	73-4	1000	500	1	80	0
12	73-8	1000	500	2	40	0
13	74-5	210	71,8	1	80	0
14	74-10	420	143,6	2	120	0
15	74-15	630	215,4	3	40	0
16	75-5	259	83,8	1	80	0
17	75-10	518	167,6	2	40	0
18	73B-4	243	39,3	1	80	0
19	73B-8	486	78,6	2	80	0
20	B3-8	516	103,2	2	40	0
21	B7-4	210	21	1	40	0
22	B7-6	326	32,6	1	40	0
23	B7-7	394	39,4	1	40	0
24	B7W-5	218	21,8	1	40	0
25	B7W-7	306	30,6	1	40	0
26	Buss-1	1000	100	1	40	0
27	Y1-1	69	6,9	1	40	0
28	S-tog-5	300	93,75	1	80	0
29	S-tog-10	600	187,5	2	40	0
30	69-1	300	68,1	1	40	0
31	69-2	600	165,6	2	40	0
32	70-1	230	50,1	1	80	0
33	70-2	460	100,2	2	40	0

Figur 3: Togmateriell i Trenklin versjon 3.1.

3.2 Ny funksjonalitet

Modellbrukeren gis valgmuligheten til å aktivere en opsjon som beregner optimalt antall sett pr avgang. Hvis brukeren ikke velger å aktivere opsjonen blir modellen kjørt likt som i Trenklin versjon 3.1. Hvis brukeren velger å aktivere opsjonen så er det tre muligheter for hvor optimalt antall sett skal beregnes: for referanse, for tiltak eller for begge. Tilpasningen til optimalt antall sett gjøres først og fremst for å minimere trengsel på avganger med høyt belegg. I dette arbeidsdokument og i modellberegningene er belegg definert som andelen opptatte plasser (både sitte- og ståplasser) i forhold til totalkapasiteten pr avgang, der totalkapasiteten er summen av antall sitteplasser og antall ståplasser.

For å beregne det optimale antallet sett for hver avgang blir modellen kjørt flere ganger til man når ønsket minimum beleggsgrad. Denne iterasjonen over den vanlige modellen (omtrent lik som tidligere versjon) gjør at metodikken for modellkjøringen ikke er vesentlig endret, men at det er lagt til funksjonalitet ved å kjøre modellen flere ganger. I tillegg til itereringen er det lagt til en modul der settbruk blir beregnet. Denne modulen kjøres på slutten av hver iterasjon, før resultatutskriftene.

Den nye modellversjonen 3.2 har følgende funksjonalitet.

1. Hvis bruker har valgt settberegning for tiltak og/eller referanse blir en løkke aktivert. Denne løkken kjører et maksimum antall iterasjoner eller til resultatet ikke gir endringer i settbruk. Hvis bruker ikke velger settberegning blir maksimalt antall iterasjoner automatisk satt til 1.
2. Kjører modellen som i tidligere versjon, med små endringer i forhold til forrige modellversjon (se vedlegg kap. 5).
3. Hvis settberegning er valgt, aktiveres modulen for settbruk som omfordeler, legger til eller fjerner sett per avgang. For en beskrivelse av modulen, se Tabell 4.
4. Hvis antall sett som skal omfordeles er null så avsluttes itereringen over modellen. Hvis settbruk ikke er valgt så avsluttes løkken her etter første iterasjon. Ellers oppdateres settantallet i tråd med det som ble beregnet i punkt 3. og blir inndata i neste iterasjon. Deretter begynner kjøringen igjen i punkt 1.

Modellbrukeren gir rammene for settberegningen. Brukeren kan velge om settberegning skal gjøres for referanse, tiltak eller begge. Hvis brukeren velger at settberegningen skal gjøres for både referanse og tiltak, blir iterasjonene over modellen avsluttet først når antall omfordelte sett er mindre enn 1 både for referanse og tiltak.

Modellbrukeren velger også om antall sett i modellkjøringen skal holdes konstant, om det er en øvre grense for antall nye sett man kan tilføre eller om man kan tilføre ubegrenset antall nye sett i modellområdet. Dette for å gi mulighet til å beregne anbefalt settantall selv om budsjettet kan legge begrensninger på antall nye sett som kan kjøpes inn.

En øvre grense for høy trengsel defineres også av modellbrukeren. Denne må angi et maksimumbelegg b_{maks} , der avganger med belegg høyere enn dette nivået får tildelt et nytt sett (gjelder ikke avganger som allerede kjører 3 sett eller hvor det å kjøre med 3 sett ikke er tillatt). På tilsvarende måte kan bruker også gi en grense for minimumbelegg, b_{min} . Hvis brukeren har valgt å tilføre ubegrenset antall nye sett, blir alle avganger med belegg under b_{min} trukket fra et sett. Hvis det er valgt å holde antall sett konstant eller

hvis det er valgt et maksimum antall nye sett, blir denne nedre grensen irrelevant siden antall sett som trekkes fra blir beregnet fra antall sett som blir lagt til avganger med for høye belegg.

Alternativt kan bruker også velge å ikke legge til eller fjerne sett, hvilket er ekvivalent med $b_{maks} = 100\%$ eller $b_{min} = 0\%$.

Hvis det trengs å begrense antall sett pr linje, for eksempel på grunn av kort plattform på noen stasjoner, kan bruker velge å definere makssett for hver linje.

Tabell 4 viser valgene som brukeren kan gjøre i modellkjøringen.

Tabell 4: Opsjoner i settberegning

Indeks	Spørsmål	Svarmulighet	Opplegg
1	Bruk fast antall togsett i referanse?	Ja/nei	Hvis «Ja» kjøres standardmodellen. Hvis «Nei» så aktiveres settbrukmodulen for referansealternativet.
1	Bruk fast antall togsett i tiltak?	Ja/nei	Hvis «Ja» kjøres standardmodellen. Hvis «Nei» så aktiveres settbrukmodulen for tiltaksalternativet.
2	Begrensing for antall nye sett?	1. Hold antall sett konstant 2. Ingen begrensning 3. Maks. antall nye sett	Hvis 1. så holdes antall sett konstant gjennom alle iterasjoner. Hvis 2. er det ingen øvre grense på hvor mange sett som kan legges til ved hver iterasjon. Hvis 3. så legges maks. antall sett til første iterasjon og er tilgjengelig gjennom hele kjøringen.
3	Minimum beleggsgrad der det fjernes ett sett fra en avgang?	1. Ikke fjerne sett fra avganger 2. b_{min} (et prosenttall)	Hvis 1. så fjernes det ikke sett fra noen avganger. Hvis 2. så definerer bruker en nedre grense for belegg.
4	Maksimum beleggsgrad der man tilfører ett sett til en avgang?	1. Ikke tilføre sett fra avganger 2. b_{maks} (et prosenttall)	Hvis 1. så blir det ikke lagt til sett på noen avganger. Hvis 2. så definerer bruker en øvre grense for belegg der det tilføres sett.
5	Begrensing for antall sett på linje?	Ja / nei	Hvis «Ja» må bruker definere maks. antall sett som kan brukes for hver avgang. Hvis «nei» så er maks. antall sett for alle linjer 3.

Merk at ikke alle kombinasjoner av valgmuligheter nødvendigvis gir mening, og noen kombinasjoner av valg kan lede til at modellen ikke kjører som ønsket. Eksempel på dette er hvis brukeren velger 1 både på spørsmål 3 og på spørsmål 4, det vil si hvis ikke det er mulig å fjerne eller tilføre sett på enkelte avganger, så kan man ikke beregne et optimalt antall sett for avgangene.

I alle beregninger er det tatt hensyn til at antall sett ikke kan være mindre enn 1 eller over 3.

Ttabell 5 viser hvordan modulen for settberegninger er satt opp.

Tabell 5: Modul for settberegning.

- a. For hver avgang identifiseres maksimal beleggsggrad.
- b. Avgangene i 1. rangeres etter maksimal beleggsggrad.
- c. For alle avganger der belegg er større enn maksbelegg b_{maks} og der settantall er 1 eller 2 fra før blir det lagt til sett, gitt at settantallet ikke overstiger mulige begrensninger for maks. antall sett pr linje (spørsmål 5 i tabell 4). Vi betegner det totale antallet sett som er lagt til med $a_{b,maks}$.
- d. Det er valgt et av alternativene i spørsmål 2 (tabell 4).
 - d.1. Hvis bruker har valgt 2.1 *Hold antall sett konstant* blir $a_{b,maks}$ antall sett trukket fra de avganger med laveste belegg. Her blir altså antall sett omfordelt.
 - d.2. Hvis bruker har valgt 2.2 *Ingen begrensning* blir sett fjernet fra alle avganger med belegg mindre enn b_{min} .
 - d.3. Hvis bruker har valgt 2.3 *Maks antall nye sett* og fylt i en tallverdi for antall nye sett, betegner vi dette tallet med $a_{nyesett}$. Da er antall sett som blir fjernet fra avganger med lavest belegg $a_{b,maks} - a_{nyesett}$. Hvis $a_{b,maks} - a_{nyesett}$ er negativt, blir ikke sett fjernet fra avganger.
Hvis antallet nye sett $a_{nyesett}$ allerede er «brukt» i forrige iterasjon er de da en del av settantallet i beregningen. Da blir sett pr avgang omfordelt slik som i punkt d.1.
Hvis antall sett ikke blir endret så stoppes løkken over modellen og resultater blir skrevet ut.
- e. Avganger med oppdatert settantall blir lagret som en variabel som brukes til å oppdatere inndata i neste iterasjon.

Som punkt d.3. i tabellen viser, så blir altså nye sett brukt opp først, før man starter en eventuell omfordeling. Dette er en prioritering som brukes i modellen, nye sett blir prioritert fremfor å omfordele.

I neste iterasjon er det ingen forskjell om et sett er nytt eller omfordelt. Det brukes alltid antall sitteplasser og ståplasser som et sett har i det toget settet blir tilført. Ordet «omfordele» skal derfor ikke tolkes som en fysisk omfordeling av enkelte sett.

Hvis bruker har valgt å beregne settbruk både for tiltak og referanse, blir løkken over modellen avsluttet først når både referanse og tiltak har oppnådd ønsket belegg eller når maksgrense for antall iterasjoner blir oppnådd.

3.3 Nytt brukergrensesnitt

Valgene som brukeren gjør i tabell 4 er lagt inn i Trenklinmodellens brukergrensesnitt i Excel i arkfanen Parametere. Her kan brukeren velge Sann/Usann (True/False) eller gi tallverdier der det er mulig. De nye radene i brukergrensesnittet er vist i figur 4.

	A	B	C	D
55	minimum byttetakst	40	tall	minste kostnad ved bytte av takstsystem
56				
57	Settbruk			
58	Bruk fast antall togsett referanse	False	"True"/"False"	hvis "False" foreslår modellen antall sett pr avgang basert på predikert trengsel
59	Bruk fast antall togsett tiltak	False	"True"/"False"	hvis "False" foreslår modellen antall sett pr avgang basert på predikert trengsel
60	<i>hvis "false" fyll i spørsmålen under</i>			
61	Begrensning for antall sett pr. linje			
62	1. hold antall sett konstant	False	"True"/"False"	Hvis "False", velg maks antall nye sett eller ingen begrensning
63	2. ingen begrensning	False	"True"/"False"	Hvis "False", velg maks antall nye sett. Hvis True, velg beleggsgard i B67 og B70
64	3. maks. antall nye sett	10	positive heltall	antall nye sett, >0
65	Minimum beleggsgard* der det fjernes sett			
66	1. Ikke fjern sett	False	"True"/"False"	Hvis "false", velg når sett skal fjernes
67	2. Fjern sett når beleggsgard er over	10	prosenttall	fjerner sett med beleggsgard mindre enn tallet her
68	Maksimum beleggsgard* der det tilføres sett			
69	1. Ikke tilfør sett	False	"True"/"False"	Hvis "false", velg når sett skal legges til
70	2. Tilfør sett når beleggsgard er over	50	prosenttall	legger til sett med beleggsgard over dette tallet, denne skal ikke være lavere enn B67
71	Begrensning for antall sett pr. avgang			
72	1. Ingen begrensning	True	"True"/"False"	Hvis "false", må maks. antall togsett defineres i arkfanan "forutsetninger"
73				
74	*beleggsgard defineres som andel av maksimumkapasitet, der maksimumkapasitet = antall stående og antall sittende som får plass på toget			
75				
76				

Figur 4: Brukerbegrensning i arkfanen Parametere

Som vist i figuren over er det her settberegningen kan aktiveres ved å velge False enten for referanse, tiltak eller begge (cellene B58 og B59). Deretter spesifiserer man hvilke krav som skal gjelde ved beregningen (cellene B62-B72). Disse kravene gjelder automatisk både referanse og tiltak, hvis begge er valgt.

I cellene B62-B64 (Figur 4) presiserer man hvorvidt beregningen blir gjort med nye sett eller med eksisterende sett. Hvis beregningen skal gjøres med eksisterende sett velger man «True» i B62. Hvis beregningen skal gjøres med ubegrenset antall nye sett, velg «False» i B62 og «True» i B63. Hvis beregningen skal gjøres med et begrenset antall nye sett, skrives antallet inn i B64. Her er det viktig å vite at kun en av cellene B62-B63 kan ha verdien «True». Begge kan ha verdien «False», og da må man spesifisere tallverdien i B64.

I cellene B66-B67 kan man velge å ikke fjerne sett eller å spesifisere en grense for minste belegg. For at tallet i B67 skal leses inn må B66 være satt til «False». Tilsvarende gjelder for B69-B70. Verdien i celle B70 er viktig for at resultatene fra modellkjøringen med settbruk skal gi mening.

Hvis celle B72 er satt til Usann (False), må maksimalt antall togsett pr linje presiseres i arkfanen Forutsetninger. For dette er det lagt til to nye kolonner, D og H, i arkfanen Forutsetninger, vist i Figur 5.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10	Liste Over Linjer									Liste over togprodukter m
11	Referanse	print	maks. antall togsett	Tiltak	print	maks. antall togsett	Nummer	Navn		
12	L2	1 Oslo S		3 L2	1 Oslo S	3		1 NSB		
13								2 Flytoget		
14										
15										
16										
17										
18								1 NSB		
19								2 Flytoget		
20										
21										
22										
23										
24								1 NSB		

Figur 5: Brukergransnitt, nye kolonner for maks. antall sett (kolonnene D og H)

I tillegg til endringene i arkfanene Parametere og Forutsetninger så er det lagt til tre nye arkfaner, *Sett_iterasjoner*, *Sett_iterasjoner_t* og *Sett_iter_oppsummering*. I det første arket blir resultater skrevet ut fra settberegning for referanse og i det andre blir tiltaksresultater skrevet ut. I fanen *Sett_iter_oppsummering* oppsummeres antallet iterasjoner og man får en oversikt over hvor mange avganger som har fått endret settantall ved hver iterasjon.

3.4 Endringer i VBA

Alle endringer og tillegg i koden og den nye modulen er markert med aktuelt versjonsnummer #3.2.1. Modulen der settberegningene blir gjort heter «ZZZ_Settbbruk» og blir kalt inn fra modulen «A_MODELL». Løkken som er nevnt i forrige kapittel er kodet inn i modulen «A_MODELL».

Målet var å inngripe så lite som mulig i eksisterende kode og istedenfor kode inn tillegg. Allikevel måtte noen små endringer gjøres. En fullstendig oversikt over endret kode er vist i vedleggskapittel 5.

3.5 Uttesting

Uttesting i løpet av modellutviklingen ble først gjort med en mindre testmodell med kun én toglinje og til slutt med Østlandsmodellen. Den mindre modellen ble brukt på grunn av lang beregningstid for Østlandsmodellen.

Resultater fra beregninger med Østlandsmodellen er presentert i Tabell 6. Testingen er gjort med de nye tidsverdiene for reiser under 70 km (Tabell 2) og med forskjellige kombinasjoner av inndataparametere, for å illustrere effekten av begrenset antall sett og ubegrenset bruk av nye sett.

Tabell 6: Resultater fra beregning med Østlandsmodellen. Kroner per virkemote.

Scenario	Trengselskostnader	Trengselskostnader	Trafikantnytte
	referanse	tiltak	
Referansekjøring. Settbruk av.	2 042 797	1 996 766	3 026 515
Fast antall sett. [10,50]*			
Settbruk på: referanse og tiltak	831 601	917 175	2 736 850
Ingen begrensning. [10,40]**			
Settbruk på: referanse og tiltak	696 001	650 762	2 862 569

*[10,50] er belegginnndata fra bruker, $b_{min} = 10$ prosent og $b_{maks} = 50$ prosent.

** [10,40] tilsvarer $b_{min} = 10$ prosent og $b_{maks} = 40$ prosent

Vi ser i Tabell 6 at trengselskostnadene med settberegning reduseres vesentlig. Dette viser hvor mye trengselskostnadene kan bli redusert. Når vi sammenligner første og andre rad ser vi at allerede omfordeling av sett har stor innvirkning på trengselskostnadene, dette gir en reduksjon på hhv 59 og 54 prosent for referanse og tiltak.

Tabell 7 viser antall sett som ble omfordelt ved hver iterasjon for referanse og tiltak i modellkjøringen uten begrensning på antall nye sett (dvs. den siste av kjøringene i tabell 6). Tabellen viser også hvor mange iterasjoner som trengtes for å redusere trengselen. Tilsvarende tabell vises også i arket *Sett_iter_oppsummering* i brukergrensesnittet til modellen.

Tabell 7: Antall sett som ble lagt til, fjernet eller omfordelt i tiltak og referanse. Modellkjøring med settbruk, ingen begrensning på antall nye sett, minste belegg 10 % og største belegg 40 %.

Iterasjon	Referanse	Tiltak
1	378	294
2	29	24
3	0	0

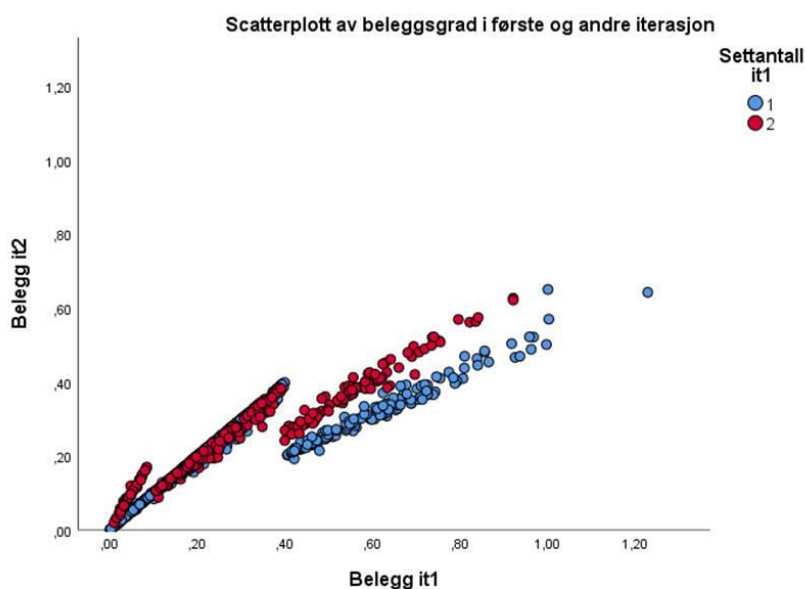
Vi ser av Tabell 7 at det trengtes tre iterasjoner for å oppnå ønsket beleggsnivå på alle avganger der det var mulig og hensiktsmessig å legge til eller fjerne sett. Tabellen viser også at antall sett som omfordeles i første iterasjon er vesentlig høyere enn i andre og at modellen oppnår en likevekt etter få iterasjoner.

For samme kjøring viser Figur 6 og 7 scatterplott av beleggsgrad på avgangsnivå mellom henholdsvis første og andre, og andre og tredje (her siste) iterasjon.² Figurene er laget i SPSS ut fra data i Excel-arket *Sett_iterasjoner_t* (gjelder altså tiltaket).

I andre iterasjon ser vi at alle avganger som hadde et belegg på over 0,4 får lavere beleggsgrad i andre iterasjon. Det er åpenbart at disse avganger fikk et ekstra sett i andre iterasjon. Vi ser at effekten på beleggsgrad er relativt større for avgangersom kjørte med ett sett i iterasjon 1 (blå sirkler), sammenlignet med de som kjørte med 2 sett (røde sirkler).

Avganger med en beleggsgrad mellom 0,1 og 0,4 har uendret antall sett i andre iterasjon. Vi ser at disse stort sett har konstant beleggsgrad. Mindre endringer (typisk reduksjoner) i beleggsgrad kan tilbakeføres til endring i rutevalget (avganger som fikk ett sett mer attraherer flere passasjer pga redusert trengselsnivå).

Effekten på avganger med belegg på mindre enn 0,1 i iterasjon 1 er todelt. De som kjørte med to sett får trukket fra ett sett, mens avganger med ett sett kjører med ett sett også i iterasjon 2.



Figur 6: Scatterplott av beleggsgrad ved overgang fra første til andre iterasjon (farger indikerer antall sett i første iterasjon, der blå betyr ett sett og rød to sett). Modellkjøring uten begrensning på antall nye sett, minste belegg 10 % og største belegg 40 %. Hver sirkel representerer én togangang i Østlandsmodellen.

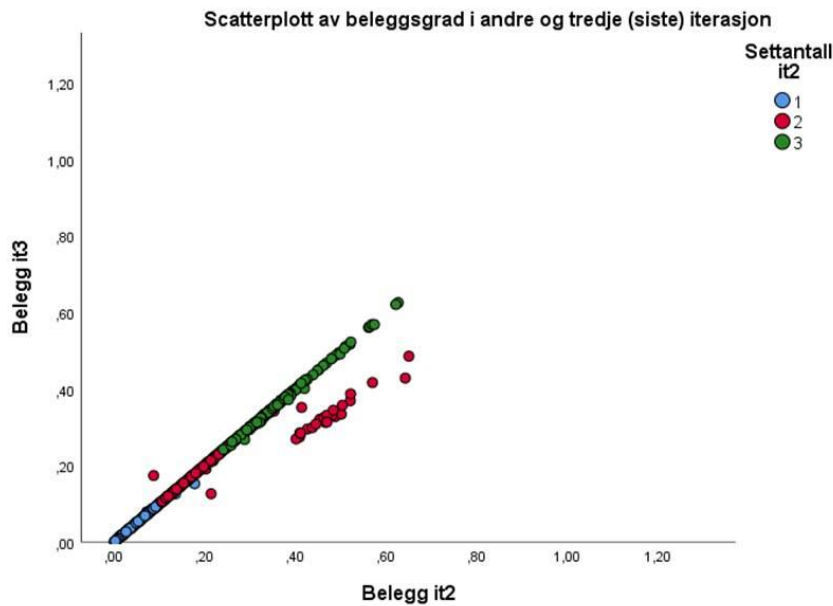
I tredje iterasjon får de avganger med 2 sett som ligger over 0,4 i andre iterasjon 1 nytt sett (og får redusert beleggsgrad (Figur 7). I denne testkjøringen har alle avganger lov til å bruke tre sett (dette må justeres i realistiske eksempler).

De avganger som allerede hadde 3 sett i andre iterasjon (grønne punkter) beholder 3 sett og har tilnærmet lik beleggsgrad i iterasjon 3.

De fleste punkter i Figur 7 ligger på 45 graderslinjen, noe som tyder på et relativt konstant rutevalg.

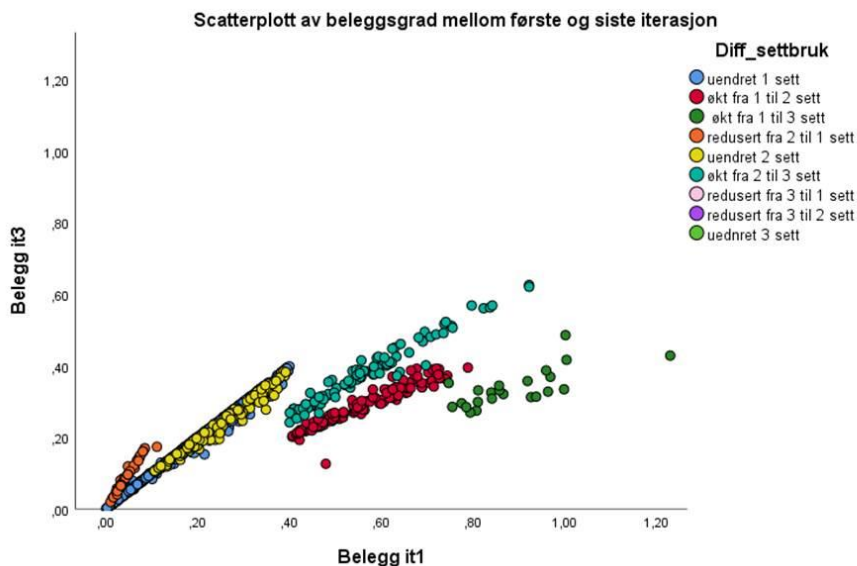
² Det er to observasjoner med en beleggsgrad større enn 1 i resultatfilen. Spesielt én observasjon med belegg på over 1,2 skiller jeg ut. Dette er den første avgangen på døgnet og det ser ut til at «etterspørselen» ble presset inn i denne avgangen. Dette skal i prinsippet unngås med variabelen «Beleggsgrad stående sperre». At vi likevel observerer belegg større enn 1 kan tyde på at det lå noen feil/svakheter i implementeringen av denne variabelen i versjon 3.1

Det ser ut til at én avgang med 2 sett i iterasjon 2 mister ett sett til iterasjon 3 (se den røde sirkelen som ligger over 45-grad-linjen).



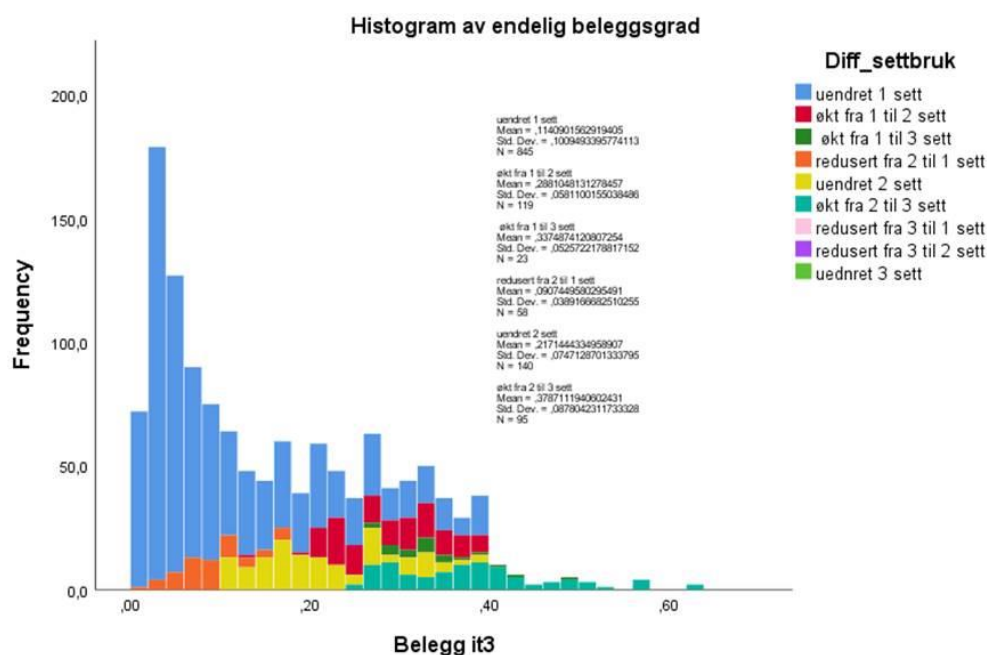
Figur 7: Scatterplott av beleggsgrad ved overgang fra andre til tredje iterasjon (fargene indikerer antall sett i andre iterasjon). Modellkjøring med settbruk, ingen begrensning på antall nye sett, minste belegg 10 % og største belegg 40 %. Hver sirkel representerer én togavgang i Østlandsmodellen.

Figur 8 sammenligner opprinnelig og endelig beleggsgrad. Fargene viser endring i settbruk. De tre siste fargene (rosa, lilla og lysgrønn) har ingen sirkler siden det er ingen avganger som kjørte med 3 sett i iterasjon 1.



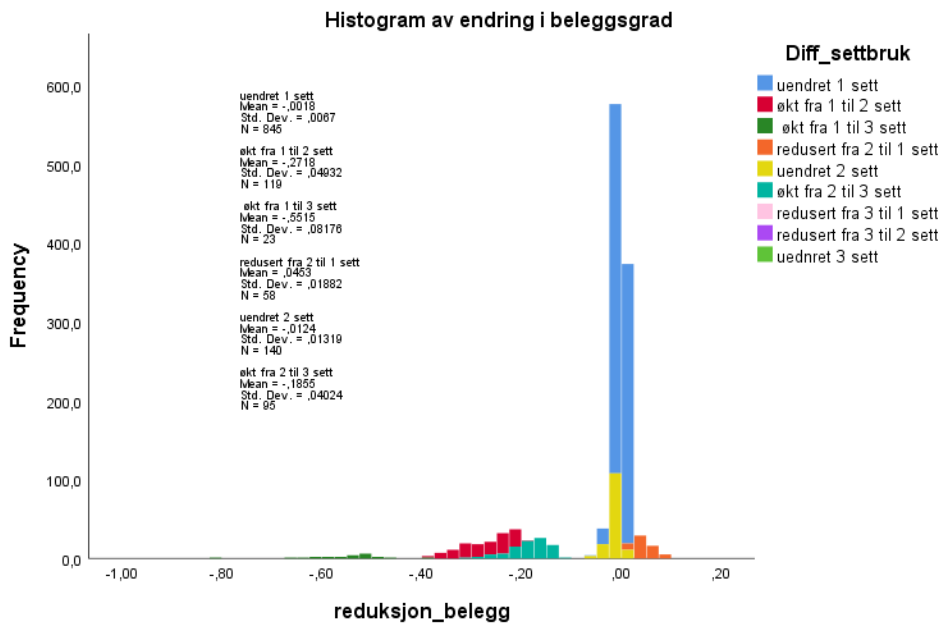
Figur 8: Scatterplott av beleggsgrad mellom første og tredje iterasjon (fargene indikerer endring i antall sett). Modellkjøring uten begrensning på antall nye sett, minste belegg 10 % og største belegg 40 %. Hver sirkel representerer én togavgang i Østlandsmodellen.

Vi ser at 23 avganger økte fra 1 til 3 sett. Dette er dog en relativ liten andel av alle avganger. Dette er også vist i Figur 9 som er et histogram med endelig beleggsgard. Her ser vi også at kun avganger som allerede kjører med 3 sett har belegg på over 0,4.



Figur 9: Histogram av endelig beleggsgard (fargene indikerer endring i antall sett). Modellkjøring uten begrensning på antall nye sett, minste belegg 10 % og største belegg 40 %.

Til slutt viser Figur 10 fordeling av endring i settbruk. Fra denne figuren ser vi at de fleste avganger har tilnærmet uendret beleggsgard (og dermed trengsel). For avgangen med ett ekstra sett ligger reduksjonen typisk rundt 15-40 prosentpoeng. At nivået i nedgang er noe lavere enn man ville forvente av en ren økning i antall sett skyldes rutevalgseffekter (redusert trengsel pga økt kapasitet på avganger attraherer flere reisende og det fører til en liten «rebound» effekt på trengselsnivå/beleggsgard).



Figur 9: Histogram over reduksjonen i beleggsgrad i prosentpoeng (fargene indikerer endring i antall sett). Modellkjøring uten begrensning på antall nye sett, minste belegg 10 % og største belegg 40 %.

3.6 Tips til brukere

Til slutt nevner vi noen tips for brukerne av Trenklin som har tenkt å anvende opsjonen for valg av antall sett.

Hvis man velger å kjøre settbruk med fast antall togsett eller med et gitt antall nye sett, er det mulig at resultatene (sett pr avgang) etter noen iterasjoner begynner å oscillere/svinge fra iterasjon til iterasjon. Dette tyder på at modellen har vanskeligheter med å finne en stabil løsning (likevekt) gitt forutsetningene. Hvis man ønsker å unngå dette kan man for eksempel velge å legge til flere sett.

Siden settberegningen er implementert som iterasjoner med full modellkjøring pr iterasjon, økes beregningstiden med antall iterasjoner. Hvis for eksempel det kreves 4 iterasjoner for å avslutte settberegningen så er beregningstiden 4 ganger lenger enn for en beregning uten settbruk. Settberegninger kan derfor ta lang tid hvis modellområdet er stort og modellen i utgangspunktet tar lang tid å kjøre.

Når man aktiverer opsjonen for settbruk i referanse er det anbefalt å også gjøre det i tiltak; dette for å unngå misvisende resultater for trafikantnytt. I noen tilfeller vil det være meningsfullt å aktivere opsjonen i tiltak men ikke i referansen. I så fall er det viktig å ta inn over seg at beregnet trafikantnytte innebærer en forbedret bruk av togsett eller bruk av flere togsett. I nytte-kostnads-sammenheng vil det da være viktig å ta med eventuelle kostnader forbundet med omrokking og/eller innkjøp av togsett.

Hvis settberegningsoptionen er aktivert er det mulig å følge med omfordelingen av sett i Immediate-vinduet. I dette vinduet vises avganger rangert etter belegg, og de er indeksert etter belegg, ikke avgang. Avganger som er over maksbelegg eller under

minimumbelegg vises i vinduet, og der ser man også hvor mange sett avgangene har før og etter settberegningen.

Det er også viktig å vite at kolliberegningen (Ranheim og Høyem 2020) ikke er kompatibel med settberegningen, det vil si at hvis man velger settberegning så gir resultatene for kolliberegning ikke mening. Dermed lønner det seg å skru av kolliberegning før man tar i bruk settberegningene eller ignorere resultatene fra kolliberegningen.

Til slutt en kort anmerkning om tolkning av resultater. Resultatene representerer den «optimale» allokeringen av togsett under de gitte forutsetninger (antall nye sett, maksimal belegggrad ...). Man skal ikke tolke resultatene som den bedriftsøkonomiske (eller samfunnsøkonomiske) optimale løsningen. For senere modellversjoner kunne man tenke å videreutvikle metodikken slik at investerings- og driftskostnader, billettinntekter og/eller trafikanntytte styrer graden av omfordelingen av togsett.

4 Referanser

Aarhaug J., Caspersen E., Fearnley N., Ramjerdi F., Ranheim P., Steinsland C. (2013) *Dokumentasjonsrapport: Inkrementelle etterspørselsmodell*, TØI rapport 1283/2013

Caspersen E., Ranheim P., Aarhaug J. (2014) *Dokumentasjonsrapport: Trenklin versjon 1.17*, TØI rapport 1341/2014

Flügel, S. & Hulleberg, N. (2016). *Trenklin 2 – Gjennomgang av modellen og drøfting av anvendelsesområde*. TØI rapport 1534/2016.

Flügel S., Halse A.H., Hulleberg N., Jordbakke G.N., Veisten K., Sundfør H.B., & Kouwenhoven M. (2020). *Verdsetting av reisetid og tidsavhengige faktorer. Dokumentasjonsrapport til Verdsettingsstudien 2018-2019*. TØI-rapport 1762/2020

Ranheim, P. (2017) *Trenklin versjon 3 - Dokumentasjon og brukerveiledning*, Jernbanedirektoratet

Ranheim P. & Høyem H. (2020) *Trenklin 3.1, Dokumentasjon av utvikling, Urbanet Analyse*

Vedlegg: Oversikt over alle nye linjer i programkoden

Modul: A__Parametere

```
'Settbruk*****  
  
*****  
***** '#3.2.1 (AUK) Her styres bruk av modulen for foreslått settbruk basert på predikert trengsel  
  
SETTBRUKAV_REF = P_ind1(58)           'Hvis True kjører modellen uten foreslått settbruk i referanse, hvis False aktiveres modul for settbruk  
  
SETTBRUKAV_TILTAK = P_ind1(59)       'Hvis True kjører modellen uten foreslått settbruk i tiltak, hvis False aktiveres modul for settbruk  
  
'Følgende gjelder for referanse og tiltak:  
  
SETTBRUK_KONST = P_ind1(62)         'Hvis True holdes antall sett konstant  
  
SETTBRUK_IKKEBEGRENSET = P_ind1(63) 'Hvis True er det ingen begrensning på antall nye sett  
  
SETTBRUK_SETTMAX = P_ind1(64)       'Maksimum antall nye sett  
  
MINBELEGG_IKKEFJERN = P_ind1(66)    'Hvis True fjernes ikke sett  
  
MINBELEGG_FJERN = P_ind1(67)       'Angi minimum beleggsggrad når sett skal fjernes  
  
MAXBELEGG_IKKETILFOR = P_ind1(69)  'Hvis True legges det ikke til sett  
  
MAXBELEGG_TILFOR = P_ind1(70)      'Maksimum belegg når sett tilføres  
  
SETTBRUK_MAXPRAVGANG = P_ind1(72)  'Bruk maksimumsett pr. avgang definert i arkefanen Forutsetninger  
  
  
  
'#3.2.1 (AUK) alle indekser i P_ind1(n) herfra er oppdatert til n+7 siden det er 7 nye rader for tjenestereiser i Parametere-arket  
  
  
  
'Kapasitet-----  
-----  
  
STÅPLASSER_PERKVM = P_ind1(34) 'til resultatutskrift for kapasitetsutnyttelse. Ved utregning av total-kapasitet vil denne legges til grunn
```

Modul: A__Variabler

```
Public Settantall_avgang() As Variant 'settantal avgang #3.2.1 (AUK)
```

```
Public Maxbeleggrad_avg() As Variant 'maksbelegg pr avgang #3.2.1 (AUK)
```

```
'ZZZ_settbruk #3.2.1 (AUK)
```

```
'Public sett_nye_ref As Double 'maks. antall nye sett
```

```
'Public sett_nye_tiltak As Double 'maks. antall nye sett
```

```
Public sett_banken_ref As Double 'antall sett som overføres fra en iterasjon til neste
```

```
Public sett_banken_tiltak As Double 'antall sett som overføres fra en iterasjon til neste
```

```
Public Sett_SumEndret_Iterasjon_ref() As Variant 'Sum over antall sett som er fjernet/tilført lagres her
```

```
Public Sett_SumEndret_Iterasjon_tiltak() As Variant 'Sum over antall sett som er fjernet/tilført lagres her
```

```
Public Setekapasitet_prsett() As Variant 'variabel der kapasiteten lagres til neste iterasjon
```

```
Public Ståareal_prsett() As Variant 'variabel der kapasiteten lagres til neste iterasjon
```

```
Public Sett_Endret_Iterasjon() As Variant 'sett lagres til neste iterasjon
```

```
Public antall_avganger_tot_ref As Double 'totalt antall avganger
```

```
Public antall_avganger_tot_tiltak As Double 'totalt antall avganger
```

```
Public Sett() As Variant 'settantall pr. avgang, oppdateres for hver iterasjon
```

Modul: A_Redimensjonering

```
ReDim Maks_Togsett_Linje(1 To 2, 1 To Antall_Linjer) As Long '#3.2.1 (AUK)
```

```
ReDim Settantall_avgang(1 To 2, 1 To Antall_Linjer, 1 To 2) As Variant '#3.2.1 (AUK)
```

```
'ZZZ_settbruk #3.2.1 (AUK)
```

```
ReDim Sett_SumEndret_Iterasjon_ref(1 To sett_iter_maxgrense) As Variant
```

```

ReDim Sett_SumEndret_Iterasjon_tiltak(1 To sett_iter_maxgrense) As Variant

If sett_iter = 1 Then

    ReDim Setekapasitet_prsett(1 To 2, 1 To Antall_Linjer, 1 To 2) As Variant

    ReDim Ståareal_prsett(1 To 2, 1 To Antall_Linjer, 1 To 2) As Variant

    ReDim Sett_Endret_Iterasjon(1 To 2, 1 To Antall_Linjer, 1 To 2) As Variant

End If

```

Modul: A_MODELL

```

' _____ HER STYRES ITERASJONER OVER SETTBRUK #3.2.1 (AUK) _____
'
iterasjonslutt = False 'begynnelsesverdi, oppdateres i ZZZ_settbruk

|

Dim sett_iter_max As Double

If SETTBRUKAV_REF = True And SETTBRUKAV_TILTAK = True Then

    sett_iter_max = 1 'hvis settbruksberegningen ikke skal brukes

Else

    sett_iter_max = sett_iter_maxgrense 'her styres maksimalt antall iterasj
    oner for å stoppe kjøringene der likevekt ikke oppnås

End If

|

For sett_iter = 1 To sett_iter_max

Debug.Print "**** Starter iterasjon nr " & sett_iter & " over Trenklin ****"

' _____

If BEREGN_MED_KOLLI = True Then 'hvilken modul som velges avhenger av om det sk
al beregnes med kolli eller ikke

    Call F1a_beleggtrengsel_kolli.beleggtrengsel_kolli 'beregne
r belegg og trengsel i nettverket med kolli

    Debug.Print "OBSERVER FEIL! Kolliberegning er ikke kompatib
el med settberegning, velg heller FALSE i Parametere\B37" '#3.2.1 (AUK) settbru
ken er ikke implementert kolliberegningene

Else

```

Modul: B_Inndata

```
'#3.2.1 (AUK) leser inn maks. antall togsett pr linje  
  
For q = 1 To 2  
    For i = 1 To Antall_Linjer  
        Maks_Togsett_Linje(q, i) = Workbooks(wbName).Worksheets("Forutsetninger  
").Cells(11 + i, 4 + (q - 1) * 4)  
    Next i  
Next q
```

Modul: C_RUTETABELLER

```
'Leser inn egenskaper per avgang - setekapasitet, ståkapasitet og plass  
til kolli  
  
Dim setvekt() As Long, ståvekt() As Long, settant() As Long, kolvekt() As Long,  
diesel() As Byte '##104 leser også inn kolli her nå, #3.2.1 (AUK) leser også i  
nn sett  
  
For k = 1 To Antall_LinjerZ(z)  
    For l = 1 To 2  
        ReDim ståvekt(1 To a(k, l, z)) As Long 'V#  
        ReDim setvekt(1 To a(k, l, z)) As Long 'V#  
        ReDim settant(1 To a(k, l, z)) As Long '#3.2.1 (AUK)  
        ReDim kolvekt(1 To a(k, l, z)) As Long 'V#  
        ReDim diesel(1 To a(k, l, z)) As Byte 'V#  
        For p = 1 To a(k, l, z)  
            setvekt(p) = Togegenskaper(GetIndex(Togtyper, Materielltype(z,  
k, l)(p)), 1)  
            ståvekt(p) = Togegenskaper(GetIndex(Togtyper, Materielltype(z,  
k, l)(p)), 2)
```

```

        settant(p) = Togeenskaper(GetIndex(Togtyper, Materielltype(z,
k, l)(p)), 3) '#3.2.1 (AUK)

        kolvekt(p) = Togeenskaper(GetIndex(Togtyper, Materielltype(z,
k, l)(p)), 4) '##104

        diesel(p) = Togeenskaper(GetIndex(Togtyper, Materielltype(z, k
, l)(p)), 5) '##104

    Next p

    Ståareal_avgang(z, k, l) = ståvekt

    Setekapasitet_avgang(z, k, l) = setvekt

    Settantall_avgang(z, k, l) = settant '#3.2.1 (AUK)

    kolliplass_avgang(z, k, l) = kolvekt '##104

    diesel_avgang(z, k, l) = diesel

Next l

Next k

Erase setvekt()

Erase ståvekt()

Erase settant()

Erase kolvekt()

Erase diesel()

' Her oppdateres setekapasitet og ståkapasitet for iterasjoner etter den første
iterasjonen #3.2.1 (AUK)

' Variabeln Sett_Endret_Iterasjon viser hvor mange sett som er trukket fra/lagt
til i forhold til FØRSTE iterasjon

If sett_iter > 1 Then 'iterasjoner over Trenklin

    If z = 1 Then 'referanse

        If SETTBROKAV_REF = False Then 'hvis referansekjøring er valgt

            For k = 1 To Antall_LinjerZ(z)

                For l = 1 To 2

```

```

        For p = 1 To a(k, l, z)
            Ståareal_avgang(z, k, l)(p) = Ståareal_avgang(z, k, l)(
p) + (Sett_Endret_Iterasjon(z, k, l)(p)) * Ståareal_prsett(z, k, l)(p)
            Setekapasitet_avgang(z, k, l)(p) = Setekapasitet_avgang
(z, k, l)(p) + (Sett_Endret_Iterasjon(z, k, l)(p)) * Setekapasitet_prsett(z, k,
l)(p)
            Settantall_avgang(z, k, l)(p) = Settantall_avgang(z, k,
l)(p) + Sett_Endret_Iterasjon(z, k, l)(p)
            'kolli og diesel?
        Next p
    Next l
Next k
End If
ElseIf z = 2 Then 'tiltak
    If SETTBROKAV_TILTAK = False Then 'hvis tiltakskjøring er valgt
        For k = 1 To Antall_LinjerZ(z)
            For l = 1 To 2
                For p = 1 To a(k, l, z)
                    Ståareal_avgang(z, k, l)(p) = Ståareal_avgang(z, k, l)(
p) + (Sett_Endret_Iterasjon(z, k, l)(p)) * Ståareal_prsett(z, k, l)(p)
                    Setekapasitet_avgang(z, k, l)(p) = Setekapasitet_avgang
(z, k, l)(p) + (Sett_Endret_Iterasjon(z, k, l)(p)) * Setekapasitet_prsett(z, k,
l)(p)
                    Settantall_avgang(z, k, l)(p) = Settantall_avgang(z, k,
l)(p) + Sett_Endret_Iterasjon(z, k, l)(p)
                    'kolli og diesel?
                Next p
            Next l
        Next k
    End If
End If
End If

```



```
If x = 1 Then
    MsgBox "Sjekk feilmeldinger for ruteplaner, rett opp og kjør på nytt"
End If
```

Modul: E_NYE_ALTERNATIVER

```
'#3.2.1 (AUK)redefinierer hensiktsavhengige trengselfunksjonsparametere
```

```
Dim f_stå_h() As Double
```

```
ReDim f_stå_h(1 To HH) As Double
```

```
f_stå_h(1) = F_STÅ
```

```
f_stå_h(2) = F_STÅ
```

```
f_stå_h(3) = F_STÅ_TJE
```

```
Dim f_sit_h() As Double
```

```
ReDim f_sit_h(1 To HH) As Double
```

```
f_sit_h(1) = F_SIT
```

```
f_sit_h(2) = F_SIT
```

```
f_sit_h(3) = F_SIT_TJE
```

```
Dim mu_sit_h() As Double
```

```
ReDim mu_sit_h(1 To HH) As Double
```

```
mu_sit_h(1) = MU_SIT
```

```
mu_sit_h(2) = MU_SIT
```

```
mu_sit_h(3) = MU_SIT_TJE
```

```
Dim f_sit_ingenledig_h() As Double
```

```
ReDim f_sit_ingenledig_h(1 To HH) As Double
```

```
f_sit_ingenledig_h(1) = F_SIT_INGENLEDIG
```

```
f_sit_ingenledig_h(2) = F_SIT_INGENLEDIG
```

```
f_sit_ingenledig_h(3) = F_SIT_INGENLEDIG_TJE
```

```
Dim sit_u_stå_h() As Double
```

```
ReDim sit_u_stå_h(1 To HH) As Double
```

```
sit_u_stå_h(1) = SIT_U_STÅ
```

```
sit_u_stå_h(2) = SIT_U_STÅ
```

```
sit_u_stå_h(3) = SIT_U_STÅ_TJE
```

```
Dim stå_u_stå_h() As Double
```

```
ReDim stå_u_stå_h(1 To HH) As Double
```

```
stå_u_stå_h(1) = STÅ_U_STÅ
```

```
stå_u_stå_h(2) = STÅ_U_STÅ
```

```
stå_u_stå_h(3) = STÅ_U_STÅ_TJE
```

Modul: E1_OPPDATERER_ALTERNATIVER

```
'-----  
'
```

```
'#3.2.1 (AUK)redefinierer hensiktsavhengige trengselfunksjonsparametere
```

```
Dim f_stå_h() As Double
```

```
ReDim f_stå_h(1 To HH) As Double
```

```
f_stå_h(1) = F_STÅ
```

```
f_stå_h(2) = F_STÅ
```

```
f_stå_h(3) = F_STÅ_TJE
```

```
Dim f_sit_h() As Double
```

```
ReDim f_sit_h(1 To HH) As Double
```

```
f_sit_h(1) = F_SIT
```

```
f_sit_h(2) = F_SIT
```

```
f_sit_h(3) = F_SIT_TJE
```

```
Dim mu_sit_h() As Double
```

```
ReDim mu_sit_h(1 To HH) As Double
```

```
mu_sit_h(1) = MU_SIT
```

```
mu_sit_h(2) = MU_SIT
```

```
mu_sit_h(3) = MU_SIT_TJE
```

```
Dim f_sit_ingenledig_h() As Double
```

```
ReDim f_sit_ingenledig_h(1 To HH) As Double
```

```
f_sit_ingenledig_h(1) = F_SIT_INGENLEDIG
```

```
f_sit_ingenledig_h(2) = F_SIT_INGENLEDIG
```

```
f_sit_ingenledig_h(3) = F_SIT_INGENLEDIG_TJE
```

```
Dim sit_u_stå_h() As Double
```

```
ReDim sit_u_stå_h(1 To HH) As Double
```

```
sit_u_stå_h(1) = SIT_U_STÅ
```

```
sit_u_stå_h(2) = SIT_U_STÅ
```

```
sit_u_stå_h(3) = SIT_U_STÅ_TJE
```

```
Dim stå_u_stå_h() As Double
```

```
ReDim stå_u_stå_h(1 To HH) As Double
```

```
stå_u_stå_h(1) = STÅ_U_STÅ
```

```
stå_u_stå_h(2) = STÅ_U_STÅ
```

```
stå_u_stå_h(3) = STÅ_U_STÅ_TJE
```

```
'-----  
'-----
```

```
Dim p_sitte() As Double 'sannsynlighet for å ta sitteplass ved påstigningsnode
```

```

Dim nypåstigning_path() As Byte '#101), denne kan brukes til takst også

Dim beleggsgrad As Single

Dim stående As Single

Dim ledig_gulvareal As Single

'Oppdaterer allerede funnede alternativer

For k = 1 To HH 'dette skal gjøres for alle hensikter (fritid og forreting påvi
rkes av trengsel, men tar ikke hensyn til det når de velger)

    For i = 1 To AST

        For j = 1 To AST

            If Turmatrise(1, k, i, j) > SPERRETALL And i <> j Then

                If z = 2 And i = 6 And j = 4 Then

                    i = i

                End If

                For s = 1 To seg_antall(k, Segmentmodell(i, j))

                    For n = 1 To Antall_R2(k, i, j)(s) 'R2 da nye alternativer
allerede har fått

                        'nullstiller påstigningssperrer

                        SS_påstigningssperre(k, i, j)(s)(n) = 0

                    Next n

                    'If Gjenfunnet_matrise(k, i, j)(n) = 0 Then

                        If ((k = 2 Or k = 3) And (beregner_likevekt(k, Segmen
tmodell(i, j)) = True)) Then 'fritid og og forreting OG det skal beregnes likev
ekt for dem (etter sannsynlighet for sitteplass)

```

```

'MsgBox Segmentmodell_beregn_likevekt_frifor(2)

modell(i, j)) = 1 Then '##105
    If Segmentmodell_beregn_likevekt_frifor(Segment
        påstigningen = SS_Path(k, i, j)(s)(n)(1)
        Togproduktet = Togprodukt(z, Node_linje(påstigningen), Node_retning(påstigningen))(Node_avgang(påstigningen)) '##105
        Perioden = Periode_minutt(node_påstigning(påstigningen))
        Stasjonen = node_stasjon(påstigningen)
        Retningen = Node_retning(påstigningen)

        Tidsverdi = (1 + Segmentmodell_beregn_likevekt_frifor(Segmentmodell(i, j)) * sannsynlighet_for_ståplass(Togproduktet, Perioden, Stasjonen, Retningen)) _
        * seg_tidsverdier(k, Segmentmodell(i, j))(s) / 60
        SS_trengsel(k, i, j)(s)(n) = SS_ombordtid(k, i, j)(s)(n) * (Tidsverdi - seg_tidsverdier(k, Segmentmodell(i, j))(s) / 60)
    Else
        Tidsverdi = seg_tidsverdier(k, Segmentmodell(i, j))(s) / 60 '##102
    End If
    Else 'ellers oppdateres trengsel for alle andre muligheter (men regens ikke nødvendigvis likevekt for)
        Tidsverdi = seg_tidsverdier(k, Segmentmodell(i, j))(s) / 60 '##102
        'standard prosedyre for arbeidsreiser

        ReDim p_sitte(1 To SS_nodesinpath(k, i, j)(s)(n))
        ReDim nypåstigning_path(1 To SS_nodesinpath(k, i, j)(s)(n)) As Byte

        For m = 1 To SS_nodesinpath(k, i, j)(s)(n) - 1

```

```

Node1 = SS_Path(k, i, j)(s)(n)(m)
Node2 = SS_Path2(k, i, j)(s)(n)(m)
lenken = Lenke(Node1)(Node2) '##3.1 opprydd
ing av koden

If BytteLenke(lenken) = 0 Then
    If seter_lenke(lenken) > 0 Then
        beleggsgrad = lenkebelegg2(lenken)
/ seter_lenke(lenken)
    End If
    If lenkebelegg2(lenken) > seter_lenke(l
enken) Then
        stående = lenkebelegg2(lenken) - se
ter_lenke(lenken)
    Else
        stående = 0
    End If
    ledig_gulvareal = ståareal_lenke(lenken
) * Andel_ledig_areal(lenken)
    End If
    '
    '
    '           E1_utr1 = ((F_STÅ_h(q) - 1) + ((lenkeb
elegg2(lenken) - seter_lenke(lenken)) / (ståareal_lenke(lenken) * Andel_ledig_a
real(lenken))) * STÅ_U_STÅ_h(q)) * Ombordtid(lenken)
    '
    End If
'
'
'
'
n
    If i = 1 And j = 4 And s = 1 And n = 60 The

        i = i
    End If

```

```

If m = 1 Or nypåstigning_path(m) = 1 Then
    If beleggsgrad < 1 Then 'ved påstigning
er det kun mulig med sitteplass om lenkebelegget etter påstigning er lavt nok

    'Hvis så er tilfelle er det uansett
sitteplass å få
    p_sitte(m) = 1
    Treng = Treng
    If MU_SIT < 1 Then
        If beleggsgrad > mu_sit_h(k) Th
en
            Treng = Treng + ((beleggsgr
ad - mu_sit_h(k)) / (1 - mu_sit_h(k))) * (f_sit_ingenledig_h(k) - f_sit_h(k)) *
Ombordtid(lenken)
        Else
            Treng = Treng
        End If
    End If
    Else
        If stående > Påstigende_node(Node1)
Then 'garantert ikke sitteplass
            p_sitte(m) = 0
            Treng = Treng + ((f_stå_h(k) -
1) + (stående / (ståareal_lenke(lenken) * Andel_ledig_areal(lenken))) * stå_u_s
tå_h(k)) * Ombordtid(lenken)
        Else
            p_sitte(m) = (seter_lenke(lenke
n) - (lenkebelegg2(lenken) - Påstigende_node(Node1))) / Påstigende_node(Node1)
            '##104
            Treng = Treng + p_sitte(m) * ((
f_sit_ingenledig_h(k) - f_sit_h(k)) + (stående / ledig_gulvareal) * sit_u_stå_h
(k)) * Ombordtid(lenken) + _
            (1 - p_sitt
e(m)) * ((f_stå_h(k) - 1) + (stående / ledig_gulvareal) * stå_u_stå_h(k)) * Omb
ordtid(lenken)
        End If
    End If

```

```

en
    If p_sitte(m) > 1 Or p_sitte(m) < 0 Then
        MsgBox "psitte over 1"
    End If

nypåstigning_path(m + 1) = 0

Else 'tre muligheter, enten analyseres en k
jørelenke, en byttelenke, eller en oppholdslenke

    If Byttelenke(lenken) = 1 Then
        nypåstigning_path(m + 1) = 1
        'Ingen oppdatering av trengsel
    Else
        If Node_ma(Node1) = 1 Then 'her er
det muligheter for at stående tar sittepass
            If stående < (Avstigende_node(N
ode1) - Avstigende_stående_node(Node1)) Then
                p_sitte(m) = 1
                If MU_SIT < 1 Then '##106
                    If beleggsggrad > mu_sit
_h(k) Then 'beregner trengsel på lenken for sittende
                        Treng = Treng + ((b
eleggsggrad - mu_sit_h(k)) / (1 - mu_sit_h(k))) * (f_sit_ingenledig_h(k) - f_sit
_h(k)) * Ombordtid(lenken)
                    Else
                        Treng = Treng
                    End If
                End If
            End If
        Else 'disse plassene blir tilgj
engelige
            If (stående + (Avstigende_n
ode(Node1) - Avstigende_stående_node(Node1))) > 0 Then

```



```

ode1) - Avstigende_stående_node(Node1)) / _
a1 = (Avstigende_node(N
(stående + (Avstigende_
node(Node1) - Avstigende_stående_node(Node1))) 'og så mange ønsker de plassene
Else
a1 = 0
End If
p_sitte(m) = 1 - (1 - p_sit
te(m - 1)) * (1 - a1) ' så hvis p_sitte var 1, vil den fremdeles være 1
Trenng = Trenng + p_sitte(m)
* ((f_sit_ingenledig_h(k) - f_sit_h(k)) + (stående / ledig_gulvareal) * sit_u_s
tå_h(k)) * Ombordtid(lenken) + _
(1 - p_sitt
e(m)) * ((f_stå_h(k) - 1) + (stående / ledig_gulvareal) * stå_u_stå_h(k)) * Omb
ordtid(lenken)
End If
'
' If p_sitte(m) > 1.00000001 Or
p_sitte(m) < -0.00001 Then
' MsgBox "psitte over 1"
' End If
Else 'da må det være en lenke uten
initiell mulighet for stående til å ta sitteplass (påstigningsnode) -
> ingen oppdatering av p_sitte, men trenngselsberegning
p_sitte(m) = p_sitte(m - 1)
If beleggsgrad < 1 Then
If beleggsgrad > mu_sit_h(k
) Then 'beregner trenngsel på lenken for sittende
Trenng = Trenng + ((beleg
gsgrad - mu_sit_h(k)) / (1 - mu_sit_h(k))) * (f_sit_ingenledig_h(k) - f_sit_h(k
)) * Ombordtid(lenken)
'Trenngsel(lenken) = p_s
itte(p)

```

```

Else
    Treng = Treng
End If

Else 'disse ønsk
er sitteclass...
    Treng = Treng + p_sitte(m)
* ((f_sit_ingenledig_h(k) - f_sit_h(k)) + (stående / ledig_gulvareal) * sit_u_s
tå_h(k)) * Ombordtid(lenken) + _
(1 - p_sitt
e(m)) * ((f_stå_h(k) - 1) + (stående / ledig_gulvareal) * stå_u_stå_h(k)) * Omb
ordtid(lenken)
End If

End If

End If

End If

If (m = 1) Or nypåstigning_path(m) = 1 Then
'sjekker ved påstigning

If (stående / ledig_gulvareal) > Be
leggsgrad_stående_sperre Then
    SS_påstigningssperre(k, i, j)(s
)(n) = 1
End If

End If

Next m

'##TIDSVERD
I (mertidsverdi som følge av trengsel) per min

SS_trengsel(k, i, j)(s)(n) = Treng * Tidsverdi

'
If SS_trengsel(k, i, j)(s)(N) > 0 And k = 1 Then
'
    i = i

```

```

'
                                End If
Trenng = 0
End If
'End If
Next n
Next s
End If
Next j
Next i
Next k

```

Modul: H6_Trengsel

```

'beregner cirka trengsel per avgang, linje og totalt basert på utlagt belegg
'Dim H6_avg_tidsverdi As Single (kommentert bort #3.2.1 (AUK))
Dim H6_treng As Single
Dim H6_beleggsgrad As Single
Dim H6_ståperkvm As Single
Dim H6_tidsbruk As Single

For r = 1 To HH
    For k = 1 To Antall_LinjerZ(z)
        For l = 1 To 2 'retninger
            For i = 1 To A_S(k, l, z) - 1
                For j = 1 To a(k, l, z)
                    On Error Resume Next
                    If IsNumeric(Ruteplan(z, k, l)(i * 2, j)) Then
                        If i = 1 Then
                            H6_tidsbruk = Kjøretid(z, k, l)(i, i + 1, j)

```

```

ElseIf i = A_S(k, m, z) - 1 Then
    H6_tidsbruk = Kjoretid(z, k, l)(i, i + 1, j)
Else
    H6_tidsbruk = Kjoretid(z, k, l)(i, i + 1, j) '+ Opp
holdstid(z, k, l)(i, j)
End If

If Belegg(z, k, l)(i, j) > 0 Then 'bruker tidsv
erdier til segmentmodell 1 som gjennomsnittlig tidsverdi for hovedsegmentet
    'estimat for en gjennomsnittlig tidsverdi på snitte
t
    H6_treng = 0

    '#3.2.1 (AUK) H6_avg_tidsverdi blir overflødig pga
beregning pr hensikt
    'H6_avg_tidsverdi = (Belegg_hensikt(1, z, k, l)(i,
j) * seg_tidsverdier(1, 1)(1) + _
    Belegg_hensikt(2, z, k, l)(i, j
) * seg_tidsverdier(2, 1)(1) + _
    Belegg_hensikt(3, z, k, l)(
i, j) * seg_tidsverdier(3, 1)(1)) _
    / Belegg(z, k, l)(i, j)

    'beregner trengselen:
    H6_beleggsgrad = (Belegg(z, k, l)(i, j) / Setekapas
itet_avgang(z, k, l)(j))

    '****#3.2.1 (AUK) Beregninger for forskejlige hens
ikter

    If r = 1 Or r = 2 Then 'hvis hensikt er arbeid elle
r fritid

    If (H6_beleggsgrad) < 1 Then 'ingen ledige sitt
eplasser, men trengsel kan ha inntreidd

```

```

                                If H6_beleggsgrad > MU_SIT Then

                                    H6_treng = H6_treng + ((H6_beleggsgrad
- MU_SIT) / (1 - MU_SIT)) * (F_SIT_INGENLEDIG - F_SIT) * H6_tidsbruk * seg_tids
verdier(r, 1)(1)

                                End If

                                Else

                                    H6_ståperkvm = (Belegg(z, k, l)(i, j) - Set
ekapasitet_avgang(z, k, l)(j)) / Ståareal_avgang(z, k, l)(j)

                                    'disse sitter

                                    H6_treng = H6_treng + (F_SIT_INGENLEDIG + H
6_ståperkvm * SIT_U_STÅ - F_SIT) * Setekapasitet_avgang(z, k, l)(j) * H6_tidsbr
uk * seg_tidsverdier(r, 1)(1)

                                    ....|

                                    '|disse står...

                                    H6_treng = H6_treng + (F_STÅ + H6_ståperkvm
* STÅ_U_STÅ - F_SIT) * (Belegg(z, k, l)(i, j) - Setekapasitet_avgang(z, k, l)(
j)) * H6_tidsbruk * seg_tidsverdier(r, 1)(1)

                                End If

                                Else 'hvis hensikt er tjeneste

                                    If (H6_beleggsgrad) < 1 Then 'ingen ledige sitt
eplasser, men trengsel kan ha inntreidd

                                        If H6_beleggsgrad > MU_SIT_TJE Then

                                            H6_treng = H6_treng + ((H6_beleggsgrad
- MU_SIT_TJE) / (1 - MU_SIT_TJE)) * (F_SIT_INGENLEDIG_TJE - F_SIT_TJE) * H6_tid
sbruk * seg_tidsverdier(r, 1)(1)

                                        End If

                                        Else

                                            H6_ståperkvm = (Belegg(z, k, l)(i, j) - Set
ekapasitet_avgang(z, k, l)(j)) / Ståareal_avgang(z, k, l)(j)

                                            'disse sitter

                                            H6_treng = H6_treng + (F_SIT_INGENLEDIG_TJE
+ H6_ståperkvm * SIT_U_STÅ_TJE - F_SIT_TJE) * Setekapasitet_avgang(z, k, l)(j)
* H6_tidsbruk * seg_tidsverdier(r, 1)(1)

```

```

.....|                                     '|disse står...

                                     H6_treng = H6_treng + (F_STÅ_TJE + H6_ståpe
rkvm * STÅ_U_STÅ_TJE - F_SIT_TJE) * (Belegg(z, k, l)(i, j) - Setekapasitet_avga
ng(z, k, l)(j)) * H6_tidsbruk * seg_tidsverdier(r, l)(1)

                                     End If

                                     End If

                                     '****

                                     H6_treng = (H6_treng * Belegg_hensikt(r, z, k, l)(i
, j) / Belegg(z, k, l)(i, j)) '#3.2.1 (AUK)vekter med antall reisende pr hensik
t for å unngå dobbelttelling

                                     H6_trengsel_avg(k, l)(i, j) = H6_treng / 60 'deler
på seksti da beregningene skjer på minuttnivå

                                     Else

                                     H6_trengsel_avg(k, l)(i, j) = 0

                                     End If

                                     If z = 1 Then

                                     H6_trengsel_linje_ref(k, l) = H6_trengsel_linje_ref
(k, l) + H6_trengsel_avg(k, l)(i, j)

                                     Else

                                     H6_trengsel_linje_tiltak(k, l) = H6_trengsel_linje_
tiltak(k, l) + H6_trengsel_avg(k, l)(i, j)

                                     End If

                                     End If

                                     Next j

                                     Next i

                                     Next l

                                     Next k

Next r

```

Modul: V_FUNKSJONER

En rad kommentert ut:

```
' QuickSortArray = SortArray (#3.2.1 (AUK) koden funker kun når denne raden er kommentert)
```

Modul: I_____Resultatutskrift

```
'#3.2.1 (AUK) printer ut settfordeling  
  
'sletter resultater fra andre kjøring  
  
If sett_iter = 1 Then  
    Workbooks(wbName).Worksheets("Sett_iterasjoner").Cells.ClearContents  
    Workbooks(wbName).Worksheets("Sett_iterasjoner_t").Cells.ClearContents  
    Workbooks(wbName).Worksheets("Sett_iter_oppsummering").Cells.ClearContents  
  
End If  
  
'referanse  
  
ReDim Blokk(1 To antall_avganger_tot_ref + 1, 1 To 5) As Variant 'lager en utskiftsblokk som skrives ut som range til slutt  
  
    If SETTBRUKAV_REF = True Then  
  
        If sett_iter = 1 Then 'printer ut melding hvis settberegning ikke er valgt  
  
            Blokk(1, 1) = "Settberegning ikke valgt. Kan endres i inndata-ark Parametere (celle B58)"  
  
            ws = "Sett_iterasjoner"  
  
            Workbooks(wbName).Worksheets(ws).Range(Workbooks(wbName).Worksheets(ws).Cells(1, 1), Workbooks(wbName).Worksheets(ws).Cells(1, 1)) = Blokk  
  
        End If  
  
    Else
```

```

z = 1 'referanse
Blokk(1, 1) = "Linje it" & sett_iter
Blokk(1, 2) = "Retning it" & sett_iter
Blokk(1, 3) = "Avgang it" & sett_iter
Blokk(1, 4) = "Settantall it" & sett_iter
Blokk(1, 5) = "Belegg it" & sett_iter

i = 2

For k = 1 To Antall_LinjerZ(z)
    For m = 1 To 2
        For p = 1 To a(k, m, z)
            Blokk(i, 1) = linjer(z, k)
            Blokk(i, 2) = Retninger(m)
            Blokk(i, 3) = p
            Blokk(i, 4) = Settantall_avgang(z, k, m)(p)
            Blokk(i, 5) = Maxbeleggrad_avg(z, k, m)(p)

            i = i + 1

        Next p
    Next m
Next k

ws = "Sett_iterasjoner"

Workbooks(wbName).Worksheets(ws).Range(Workbooks(wbName).Worksheets(ws)
.Cells(1, 1 + (sett_iter - 1) * 5), Workbooks(wbName).Worksheets(ws).Cells(i, 5
+ (sett_iter - 1) * 5)) = Blokk

End If

'tiltak

ReDim Blokk(1 To antall_avganger_tot_tiltak + 1, 1 To 5) As Variant 'lager en u
tskiftsblokk som skrives ut som range til slutt

If SETTBROKAV_TILTAK = True Then

    If sett_iter = 1 Then 'printer ut melding hvis settberegning ikke er va
lgt

```



```

        Blokk(1, 1) = "Settberegning ikke valgt. Kan endres i inndata-
ark Parametere (celle B59)"

        ws = "Sett_iterasjoner_t"

        Workbooks(wbName).Worksheets(ws).Range(Workbooks(wbName).Worksheets
(ws).Cells(1, 1), Workbooks(wbName).Worksheets(ws).Cells(1, 1)) = Blokk

    End If

Else

    z = 2 'tiltak

    Blokk(1, 1) = "Linje it" & sett_iter

    Blokk(1, 2) = "Retning it" & sett_iter

    Blokk(1, 3) = "Avgang it" & sett_iter

    Blokk(1, 4) = "Settantall it" & sett_iter

    Blokk(1, 5) = "Belegg it" & sett_iter

    i = 2

    For k = 1 To Antall_LinjerZ(z)

        For m = 1 To 2

            For p = 1 To a(k, m, z)

                Blokk(i, 1) = linjer(z, k)

                Blokk(i, 2) = Retninger(m)

                Blokk(i, 3) = p

                Blokk(i, 4) = Settantall_avgang(z, k, m)(p)

                Blokk(i, 5) = Maxbeleggrad_avg(z, k, m)(p)

                i = i + 1

            Next p

        Next m

    Next k

    ws = "Sett_iterasjoner_t"

    Workbooks(wbName).Worksheets(ws).Range(Workbooks(wbName).Worksheets(ws)
.Cells(1, 1 + (sett_iter - 1) * 5), Workbooks(wbName).Worksheets(ws).Cells(i, 5
+ (sett_iter - 1) * 5)) = Blokk

End If

```

```

'printer ut oppsummering på settberegningene

If sett_iter = 1 Then

    ReDim Blokk(1 To 4, 1 To 4) As Variant 'lager en utskiftsblokk som skrives
    ut som range til slutt

    Blokk(1, 1) = "Referanse"

    Blokk(1, 2) = "Antall sett omfordelt"

    Blokk(1, 3) = "Tiltak"

    Blokk(1, 4) = "Antall sett omfordelt"

    ws = "Sett_iter_oppsummering"

    Workbooks(wbName).Worksheets(ws).Range(Workbooks(wbName).Worksheets(ws).Cells(1, 1), Workbooks(wbName).Worksheets(ws).Cells(sett_iter_maxgrense + 1, 4)) = Blokk

End If

ReDim Blokk(1 To 1, 1 To 4) As Variant 'lager en utskiftsblokk som skrives ut s
om range til slutt

Blokk(1, 1) = "Iterasjon " & sett_iter

Blokk(1, 2) = Sett_SumEndret_Iterasjon_ref(sett_iter)

Blokk(1, 3) = "Iterasjon " & sett_iter

Blokk(1, 4) = Sett_SumEndret_Iterasjon_tiltak(sett_iter)

ws = "Sett_iter_oppsummering"

Workbooks(wbName).Worksheets(ws).Range(Workbooks(wbName).Worksheets(ws).Cells(sett_iter + 1, 1), Workbooks(wbName).Worksheets(ws).Cells(sett_iter + 1, 4)) = Blokk

```

Modul: ZZZ_Settbruk

```
'Disse settingen er satt i alle moduler
```

```
Option Explicit 'Alle variabler/navn må erklæres med dim/Public el hvis ikke bl  
ir det feilmelding fra vba - dette gjør koden sikrere mot feil
```

```
Option Base 1 'Alle vektorer starter på plass én (ikke 0)
```

```
Sub settbruk()
```

```
' Ny modul i versjon #3.2.1. (AUK)
```

```
' Her blir alle avganger i både referanse og tiltak rangert etter maksimal bele  
gggrad på avgangen.
```

```
' Gitt info fra bruker så blir sett lagt til på avganger med høyeste belegg og  
fjernet fra avganger med laveste belegg.
```

```
' Settbruk-modulen beregner hvordan antall sett skal fordeles på avganger.
```

```
' Modulen er delt opp i følgende deler
```

```
' A. Identifiserer hvis settbruk skal kjøres for referanse og/eller tiltak
```

```
' B. Identifiserer maksimal beleggsgrad for hver avgang
```

```
' C. Rangerer avgangene i B
```

```
' D. Omfordeler eller legger til/fjerner sett
```

```
Dim Kapasitet_avg() As Variant 'kapasitet pr. avgang
```

```
Dim Maxbelegg_avg() As Variant 'maksimalt belegg pr. avgang
```

```
ReDim Kapasitet_avg(1 To 2, 1 To Antall_Linjer, 1 To 2) 'Antall_Linjer = maximu  
m_z(Antall_LinjerZ(1),Antall_LinjerZ(2))
```

```
ReDim Maxbelegg_avg(1 To 2, 1 To Antall_Linjer, 1 To 2)
```

```
ReDim Maxbeleggrad_avg(1 To 2, 1 To Antall_Linjer, 1 To 2) 'denne er public
```

```
ReDim Sett(1 To 2, 1 To Antall_Linjer, 1 To 2)
```

```
Dim belegg_ref() As Variant
```

```
Dim belegg_ref_sortert() As Variant
```

```
Dim belegg_tiltak() As Variant
```

```
Dim belegg_tiltak_sortert() As Variant
```

```
'-----  
' A. Styrer hvis settbruk skal beregnes for referanse og/eller tiltak
```

```
Dim A0 As Double 'hjelpevariabel
```

```
Dim B0 As Double 'hjelpevariabel
```

```
If SETTBROKAV_REF = False Then
```

```
    A0 = 1
```

```
    Debug.Print "- Kjører settbruk-modul for referanse -"
```

```
Else
```

```
    A0 = 2
```

```
End If
```

```
If SETTBROKAV_TILTAK = False Then
```

```
    B0 = 2
```

```
    Debug.Print "- Kjører settbruk-modul for tiltak -"
```

```
Else
```

```
    B0 = 1
```

```
End If
```

```
'-----
```

```

' B. og C. Identifiserer maks. beleggsgrad pr avgang og beregner belegg
'
'           for beregning med sett som legges til/fjernes

Dim maks_b As Double 'hjelpevariabel

antall_avganger_tot_ref = 0 'nullstiller
antall_avganger_tot_tiltak = 0 'nullstiller

' definerer midlertidige hjelpevektorer for løkken

Dim temp_Kapasitet_avg() As Double

Dim temp_Maxbelegg_avg() As Double

Dim temp_Maxbeleggrad_avg() As Double

Dim temp_Set() As Double

Dim temp_Setekapasitet_prsett() As Double

Dim temp_Ståareal_prsett() As Double

Dim temp_Set_Endret_Iterasjon() As Double

For z = A0 To B0 'tiltak og referanse

    For k = 1 To Antall_LinjerZ(z) 'linjer

        For m = 1 To 2 'retning

            ReDim temp_Kapasitet_avg(1 To a(k, m, z))

            ReDim temp_Maxbelegg_avg(1 To a(k, m, z))

            ReDim temp_Maxbeleggrad_avg(1 To a(k, m, z))

            'ReDim temp_Kapasitet_prsett_avgang(1 To a(k, m, z))

            ReDim temp_Set(1 To a(k, m, z))

            ReDim temp_Setekapasitet_prsett(1 To a(k, m, z))

            ReDim temp_Ståareal_prsett(1 To a(k, m, z))

            ReDim temp_Set_Endret_Iterasjon(1 To a(k, m, z))

```

```

    For p = 1 To a(k, m, z) 'avganger
        ' initierer/nullstiller hjelpevariabel som oppdateres ifølgende
        neste iterasjon
        If sett_iter = 1 Then temp_Set_Endret_Iterasjon(p) = 0

        ' beregner setekapasitet og ståkapasitet ved første iterasjon
        If sett_iter = 1 Then
            temp_Setekapasitet_prsett(p) = Setekapasitet_avgang(z, k, m
            )(p) / Settantall_avgang(z, k, m)(p)
            temp_Ståareal_prsett(p) = Ståareal_avgang(z, k, m)(p) / Set
            tantall_avgang(z, k, m)(p)
        End If

        'midlertidig settvektor
        temp_Set(p) = Settantall_avgang(z, k, m)(p)

        'beregner kapasiteter
        temp_Kapasitet_avg(p) = Setekapasitet_avgang(z, k, m)(p) + (Stå
        areal_avgang(z, k, m)(p) * STÅPLASSER_PERKVM)

        ' B. Identifiserer maksimal beleggsgard for hver avgang
        maks_b = 0 'hjelpevariabel
        For i = 1 To A_S(k, m, z) - 1 'stasjoner
            If Belegg(z, k, m)(i, p) > maks_b Then
                temp_Maxbelegg_avg(p) = Belegg(z, k, m)(i, p)
                temp_Maxbeleggrad_avg(p) = temp_Maxbelegg_avg(p) / temp
                _Kapasitet_avg(p)
                maks_b = Belegg(z, k, m)(i, p)
            End If
        Next i
    
```

```

        ' Beregner totalt antall avganger (alle linjer, alle retninger,
        alle avganger). Trengs i beregning etter loop.

        If z = 1 Then
            antall_avganger_tot_ref = antall_avganger_tot_ref + 1
        End If

        If z = 2 Then
            antall_avganger_tot_tiltak = antall_avganger_tot_tiltak + 1
        End If

    Next p

    ' Oppdaterer
    Kapasitet_avg(z, k, m) = temp_Kapasitet_avg
    Maxbelegg_avg(z, k, m) = temp_Maxbelegg_avg
    Maxbeleggrad_avg(z, k, m) = temp_Maxbeleggrad_avg
    'Kapasitet_prsett_avgang(z, k, m) = temp_Kapasitet_prsett_avgang
    Sett(z, k, m) = temp_Sett

    If sett_iter = 1 Then Sett_Endret_Iterasjon(z, k, m) = temp_Sett_Endret_Iterasjon 'forbereder

    'hvis det er første iterasjon så lagrer vi variabel for kapasitet p
    r. sett

    If sett_iter = 1 Then
        Setekapasitet_prsett(z, k, m) = temp_Setekapasitet_prsett
        Ståareal_prsett(z, k, m) = temp_Ståareal_prsett
    End If

Next m

Next k

Next z

```

```

'-----
' C. Rangerer avgangene i B fra minste til største belegg

' redimmet beleggsmatriser

If SETTBROKAV_REF = False Then
    ReDim belegg_ref(1 To antall_avganger_tot_ref, 1 To 4)
    ReDim belegg_ref_sortert(1 To antall_avganger_tot_ref, 1 To 4)
End If

If SETTBROKAV_TILTAK = False Then
    ReDim belegg_tiltak(1 To antall_avganger_tot_tiltak, 1 To 4)
    ReDim belegg_tiltak_sortert(1 To antall_avganger_tot_tiltak, 1 To 4)
End If

Dim ii As Double

' "resize" beleggsmatriser for å kunne sortere dem etterpå
For z = A0 To B0 'alt. For z = 1 To 2 referanse og tiltak
    ii = 1
    For k = 1 To Antall_LinjerZ(z)
        For m = 1 To 2
            For p = 1 To a(k, m, z)
                'referanse
                If z = 1 Then
                    belegg_ref(ii, 1) = k 'linje
                    belegg_ref(ii, 2) = m 'retning
                    belegg_ref(ii, 3) = p 'avgang
                End If
            Next p
        Next m
    Next k
Next z

```



```

        belegg_ref(ii, 4) = Maxbeleggrad_avg(z, k, m)(p)
        ii = ii + 1
    End If
    'tiltak
    If z = 2 Then
        belegg_tiltak(ii, 1) = k 'linje
        belegg_tiltak(ii, 2) = m 'retning
        belegg_tiltak(ii, 3) = p 'avgang
        belegg_tiltak(ii, 4) = Maxbeleggrad_avg(z, k, m)(p)
        ii = ii + 1
    End If
Next p
Next m
Next k
Next z

' Her sorteres listen med avganger fra minste til største belegg. Her brukes funksjonen QuickSortArray som er kodet i modulen V_Funksjoner.
For z = A0 To B0
    If z = 1 Then 'referanse
        belegg_ref_sortert = belegg_ref
        QuickSortArray belegg_ref_sortert, , , 4
    End If

    If z = 2 Then 'tiltak
        belegg_tiltak_sortert = belegg_tiltak
        QuickSortArray belegg_tiltak_sortert, , , 4
    End If

```

Next z

```
'-----  
' D. Omfordeler eller legger til/fjerner sett  
  
' Hvis tid: inndatasjekk, f.eks prosenttall etc., sjekk at b_min < b_maks  
  
' Identifiserer intervall [b_min, b_max]. De avganger som er innen intervallet  
blir ikke endret  
  
Dim b_min As Double  
  
Dim b_max As Double  
  
If MINBELEGG_IKKEFJERN = True Then  
    b_min = 0  
Else  
    b_min = MINBELEGG_FJERN / 100  
End If  
  
If MAXBELEGG_IKKETILFOR = True Then  
    b_max = 1  
Else  
    b_max = MAXBELEGG_TILFOR / 100  
End If  
  
Dim jj As Double  
  
'definerer nye sett fra hva brukeren har skrevet inn i modellen i første iteras  
jon  
  
If sett_iter = 1 Then  
    If SETTBROK_KONST = True Then
```

```

    sett_banken_ref = 0
    sett_banken_tiltak = 0
Else
    sett_banken_ref = SETTBROK_SETTMAX
    sett_banken_tiltak = SETTBROK_SETTMAX
End If
End If

' Her går vi over avganger der belegg er større enn maksbelegg eller mindre enn
minimumbelegg

' Sett_SumEndret_Iterasjon_ref (>0) lagrer hvor mange sett som blir tilført/fje
rnet totalt

'----- Referanse -----
If SETTBROKAV_REF = False Then 'kjører for referanse
    z = 1 'referanse
    ' Løkke over avganger med største belegg og legger til der det trengs
    For jj = antall_avganger_tot_ref To 1 Step -1
        If belegg_ref_sortert(jj, 4) > b_max Then
            k = belegg_ref_sortert(jj, 1) 'linje
            m = belegg_ref_sortert(jj, 2) 'retning
            p = belegg_ref_sortert(jj, 3) 'avgang
            Debug.Print jj, "Sett før omrokking:"; Sett(z, k, m)(p)
            ' ingen begrensning på sett pr. avgang:
            If SETTBROK_MAXPRAVGANG = True Then
                If Sett(z, k, m)(p) < 3 Then
                    Sett(z, k, m)(p) = Sett(z, k, m)(p) + 1
                    Sett_SumEndret_Iterasjon_ref(sett_iter) = Sett_SumEndret_I
erasjon_ref(sett_iter) + 1
                    Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z
, k, m)(p) + 1

```

```

        If SETTBROK_IKKEBEGRENSET = False Then
            sett_banken_ref = sett_banken_ref - 1 'oppdaterer banke
n
        End If
    End If

    Else 'begrensning på antall sett pr avgang:
        If Sett(z, k, m)(p) < 3 And Sett(z, k, m)(p) + 1 <= Maks_Togset
t_Linje(z, k) Then
            Sett(z, k, m)(p) = Sett(z, k, m)(p) + 1

            Sett_SumEndret_Iterasjon_ref(sett_iter) = Sett_SumEndret_It
erasjon_ref(sett_iter) + 1

            Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z
, k, m)(p) + 1

            If SETTBROK_IKKEBEGRENSET = False Then
                sett_banken_ref = sett_banken_ref - 1 'oppdaterer banke
n
            End If
        End If
    End If

    Else ' trenger ikke å legge til sett

        Sett_SumEndret_Iterasjon_ref(sett_iter) = Sett_SumEndret_Iterasjon_
ref(sett_iter) + 0

        Exit For
    End If

Next jj

jj = 1 'for sikkerhets skyld

' Løkke over avganger med minste belegg og fratrekker sett her hvis det går

For jj = 1 To antall_avganger_tot_ref

    ' FJERNE SETT

```

```

        k = belegg_ref_sortert(jj, 1)
        m = belegg_ref_sortert(jj, 2)
        p = belegg_ref_sortert(jj, 3)

        ' Ingen begrensning på antall nye sett, ta bort alle sett der belegg < b_min:

        If SETTBROK_IKKEBEGRENSET = True Then

            If belegg_ref_sortert(jj, 4) < b_min Then ' bruker b_min som øvre grense for fjerning av sett

                Debug.Print jj, "Sett før omrokking:"; Sett(z, k, m)(p)

                If Sett(z, k, m)(p) > 1 Then

                    Sett(z, k, m)(p) = Sett(z, k, m)(p) - 1

                    Sett_SumEndret_Iterasjon_ref(sett_iter) = Sett_SumEndret_Iterasjon_ref(sett_iter) + 1 'oppdaterer antall sett som blir endret

                    Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z, k, m)(p) - 1

                End If

                Debug.Print jj, "Sett etter omrokking:"; Sett(z, k, m)(p), "I banken:", sett_banken_ref

            End If

            Else ' Begrensning på antall nye sett; fjern sett til det ikke trenge lenger (til sett_banken_ref = 0)

                If sett_banken_ref < 0 And Sett(z, k, m)(p) > 1 Then

                    Debug.Print jj, "Sett før omrokking:"; Sett(z, k, m)(p)

                    Sett(z, k, m)(p) = Sett(z, k, m)(p) - 1

                    sett_banken_ref = sett_banken_ref + 1

                    Sett_SumEndret_Iterasjon_ref(sett_iter) = Sett_SumEndret_Iterasjon_ref(sett_iter) + 1 'oppdaterer antall sett som blir endret

                    Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z, k, m)(p) - 1

                    Debug.Print jj, "Sett etter omrokking:"; Sett(z, k, m)(p), "I banken:", sett_banken_ref

                ElseIf sett_banken_ref >= 0 Then Exit For

            End If

        End If

```

```

Next jj
End If 'avslutter for referanse

'----- tiltak -----

If SETTBROKAV_TILTAK = False Then 'beregne for tiltak

    z = 2 'tiltak

    ' Løkke over avganger med største belegg og legger til der det trengs

    For jj = antall_avganger_tot_tiltak To 1 Step -
1 'løkke over avganger, høyeste belegg til laveste belegg

        If belegg_tiltak_sortert(jj, 4) > b_max Then

            k = belegg_tiltak_sortert(jj, 1) 'linje

            m = belegg_tiltak_sortert(jj, 2) 'retning

            p = belegg_tiltak_sortert(jj, 3) 'avgang

            Debug.Print jj, "Sett før omrokking:"; Sett(z, k, m)(p)

            ' ingen begrensning på sett pr. avgang:

            If SETTBROK_MAXPRAVGANG = True Then

                If Sett(z, k, m)(p) < 3 Then

                    Sett(z, k, m)(p) = Sett(z, k, m)(p) + 1

                    Sett_SumEndret_Iterasjon_tiltak(sett_iter) = Sett_SumEndret
_Iterasjon_tiltak(sett_iter) + 1

                    Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z
, k, m)(p) + 1 'legger til sett på avgang

                    If SETTBROK_IKKEBEGRENSET = False Then 'hvis begrensning på
antall nye sett

                        sett_banken_tiltak = sett_banken_tiltak - 1 'oppdaterer
settbanken

                    End If

                End If

            Else 'begrensning på antall sett pr avgang:

                If Sett(z, k, m)(p) < 3 And Sett(z, k, m)(p) + 1 <= Maks_Togset
t_Linje(z, k) Then

                    Sett(z, k, m)(p) = Sett(z, k, m)(p) + 1

```

```

        Sett_SumEndret_Iterasjon_tiltak(sett_iter) = Sett_SumEndret
_Iterasjon_tiltak(sett_iter) + 1

        Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z
, k, m)(p) + 1 'legger til sett på avgang

        If SETTBRUK_IKKEBEGRENSET = False Then 'hvis begrensning på
antall nye sett

            sett_banken_tiltak = sett_banken_tiltak - 1 'oppdaterer
settbanken

        End If

    End If

    End If

    Debug.Print jj, "Sett etter omrokking:"; Sett(z, k, m)(p), "I ban
ken:", sett_banken_tiltak

    Else ' trenger ikke å legge til sett

        Sett_SumEndret_Iterasjon_tiltak(sett_iter) = Sett_SumEndret_Iterasj
on_tiltak(sett_iter) + 0

        Exit For

    End If

Next jj

jj = 1 'for sikkerhets skyld

' Løkke over avganger med minste belegg og fratrekker sett her hvis det går
/trengs

For jj = 1 To antall_avganger_tot_tiltak

    ' FJERNE SETT

    k = belegg_tiltak_sortert(jj, 1)

    m = belegg_tiltak_sortert(jj, 2)

    p = belegg_tiltak_sortert(jj, 3)

    ' Ingen begrensning på antall nye sett, ta bort alle sett der beleg
g < b_min:

    If SETTBRUK_IKKEBEGRENSET = True Then

        If belegg_tiltak_sortert(jj, 4) < b_min Then ' bruker b_min som
øvre grense for fjerning av sett

            Debug.Print jj, "Sett før omrokking:"; Sett(z, k, m)(p)

```

```

        If Sett(z, k, m)(p) > 1 Then
            Sett(z, k, m)(p) = Sett(z, k, m)(p) - 1

            Sett_SumEndret_Iterasjon_tiltak(sett_iter) = Sett_SumEndret_Iterasjon_tiltak(sett_iter) + 1 'oppdaterer antall sett som blir endret

            Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z, k, m)(p) - 1

            End If

            Debug.Print jj, "Sett etter omrokking:"; Sett(z, k, m)(p), "I banken:", sett_banken_tiltak

            End If

            Else ' Begrensning på antall nye sett; fjern sett til det ikke tren
            gs lenger (til sett_banken_tiltak = 0)

                If sett_banken_tiltak < 0 And Sett(z, k, m)(p) > 1 Then

                    Debug.Print jj, "Sett før omrokking:"; Sett(z, k, m)(p)

                    Sett(z, k, m)(p) = Sett(z, k, m)(p) - 1

                    sett_banken_tiltak = sett_banken_tiltak + 1

                    Sett_SumEndret_Iterasjon_tiltak(sett_iter) = Sett_SumEndret_Iterasjon_tiltak(sett_iter) + 1 'oppdaterer antall sett som blir endret

                    Sett_Endret_Iterasjon(z, k, m)(p) = Sett_Endret_Iterasjon(z, k, m)(p) - 1

                    Debug.Print jj, "Sett etter omrokking:"; Sett(z, k, m)(p), "I banken:", sett_banken_tiltak

                    ElseIf sett_banken_tiltak >= 0 Then Exit For

                End If

            End If

        Next jj

    End If 'avslutter for tiltak

'-----

' hvis ikke antall sett pr. avgang endres fra den forrige så hopper vi ut av loopen

If Sett_SumEndret_Iterasjon_ref(sett_iter) < 1 And Sett_SumEndret_Iterasjon_tiltak(sett_iter) < 1 Then 'denne kan muligens effektiviseres ved å koble Sett_SumEndret_Iterasjon_ref til Sett_Endret_Iterasjon

```



```
Debug.Print "Antall sett som blitt omfordelt er < 1 derfor avsluttes iteras  
jonene over Trenklin"
```

```
iterasjonslutt = True
```

```
End If
```

```
End Sub
```