

Change Log

1. Model 2016_08_01 and Model 2016_08_01_short

Modified by Francesco Gardumi

Main changes to previous version OSeMOSYS_2015_08_27 and OSeMOSYS_2015_08_27_short

- Fixed bugs in the **Storage Equations** and **Storage Constraints** of OSeMOSYS_2015_08_27_short.
- Parameter **REMinProductionTarget** is no more indicated by the user as a percentage of the demand for a given fuel f , but as a percentage of the production of a given fuel f . With the previous formulation the target could be put only on fuels for which **SpecifiedAnnualDemand** is indicated. The current formulation is more general and it allows the target to be put not only on fuels for final consumption, but also on intermediate fuels. This necessity arose in some cases.

Storage Equations

Previous equations

s.t. S5_and_S6_StorageLevelYearStart{ r in REGION, s in STORAGE, y in YEAR}: if $y = \min\{yy \text{ in YEAR} \mid \min(yy) \text{ then StorageLevelStart}[r,s]$ else $\text{StorageLevelYearStart}[r,s,y-1] + \sum\{ls \text{ in SEASON, } ld \text{ in DAYTYPE, } lh \text{ in DAILYTIMEBRACKET}\} \sum\{l \text{ in TIMESLICE:Conversions}[l,ls]>0 \& \& \text{ConversionId}[l,ld]>0 \& \& \text{Conversionlh}[l,lh]>0\} (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION:TechnologyToStorage}[r,t,s,m]>0\} (\text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversions}[l,ls] * \text{ConversionId}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION:TechnologyFromStorage}[r,t,s,m]>0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversions}[l,ls] * \text{ConversionId}[l,ld] * \text{Conversionlh}[l,lh])) * \text{YearSplit}[l,y] * \text{Conversions}[l,ls] * \text{ConversionId}[l,ld] * \text{Conversionlh}[l,lh] = \text{StorageLevelYearStart}[r,s,y];$

s.t. S9_and_S10_StorageLevelSeasonStart{ r in REGION, s in STORAGE, ls in SEASON, y in YEAR}: if $ls = \min\{lsls \text{ in SEASON}\} \min(lsls)$ then $\text{StorageLevelYearStart}[r,s,y]$ else $\text{StorageLevelSeasonStart}[r,s,ls-1,y] + \sum\{ld \text{ in DAYTYPE, } lh \text{ in DAILYTIMEBRACKET}\} \sum\{l \text{ in TIMESLICE:Conversions}[l,ls]>0 \& \& \text{ConversionId}[l,ld]>0 \& \& \text{Conversionlh}[l,lh]>0\} (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION:TechnologyToStorage}[r,t,s,m]>0\} (\text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversions}[l,ls] * \text{ConversionId}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION:TechnologyFromStorage}[r,t,s,m]>0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversions}[l,ls] * \text{ConversionId}[l,ld] * \text{Conversionlh}[l,lh])) * \text{YearSplit}[l,y] * \text{Conversions}[l,ls] * \text{ConversionId}[l,ld] * \text{Conversionlh}[l,lh] = \text{StorageLevelSeasonStart}[r,s,ls,y];$

s.t. S11_and_S12_StorageLevelDayTypeStart{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, y in YEAR}: if ld = min{ldld in DAYTYPE} min(ldld) then StorageLevelSeasonStart[r,s,ls,y] else StorageLevelDayTypeStart[r,s,ls,ld-1,y] + sum{lh in DAILYTIMEBRACKET} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionsl[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionsl[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh])) * DaySplit[lh,y]) * DaysInDayType[ls,ld-1,y] = StorageLevelDayTypeStart[r,s,ls,ld,y];

s.t. S13_and_S14_and_S15_StorageLevelDayTypeFinish{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, y in YEAR}: if ls = max{lsls in SEASON} max(lsls) && ld = max{ldld in DAYTYPE} max(ldld) then StorageLevelYearFinish[r,s,y] else if ld = max{ldld in DAYTYPE} max(ldld) then StorageLevelSeasonStart[r,s,ls+1,y] else StorageLevelDayTypeFinish[r,s,ls,ld+1,y] - sum{lh in DAILYTIMEBRACKET} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionsl[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionsl[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh])) * DaySplit[lh,y]) * DaysInDayType[ls,ld+1,y] = StorageLevelDayTypeFinish[r,s,ls,ld,y];

New equations

s.t. S5_and_S6_StorageLevelYearStart{r in REGION, s in STORAGE, y in YEAR}: if y = min{yy in YEAR} min(yy) then StorageLevelStart[r,s] else StorageLevelYearStart[r,s,y-1] + sum{ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET} sum{l in TIMESLICE:Conversionsl[l,ls]>0&&Conversionld[l,ld]>0&&Conversionlh[l,lh]>0} (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION:TechnologyToStorage[r,t,s,m]>0} (RateOfActivity[r,l,t,m,y-1] * TechnologyToStorage[r,t,s,m] * Conversionsl[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y-1] * TechnologyFromStorage[r,t,s,m] * Conversionsl[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh])) * YearSplit[l,y-1] * Conversionsl[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh] = StorageLevelYearStart[r,s,y];

s.t. S9_and_S10_StorageLevelSeasonStart{r in REGION, s in STORAGE, ls in SEASON, y in YEAR}: if ls = min{lsls in SEASON} min(lsls) then StorageLevelYearStart[r,s,y] else StorageLevelSeasonStart[r,s,ls-1,y] + sum{ld in DAYTYPE, lh in DAILYTIMEBRACKET} sum{l in TIMESLICE:Conversionsl[l,ls-1]>0&&Conversionld[l,ld]>0&&Conversionlh[l,lh]>0} (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION:TechnologyToStorage[r,t,s,m]>0} (RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionsl[l,ls-1] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionsl[l,ls-1] * Conversionld[l,ld] * Conversionlh[l,lh])) * YearSplit[l,y] * Conversionsl[l,ls-1] * Conversionld[l,ld] * Conversionlh[l,lh] = StorageLevelSeasonStart[r,s,ls,y];

s.t. S11_and_S12_StorageLevelDayTypeStart{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, y in YEAR}: if ld = min{ldld in DAYTYPE} min(ldld) then StorageLevelSeasonStart[r,s,ls,y] else StorageLevelDayTypeStart[r,s,ls,ld-1,y] + sum{lh in DAILYTIMEBRACKET} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld-1] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld-1] * Conversionlh[l,lh])) * DaySplit[lh,y]) * DaysInDayType[ls,ld-1,y] = StorageLevelDayTypeStart[r,s,ls,ld,y];

s.t. S13_and_S14_and_S15_StorageLevelDayTypeFinish{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, y in YEAR}: if ls = max{lsls in SEASON} max(lsls) && ld = max{ldld in DAYTYPE} max(ldld) then StorageLevelYearFinish[r,s,y] else if ld = max{ldld in DAYTYPE} max(ldld) then StorageLevelSeasonStart[r,s,ls+1,y] else StorageLevelDayTypeFinish[r,s,ls,ld+1,y] - sum{lh in DAILYTIMEBRACKET} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld+1] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld+1] * Conversionlh[l,lh])) * DaySplit[lh,y]) * DaysInDayType[ls,ld+1,y] = StorageLevelDayTypeFinish[r,s,ls,ld,y];

Found by Taco Niet and modified by Francesco Gardumi

Storage Constraints

Previous equations

s.t. SC1_LowerLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $0 \leq (\text{StorageLevelDayTypeStart}[r,s,ls,ld,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyToStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyFromStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]))) * \text{DaySplit}[lh,y])) - \text{MinStorageCharge}[r,s,y] * (\sum\{yy \text{ in YEAR: } y-yy < \text{OperationalLifeStorage}[r,s] \ \&\& \ y-yy \geq 0\} \text{NewStorageCapacity}[r,s,yy] + \text{ResidualStorageCapacity}[r,s,y]);$

s.t. SC1_UpperLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $(\text{StorageLevelDayTypeStart}[r,s,ls,ld,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyToStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyFromStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]))) * \text{DaySplit}[lh,y])) - \text{MinStorageCharge}[r,s,y] * (\sum\{yy \text{ in YEAR: } y-yy < \text{OperationalLifeStorage}[r,s] \ \&\& \ y-yy \geq 0\} \text{NewStorageCapacity}[r,s,yy] + \text{ResidualStorageCapacity}[r,s,y]);$

DaySplit[lh,y]))-(sum{yy in YEAR: y-yy < OperationalLifeStorage[r,s] && y-yy>=0}
NewStorageCapacity[r,s,yy]+ResidualStorageCapacity[r,s,y]) <= 0;

s.t. SC2_LowerLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: 0 <= if ld > min{ldld in DAYTYPE} min(ldld) then (StorageLevelDayTypeStart[r,s,ls,ld,y]-sum{lh in DAILYTIMEBRACKET:lh-lh<0} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]))-MinStorageCharge[r,s,y]*(sum{yy in YEAR: y-yy < OperationalLifeStorage[r,s] && y-yy>=0} NewStorageCapacity[r,s,yy]+ResidualStorageCapacity[r,s,y]);

s.t. SC2_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: if ld > min{ldld in DAYTYPE} min(ldld) then (StorageLevelDayTypeStart[r,s,ls,ld,y]-sum{lh in DAILYTIMEBRACKET:lh-lh<0} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]))-(sum{yy in YEAR: y-yy < OperationalLifeStorage[r,s] && y-yy>=0} NewStorageCapacity[r,s,yy]+ResidualStorageCapacity[r,s,y]) <= 0;

s.t. SC3_LowerLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: 0 <= (StorageLevelDayTypeFinish[r,s,ls,ld,y] - sum{lh in DAILYTIMEBRACKET:lh-lh<0} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]))-MinStorageCharge[r,s,y]*(sum{yy in YEAR: y-yy < OperationalLifeStorage[r,s] && y-yy>=0} NewStorageCapacity[r,s,yy]+ResidualStorageCapacity[r,s,y]);

s.t. SC3_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: (StorageLevelDayTypeFinish[r,s,ls,ld,y] - sum{lh in DAILYTIMEBRACKET:lh-lh<0} (((sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyToStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (sum{t in TECHNOLOGY, m in MODE_OF_OPERATION, l in TIMESLICE:TechnologyFromStorage[r,t,s,m]>0} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]))-(sum{yy in YEAR: y-yy < OperationalLifeStorage[r,s] && y-yy>=0} NewStorageCapacity[r,s,yy]+ResidualStorageCapacity[r,s,y]) <= 0;

s.t. SC4_LowerLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $0 \leq$
 if $ld > \min\{ldld \text{ in DAYTYPE}\} \min\{ldld\}$ then $(\text{StorageLevelDayTypeFinish}[r,s,ls,ld-1,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyToStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyFromStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh])) * \text{DaySplit}[lh,y])) - \text{MinStorageCharge}[r,s,y] * (\sum\{yy \text{ in YEAR: } y-yy < \text{OperationalLifeStorage}[r,s] \ \&\& \ y-yy \geq 0\} \text{NewStorageCapacity}[r,s,yy] + \text{ResidualStorageCapacity}[r,s,y]))$;

s.t. SC4_UpperLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: if $ld > \min\{ldld \text{ in DAYTYPE}\} \min\{ldld\}$ then $(\text{StorageLevelDayTypeFinish}[r,s,ls,ld-1,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyToStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyFromStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh])) * \text{DaySplit}[lh,y])) - (\sum\{yy \text{ in YEAR: } y-yy < \text{OperationalLifeStorage}[r,s] \ \&\& \ y-yy \geq 0\} \text{NewStorageCapacity}[r,s,yy] + \text{ResidualStorageCapacity}[r,s,y])) \leq 0$;

New equations

s.t. SC1_LowerLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $0 \leq$
 $(\text{StorageLevelDayTypeStart}[r,s,ls,ld,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyToStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyFromStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh])) * \text{DaySplit}[lh,y])) - \text{MinStorageCharge}[r,s,y] * (\sum\{yy \text{ in YEAR: } y-yy < \text{OperationalLifeStorage}[r,s] \ \&\& \ y-yy \geq 0\} \text{NewStorageCapacity}[r,s,yy] + \text{ResidualStorageCapacity}[r,s,y]))$;

s.t. SC1_UpperLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}:
 $(\text{StorageLevelDayTypeStart}[r,s,ls,ld,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyToStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyToStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE:TechnologyFromStorage}[r,t,s,m] > 0\} \text{RateOfActivity}[r,l,t,m,y] * \text{TechnologyFromStorage}[r,t,s,m] * \text{Conversionls}[l,ls] * \text{Conversionld}[l,ld] * \text{Conversionlh}[l,lh])) * \text{DaySplit}[lh,y])) - (\sum\{yy \text{ in YEAR: } y-yy < \text{OperationalLifeStorage}[r,s] \ \&\& \ y-yy \geq 0\} \text{NewStorageCapacity}[r,s,yy] + \text{ResidualStorageCapacity}[r,s,y])) \leq 0$;

s.t. SC2_LowerLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $0 \leq$ if $ld > \min\{ldd \text{ in } DAYTYPE\} \min(ldd)$ then $(StorageLevelDayTypeStart[r,s,ls,ld,y] - \sum\{lh \text{ in } DAILYTIMEBRACKET:lh-lh < 0\} (((\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyToStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyFromStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]) - MinStorageCharge[r,s,y] * (\sum\{yy \text{ in } YEAR: y-yy < OperationalLifeStorage[r,s] \&\& y-yy \geq 0\} NewStorageCapacity[r,s,yy] + ResidualStorageCapacity[r,s,y]);$

s.t. SC2_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: if $ld > \min\{ldd \text{ in } DAYTYPE\} \min(ldd)$ then $(StorageLevelDayTypeStart[r,s,ls,ld,y] - \sum\{lh \text{ in } DAILYTIMEBRACKET:lh-lh < 0\} (((\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyToStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyFromStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]) - (\sum\{yy \text{ in } YEAR: y-yy < OperationalLifeStorage[r,s] \&\& y-yy \geq 0\} NewStorageCapacity[r,s,yy] + ResidualStorageCapacity[r,s,y]) \leq 0;$

s.t. SC3_LowerLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $0 \leq$ $(StorageLevelDayTypeFinish[r,s,ls,ld,y] - \sum\{lh \text{ in } DAILYTIMEBRACKET:lh-lh < 0\} (((\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyToStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyFromStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]) - MinStorageCharge[r,s,y] * (\sum\{yy \text{ in } YEAR: y-yy < OperationalLifeStorage[r,s] \&\& y-yy \geq 0\} NewStorageCapacity[r,s,yy] + ResidualStorageCapacity[r,s,y]);$

s.t. SC3_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $(StorageLevelDayTypeFinish[r,s,ls,ld,y] - \sum\{lh \text{ in } DAILYTIMEBRACKET:lh-lh < 0\} (((\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyToStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (\sum\{t \text{ in } TECHNOLOGY, m \text{ in } MODE_OF_OPERATION, l \text{ in } TIMESLICE:TechnologyFromStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]))) * DaySplit[lh,y]) - (\sum\{yy \text{ in } YEAR: y-yy < OperationalLifeStorage[r,s] \&\& y-yy \geq 0\} NewStorageCapacity[r,s,yy] + ResidualStorageCapacity[r,s,y]) \leq 0;$

s.t. SC4_LowerLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: $0 \leq$

if $ld > \min\{ldld \text{ in DAYTYPE}\} \min(ldld)$ then $(StorageLevelDayTypeFinish[r,s,ls,ld-1,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE: } TechnologyToStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE: } TechnologyFromStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh])) * DaySplit[lh,y]) - MinStorageCharge[r,s,y] * (\sum\{yy \text{ in YEAR: } y-yy < OperationalLifeStorage[r,s] \ \&\& \ y-yy \geq 0\} NewStorageCapacity[r,s,yy] + ResidualStorageCapacity[r,s,y]);$

s.t. SC4_UpperLimit_BeginningOfDailyTimeBracketOfFirstInstanceOfDayTypeInLastWeekConstraint{r in REGION, s in STORAGE, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, y in YEAR}: if $ld > \min\{ldld \text{ in DAYTYPE}\} \min(ldld)$ then $(StorageLevelDayTypeFinish[r,s,ls,ld-1,y] + \sum\{lh \text{ in DAILYTIMEBRACKET: } lh-lh > 0\} ((\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE: } TechnologyToStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyToStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh]) - (\sum\{t \text{ in TECHNOLOGY, } m \text{ in MODE_OF_OPERATION, } l \text{ in TIMESLICE: } TechnologyFromStorage[r,t,s,m] > 0\} RateOfActivity[r,l,t,m,y] * TechnologyFromStorage[r,t,s,m] * Conversionls[l,ls] * Conversionld[l,ld] * Conversionlh[l,lh])) * DaySplit[lh,y]) - (\sum\{yy \text{ in YEAR: } y-yy < OperationalLifeStorage[r,s] \ \&\& \ y-yy \geq 0\} NewStorageCapacity[r,s,yy] + ResidualStorageCapacity[r,s,y]) \leq 0;$

Found by Taco Niet and modified by Francesco Gardumi

RE Production Target

Previous equations

s.t. RE3_FuelIncluded{r in REGION, y in YEAR}: $\sum\{l \text{ in TIMESLICE, } f \text{ in FUEL}\}$

RateOfDemand[r,l,f,y] * YearSplit[l,y] * REtagFuel[r,f,y] = RETotalDemandOfTargetFuelAnnual[r,y];

s.t. RE4_EnergyConstraint{r in REGION, y in YEAR}:

$REMinProductionTarget[r,y] * RETotalDemandOfTargetFuelAnnual[r,y] \leq$

$TotalREProductionAnnual[r,y];$

New equations

s.t. RE3_FuelIncluded{r in REGION, y in YEAR}: $\sum\{l \text{ in TIMESLICE, } f \text{ in FUEL}\}$

RateOfProduction[r,l,f,y] * YearSplit[l,y] * REtagFuel[r,f,y] =

RETtotalProductionOfTargetFuelAnnual[r,y];

s.t. RE4_EnergyConstraint{r in REGION, y in YEAR}:

$REMinProductionTarget[r,y] * RETotalProductionOfTargetFuelAnnual[r,y] \leq$

$TotalREProductionAnnual[r,y];$

2. Model 2015_08_27 and Model 2015_08_27_short

Modified by Abhishek Shivakumar and Manuel Welsch

Main changes to previous version OSeMOSYS_2013_05_10 and OSeMOSYS_2013_05_10_short

- Removed the parameter **TechWithCapacityNeededToMeetPeakTS** from constraint **CAa4_Constraint_Capacity**. This parameter caused **CapacityFactor** values to be ignored when the user did not set **TechWithCapacityNeededToMeetPeakTS** to a non-zero value. The removal of this parameter helps users avoid committing this error.
TechWithCapacityNeededToMeetPeakTS was used only once in the entire model and its removal does not affect the results-writing section of the code
- Fixed a bug related to using **CapacityOfOneTechnologyUnit** in constraint **CAa5_TotalNewCapacity**. The new fix allows the model to correctly employ a constraint on the capacity of individual technology units (which the model optimises using mixed integer programming (MIP))
- Fixed a bug in the storage equations which caused an error if more than one day type was used
- **DiscountRate** is no longer technology-specific. Therefore, **DiscountRateStorage** is now replaced by **DiscountRate**. In most OSeMOSYS models a single DiscountRate value is applied for all technologies, since data on financing levels for different technologies in different countries/regions is often lacking, a common region-specific (without including technology-specificity) DiscountRate is deemed to be sufficient.

CAa5_TotalNewCapacity equations

Previous equation

s.t. CAa1_TotalNewCapacity{r in REGION, t in TECHNOLOGY, y in YEAR}: AccumulatedNewCapacity[r,t,y] = sum{yy in YEAR: y-yy < OperationalLife[r,t] && y-yy >= 0}
if CapacityOfOneTechnologyUnit[r,t,y]=0 then NewCapacity[r,t,yy]
else CapacityOfOneTechnologyUnit[r,t,yy]*NumberOfNewTechnologyUnits[r,t,yy];

New equations

s.t. CAa1_TotalNewCapacity{r in REGION, t in TECHNOLOGY, y in YEAR}: AccumulatedNewCapacity[r,t,y] = sum{yy in YEAR: y-yy < OperationalLife[r,t] && y-yy >= 0}
NewCapacity[r,t,yy];

s.t. CAa5_TotalNewCapacity{r in REGION, t in TECHNOLOGY, y in YEAR:
CapacityOfOneTechnologyUnit[r,t,y] <> 0}:
CapacityOfOneTechnologyUnit[r,t,y]*NumberOfNewTechnologyUnits[r,t,y] = NewCapacity[r,t,y];

Found by Nawfal Saadi and modified by Manuel Welsch

CAa4_Constraint_Capacity

Previous equation (OSeMOSYS 2013 05 10)

s.t. CAa4_Constraint_Capacity{r in REGION, l in TIMESLICE, t in TECHNOLOGY, y in YEAR:
TechWithCapacityNeededToMeetPeakTS[r,t] <> 0}: RateOfTotalActivity[r,t,l,y] <=
TotalCapacityAnnual[r,t,y] * CapacityFactor[r,t,l,y]*CapacityToActivityUnit[r,t];

New equation (OSeMOSYS 2015 08 24)

s.t. CAa4_Constraint_Capacity{r in REGION, l in TIMESLICE, t in TECHNOLOGY, y in YEAR}:
 RateOfTotalActivity[r,t,l,y] <= TotalCapacityAnnual[r,t,y] *
 CapacityFactor[r,t,l,y]*CapacityToActivityUnit[r,t];

Previous equation (OSeMOSYS 2013 05 10 short)

s.t. CAa4_Constraint_Capacity{r in REGION, l in TIMESLICE, t in TECHNOLOGY, y in YEAR}:
TechWithCapacityNeededToMeetPeakTS[r,t]<>0: sum{m in MODE_OF_OPERATION}
 RateOfActivity[r,l,t,m,y] <= ((sum{yy in YEAR: y-yy < OperationalLife[r,t] && y-yy>=0}
 NewCapacity[r,t,yy]) + ResidualCapacity[r,t,y])*CapacityFactor[r,t,l,y]*CapacityToActivityUnit[r,t];

New equation (OSeMOSYS 2013 08 24 short)

s.t. CAa4_Constraint_Capacity{r in REGION, l in TIMESLICE, t in TECHNOLOGY, y in YEAR}: sum{m in
 MODE_OF_OPERATION} RateOfActivity[r,l,t,m,y] <= ((sum{yy in YEAR: y-yy < OperationalLife[r,t] &&
 y-yy>=0} NewCapacity[r,t,yy]) +
 ResidualCapacity[r,t,y])*CapacityFactor[r,t,l,y]*CapacityToActivityUnit[r,t];

Found by Nawfal Saadi and modified by Abhishek Shivakumar

Storage equations

Previous equation

s.t. SC2_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{s in
 STORAGE, y in YEAR, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, r in REGION}: if ld >
 min{ldld in DAYTYPE} min(ldld) then (StorageLevelDayTypeStart[s,y,ls,ld+1,r]-sum{lhlh in
 DAILYTIMEBRACKET:lh-lhlh<0} NetChargeWithinDay[s,y,ls,ld-1,lhlh,r])-StorageUpperLimit[s,y,r] <= 0;

New equation

s.t. SC2_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{s in
 STORAGE, y in YEAR, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, r in REGION}: if ld >
 min{ldld in DAYTYPE} min(ldld) then (StorageLevelDayTypeStart[s,y,ls,ld,r]-sum{lhlh in
 DAILYTIMEBRACKET:lh-lhlh<0} NetChargeWithinDay[s,y,ls,ld-1,lhlh,r])-StorageUpperLimit[s,y,r] <= 0;

Found and modified by Manuel Welsch

DiscountRate

Previous parameter definition

DiscountRate[r in REGION, t in TECHNOLOGY]
 DiscountRateStorage[r in REGION, s in STORAGE]

New parameter definition

DiscountRate[r in REGION]

Found by Tom Alfstad and modified by Abhishek Shivakumar

3. Model 2013_05_10_short

Modified by Abhishek Shivakumar and Manuel Welsch

Main changes to previous version OSeMOSYS_2013_05_10

- Significantly reduced the total number of equations by integrating them into the existing inequalities. This eliminates the need to calculate and store intermediate values.

The reduction in the number of equations translates into the generation of a smaller matrix to be solved. This significantly reduces the memory usage (~10x) and the processing time (~5x).

This version of the OSeMOSYS code contains only the essential equations required for running the model. However, all the previous equations have been left as before, and "commented out" to better understand the methodology followed to shorten the code.

It is important to note that the shortening of the code does not change any aspect of the functionality of OSeMOSYS. Furthermore, there are no special formatting requirements of data file required to run this version instead of the regular version. The short version of OSeMOSYS only serves to reduce the memory usage as well as the processing time for finding the model solution. In the future, both the regular and the short versions of OSeMOSYS will be developed and released simultaneously.

4. Model 2013_05_10

Modified by Abhishek Shivakumar and Manuel Welsch

Main changes to previous version OSeMOSYS_2013_04_30

- Re-ordered the arguments of parameters, variables and constraints to be compatible for use with the ANSWER interface which is currently being developed by Noble-Soft Systems Pty Ltd.

Henceforth, this reordered indexing will be applied to all future versions of the OSeMOSYS model.

Example

{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, m in MODE_OF_OPERATION, f in FUEL, r in REGION} is

now

{r in REGION, l in TIMESLICE, t in TECHNOLOGY, m in MODE_OF_OPERATION, f in FUEL, y in YEAR}

Following the above index order,

RateOfProductionByTechnologyByMode[y,l,t,m,f,r] is

now

RateOfProductionByTechnologyByMode[r,l,t,m,f,y]

5. Model 2013_04_30

Modified by Manuel Welsch

A bug was fixed which occurred if concrete blocks of capacity were to be considered, i.e., if the parameter `CapacityOfOneTechnologyUnit` was chosen to be unequal to zero. In this case the `NewCapacity` variable was left undefined. The derived variable `CapitalInvestment` was therefore not calculated correctly.

Previous equation

s.t. `CAa1_TotalNewCapacity`{y in YEAR, t in TECHNOLOGY, r in REGION}:

`AccumulatedNewCapacity`[y,t,r] = `sum`{yy in YEAR: y-yy < `OperationalLife`[t,r] && y-yy>=0}

if `CapacityOfOneTechnologyUnit`[y,t,r]=0 then `NewCapacity`[yy,t,r] else

`CapacityOfOneTechnologyUnit`[yy,t,r]*`NumberOfNewTechnologyUnits`[yy,t,r];

New equations

s.t. `CAa1_TotalNewCapacity`{r in REGION, t in TECHNOLOGY, y in YEAR}:

`AccumulatedNewCapacity`[y,t,r] = `sum`{yy in YEAR: y-yy < `OperationalLife`[t,r] && y-yy>=0}

`NewCapacity`[yy,t,r];

s.t. `CAa5_TotalNewCapacity`{r in REGION, t in TECHNOLOGY, y in YEAR:

`CapacityOfOneTechnologyUnit`[y,t,r]<>0}:

`CapacityOfOneTechnologyUnit`[y,t,r]*`NumberOfNewTechnologyUnits`[y,t,r] = `NewCapacity`[y,t,r];

6. Model 2013_03_14

Modified by Manuel Welsch

Main changes to previous version `OSeMOSYS_2012_06_01_BETA`

- Introduced the option to choose between sinking fund and straight line depreciation
- Removed parameter `SalvageFactor`, which was not used by the model
- Fixed a bug in the storage equations which caused an error if more than one day type was used
- Included table statements in the model file, immediately before the objective function and after the solve statement. This was done to demonstrate how parameters can be imported and exported. The table statements are commented out, and just serve as examples.

Option to choose the depreciation method

The previously used sinking fund depreciation results in a high “residual value” if a technology is invested in closely before the end of the modelling period. This might make investments in more capital intensive technologies with lower running costs more attractive. If this effect should be avoided, the

analyst now has the option to choose between sinking fund and straight line depreciation. This is done by setting the new parameter “DepreciationMethod” equal to 1 in order to use the straight line depreciation, or equal to 2 for straight line depreciation.

Changes in the model file

The following equations were removed:

s.t. SI7_SalvageValueStorageAtEndOfPeriod2{s in STORAGE, y in YEAR, r in REGION:
 $(y + \text{OperationalLifeStorage}[s,r] - 1) > (\max\{yy \text{ in YEAR}\} \max\{yy\}) \ \&\& \ \text{DiscountRateStorage}[s,r] = 0$:
 $\text{CapitalInvestmentStorage}[s,y,r] * (1 - (\max\{yy \text{ in YEAR}\} \max\{yy\} - y + 1) / \text{OperationalLifeStorage}[s,r]) =$
 $\text{SalvageValueStorage}[s,y,r]$;

s.t. SI8_SalvageValueStorageAtEndOfPeriod3{s in STORAGE, y in YEAR, r in REGION:
 $(y + \text{OperationalLifeStorage}[s,r] - 1) > (\max\{yy \text{ in YEAR}\} \max\{yy\}) \ \&\& \ \text{DiscountRateStorage}[s,r] > 0$:
 $\text{CapitalInvestmentStorage}[s,y,r] * (1 - (((1 + \text{DiscountRateStorage}[s,r])^{(\max\{yy \text{ in YEAR}\} \max\{yy\} - y + 1) - 1) / ((1 + \text{DiscountRateStorage}[s,r])^{\text{OperationalLifeStorage}[s,r] - 1}))) =$
 $\text{SalvageValueStorage}[s,y,r]$;

s.t. SV1_SalvageValueAtEndOfPeriod1{y in YEAR, t in TECHNOLOGY, r in REGION: $(y +$
 $\text{OperationalLife}[t,r] - 1) > (\max\{yy \text{ in YEAR}\} \max\{yy\}) \ \&\& \ \text{DiscountRate}[t,r] > 0$: $\text{SalvageValue}[y,t,r] =$
 $\text{CapitalCost}[y,t,r] * \text{NewCapacity}[y,t,r] * (1 - (((1 + \text{DiscountRate}[t,r])^{(\max\{yy \text{ in YEAR}\} \max\{yy\} - y + 1) - 1) / ((1 + \text{DiscountRate}[t,r])^{\text{OperationalLife}[t,r] - 1})))$;

s.t. SV2_SalvageValueAtEndOfPeriod2{y in YEAR, t in TECHNOLOGY, r in REGION: $(y +$
 $\text{OperationalLife}[t,r] - 1) > (\max\{yy \text{ in YEAR}\} \max\{yy\}) \ \&\& \ \text{DiscountRate}[t,r] = 0$: $\text{SalvageValue}[y,t,r] =$
 $\text{CapitalCost}[y,t,r] * \text{NewCapacity}[y,t,r] * (1 - (\max\{yy \text{ in YEAR}\} \max\{yy\} - y + 1) / \text{OperationalLife}[t,r])$;

The following equations were added:

The following equation only applies if the sinking fund depreciation is chosen and if the discount rate is larger than zero:

s.t. SI7_SalvageValueStorageAtEndOfPeriod2{s in STORAGE, y in YEAR, r in REGION:
 $(\text{DepreciationMethod}[r] = 1 \ \&\& \ (y + \text{OperationalLifeStorage}[s,r] - 1) > (\max\{yy \text{ in YEAR}\} \max\{yy\}) \ \&\& \ \text{DiscountRateStorage}[s,r] = 0 \ || \ (\text{DepreciationMethod}[r] = 2 \ \&\& \ (y + \text{OperationalLifeStorage}[s,r] - 1) > (\max\{yy \text{ in YEAR}\} \max\{yy\}))$:
 $\text{CapitalInvestmentStorage}[s,y,r] * (1 - (\max\{yy \text{ in YEAR}\} \max\{yy\} - y + 1) / \text{OperationalLifeStorage}[s,r]) =$
 $\text{SalvageValueStorage}[s,y,r]$;

The following equation will be used if the straight line depreciation will be chosen, or if the sinking fund depreciation will be chosen and the discount rate equals zero:

s.t. SI8_SalvageValueStorageAtEndOfPeriod3{s in STORAGE, y in YEAR, r in REGION:
 $\text{DepreciationMethod}[r] = 1 \ \&\& \ (y + \text{OperationalLifeStorage}[s,r] - 1) > (\max\{yy \text{ in YEAR}\} \max\{yy\}) \ \&\& \ \text{DiscountRateStorage}[s,r] > 0$:
 $\text{CapitalInvestmentStorage}[s,y,r] * (1 - (((1 + \text{DiscountRateStorage}[s,r])^{(\max\{yy \text{ in YEAR}\} \max\{yy\} - y + 1) - 1) / ((1 + \text{DiscountRateStorage}[s,r])^{\text{OperationalLifeStorage}[s,r] - 1}))) =$
 $\text{SalvageValueStorage}[s,y,r]$;

The following equation only applies if the sinking fund depreciation is chosen and if the discount rate is larger than zero:

s.t. SV1_SalvageValueAtEndOfPeriod1{y in YEAR, t in TECHNOLOGY, r in REGION:
 DepreciationMethod[r]=1 && (y + OperationalLife[t,r]-1) > (max{yy in YEAR} max(yy)) &&
 DiscountRate[t,r]>0}: SalvageValue[y,t,r] = CapitalCost[y,t,r]*NewCapacity[y,t,r]*(1-
 (((1+DiscountRate[t,r])^(max{yy in YEAR} max(yy)) - y+1)-
 1)/((1+DiscountRate[t,r])^OperationalLife[t,r]-1));

The following equation will be used if the straight line depreciation will be chosen, or if the sinking fund depreciation will be chosen and the discount rate equals zero:

s.t. SV2_SalvageValueAtEndOfPeriod2{y in YEAR, t in TECHNOLOGY, r in REGION:
 (DepreciationMethod[r]=1 && (y + OperationalLife[t,r]-1) > (max{yy in YEAR} max(yy)) &&
 DiscountRate[t,r]=0) || (DepreciationMethod[r]=2 && (y + OperationalLife[t,r]-1) > (max{yy in YEAR} max(yy))): SalvageValue[y,t,r] = CapitalCost[y,t,r]*NewCapacity[y,t,r]*(1-(max{yy in YEAR} max(yy)) - y+1)/OperationalLife[t,r];

Changes in the data file

The parameter DepreciationMethod has to be defined in the data file. It equals to 1 for the sinking fund and 2 for the straight line depreciation, e.g.,

param DepreciationMethod default 1 :=;

Reported by Bryce McCall and Philip Goyns, modified by Manuel Welsch

Storage equations

Previous equation

s.t. SC2_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{s in STORAGE, y in YEAR, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, r in REGION}: if ld > min{ldld in DAYTYPE} min(ldld) then (StorageLevelDayTypeStart[s,y,ls,ld+1,r]-sum{lh in DAILYTIMEBRACKET:lh-lh<0} NetChargeWithinDay[s,y,ls,ld-1,lh,r])-StorageUpperLimit[s,y,r] <= 0;

New equation

s.t. SC2_UpperLimit_EndOfDailyTimeBracketOfLastInstanceOfDayTypeInFirstWeekConstraint{s in STORAGE, y in YEAR, ls in SEASON, ld in DAYTYPE, lh in DAILYTIMEBRACKET, r in REGION}: if ld > min{ldld in DAYTYPE} min(ldld) then (StorageLevelDayTypeStart[s,y,ls,ld,r]-sum{lh in DAILYTIMEBRACKET:lh-lh<0} NetChargeWithinDay[s,y,ls,ld-1,lh,r])-StorageUpperLimit[s,y,r] <= 0;

7. Model 2012_06_01_BETA

Modified by Manuel Welsch

Trade between regions

Users requested an addition of a trade functionality between regions. This is done defining trade links between regions and adding the trade to the energy balance. If investments in, e.g., transmission capacity and losses should be modeled, a technology could be defined accordingly in one region which produces a fuel which is specifically intended for trade to another region.

Changes in model file

- Addition of new parameter to define the trade links between different regions: param TradeRoute{y in YEAR, f in FUEL, r in REGION, rr in REGION};
- Addition of new variables to calculate the actual trade from a region r to a region rr:
var Trade{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION, rr in REGION}; var TradeAnnual{y in YEAR, f in FUEL, r in REGION, rr in REGION};
- Added a new constraint EBa10, which ensures that whatever is exported from a region r to a region rr is as well imported from the region rr to the region r:
s.t. EBa10_EnergyBalanceEachTS4{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION, rr in REGION}: Trade[y,l,f,r,rr] = -Trade[y,l,f,rr,r];
- Renamed the previous constraint EBa10 to EBa11 and added the sum of the exports from a region r to all other regions rr to which export links exist:
s.t. EBa11_EnergyBalanceEachTS5{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION}:
Production[y,l,f,r] >= Demand[y,l,f,r] + Use[y,l,f,r] + sum{rr in REGION}
Trade[y,l,f,r,rr]*TradeRoute[y,f,r,rr];
- Added a new constraint EBb3 to calculate the annual trade:
s.t. EBb3_EnergyBalanceEachYear3{y in YEAR, f in FUEL, r in REGION, rr in REGION}: sum{l in TIMESLICE} Trade[y,l,f,r,rr] = TradeAnnual[y,f,r,rr];
- Renamed the previous constraint EBb3 to EBb4 and added the sum of the annual exports from a region r to all other regions rr to which export links exist:
s.t. EBb4_EnergyBalanceEachYear4{y in YEAR, f in FUEL, r in REGION}:
ProductionAnnual[y,f,r] >= UseAnnual[y,f,r] + sum{rr in REGION} TradeAnnual[y,f,r,rr]
*TradeRoute[y,f,r,rr] + AccumulatedAnnualDemand[y,f,r];

Changes in data file:

- The parameter TradeRoute will need to be defined. If everything should remain as it was previously, i.e., if no trade should be considered, the parameter should simply be set to zero:
param TradeRoute default 0 := ;

Introduction of technology additions in predefined blocks of capacity

Until now, OSeMOSYS was calculating the addition of technologies without any possibility to predefine the size of the capacities which could be added. For example, OSeMOSYS may have calculated that it

would be optimal to add 0.7 MW of coal-fired power plants in 2020, despite the unrealistically small size of such a unit. While this has significant advantages with regard to the computational power required to solve the model, we now have introduced mixed integer linear programming (MILP) to enable more realistic assessments. The user may now decide whether a faster but less accurate model run without MILP is preferred, or if the possibility to capture realistic power plant or technology additions justifies a more intense effort.

Changes in model file:

- Addition of new parameter to define the minimum size of one capacity addition:
param CapacityOfOneTechnologyUnit{y in YEAR, t in TECHNOLOGY, r in REGION};
- Addition of a new variable to calculate the number of units to be added:
var NumberOfNewTechnologyUnits{y in YEAR, t in TECHNOLOGY, r in REGION} >= 0, integer;
- Changes in the constraint CAa1 to ensure the previous method without MILP is used if no minimum size of one capacity addition is defined (i.e., if CapacityOfOneTechnologyUnit has a value of zero), and to change to MILP otherwise:
s.t. CAa1_TotalNewCapacity{y in YEAR, t in TECHNOLOGY, r in REGION}:
AccumulatedNewCapacity[y,t,r] = sum{yy in YEAR: y-yy < OperationalLife[t,r] && y-yy>=0}
if CapacityOfOneTechnologyUnit[y,t,r]=0 then NewCapacity[yy,t,r]
else CapacityOfOneTechnologyUnit[yy,t,r]*NumberOfNewTechnologyUnits[yy,t,r];

Changes in data file:

- The parameter CapacityOfOneTechnologyUnit will need to be defined. If everything should remain as it was previously, i.e., if no MILP should be used, the parameter should simply be set to zero: param CapacityOfOneTechnologyUnit default 0 := ;

Variability in generation

In order to introduce the modelling of variable electricity generation, the dimensions of the capacity factor are extended to include time slices in addition to years. This requires changing the parameter definition and the equations CAa4 and CAb1.

For a more detailed explanation refer to the following working paper at desa.kth.se: *Manuel Welsch, Mark Howells, Morgan Bazilian, Joseph DeCarolis, Sebastian Hermann, Hans-Holger Rogner. 2012. Modelling Selected Attributes of Smart Grids - A Contribution to Open Source Energy Systems Analysis.*

Changes in model file:

- param CapacityFactor{y in YEAR, t in TECHNOLOGY, l in TIMESLICE, r in REGION};
- s.t. CAa4_Constraint_Capacity{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, r in REGION: TechWithCapacityNeededToMeetPeakTS[t,r]<>0}: RateOfTotalActivity[y,l,t,r] <= TotalCapacityAnnual[y,t,r] * CapacityFactor[y,t,l,r]*CapacityToActivityUnit[t,r];
- s.t. CAb1_PlannedMaintenance{y in YEAR, t in TECHNOLOGY, r in REGION}: sum{l in TIMESLICE} RateOfTotalActivity[y,l,t,r]*YearSplit[y,l] <= sum{l in TIMESLICE}

$$(\text{TotalCapacityAnnual}[y,t,r] * \text{CapacityFactor}[y,t,l,r] * \text{YearSplit}[y,l]) * \text{AvailabilityFactor}[y,t,r] * \text{CapacityToActivityUnit}[t,r];$$

Changes in data file:

- Due to the additional dimension of the CapacityFactor, the parameter will have to be redefined accordingly. Refer to the parameter definition within the UTOPIA application for guidance.

Storage

Until previously, it was only possible to model storage in a basic fashion: Times within a year where extreme storage level were expected to occur had to be entered manually by the analyst. Similarly, storage capacities were exogenously defined and not optimised. A more sophisticated approach is therefore introduced, based on a sequential analysis of extreme storage levels throughout the year. If existing storage boundaries don't suffice, the model will investigate if new storage capacities should be added at a given cost of investment per unit of energy stored.

For a more detailed explanation of the concept and the required modifications in the model refer to the following working paper at desa.kth.se: *Manuel Welsch, Mark Howells, Morgan Bazilian, Joseph DeCarolis, Sebastian Hermann, Hans-Holger Rogner. 2012. Modelling Selected Attributes of Smart Grids - A Contribution to Open Source Energy Systems Analysis.*

Changes in data file:

- The following set is not required any longer: BOUNDARY_INSTANCES;
- The following sets need to be added:
 - set SEASON;
 - set DAYTYPE;
 - set DAILYTIMEBRACKET;
- The following parameters are not required any longer:
 - param StorageInflectionTimes{y in YEAR, l in TIMESLICE, b in BOUNDARY_INSTANCES};
 - param StorageUpperLimit{s in STORAGE, r in REGION};
 - param StorageLowerLimit{s in STORAGE, r in REGION};
- The following parameters need to be defined:
 - param ConversionIs{l in SEASON, l in TIMESLICE};
 - param ConversionId{l in DAYTYPE, l in TIMESLICE};
 - param ConversionIh{l in DAILYTIMEBRACKET, l in TIMESLICE};
 - param DaySplit{y in YEAR, lh in DAILYTIMEBRACKET};
 - param DaysInDayType{y in YEAR, ls in SEASON, ld in DAYTYPE};
 - param StorageLevelStart{s in STORAGE, r in REGION};
 - param StorageMaxChargeRate{s in STORAGE, r in REGION};
 - param StorageMaxDischargeRate{s in STORAGE, r in REGION};
 - param MinStorageCharge{s in STORAGE, y in YEAR, r in REGION};
 - param OperationalLifeStorage{s in STORAGE, r in REGION};

- param CapitalCostStorage{s in STORAGE, y in YEAR, r in REGION};
- param DiscountRateStorage{s in STORAGE, r in REGION};
- param ResidualStorageCapacity{s in STORAGE, y in YEAR, r in REGION};

Other changes:

- The following constraints will now only be used if the OutputActivityRatio[y,t,f,m,r] or the InputActivityRatio[y,t,f,m,r]<>0 in order to reduce the matrix size when solving the optimisation problem:
 - EBa1_RateOfFuelProduction1
 - EBa2_RateOfFuelProduction2
 - EBa4_RateOfFuelUse1
 - EBa5_RateOfFuelUse2
- In the following constraints, the limitation of the validity of the equation for upper parameter values < 9999 was removed:
 - TCC1_TotalAnnualMaxCapacityConstraint
 - NCC1_TotalAnnualMaxNewCapacityConstraint
 - AAC2_TotalAnnualTechnologyActivityUpperLimit
 - TAC2_TotalModelHorizonTechnologyActivityUpperLimit
- Changed model file extension to .mod and data file extension of the UTOPIA application to .dat to ensure a better integration within GUSEK (<http://gusek.sourceforge.net/gusek.html>), an integrated development environment (IDE) for the glpk solver.
- Deleted the variable definition of “TotalGenerationByRETechnologies”, as it is not used in the model.

Found by Oliver Broad

- Deleted the variable definition of “EmissionsProduction”, as it is not used in the model.

8. Model 2011/11/08

Found and corrected by Manuel Welsch

Previous equations:

s.t. Acc3_ModelPeriodCostByRegion{r in REGION};sum{y in YEAR, t in TECHNOLOGY}TotalDiscountedCost[y,t,r]=ModelPeriodCostByRegion[r];

New equations:

s.t. Acc4_ModelPeriodCostByRegion{r in REGION};sum{y in YEAR, t in TECHNOLOGY}TotalDiscountedCost[y,t,r]=ModelPeriodCostByRegion[r];

Reasons for the change

Cosmetic reasons.

Previous equations: CBa1-4
& CBb1...

New equations:
...have been renamed to CAa1-4 & CAb1

Reasons for the change

For consistency with paper:

Howells, M., Rogner, H., Strachan, N., Heaps, C., Huntington, H., Kypreos, S., Hughes, A., Silveira, S., DeCarolis, J., Bazillian, M., Roehrl, A., 2011. OSeMOSYS: The Open Source Energy Modeling System: An introduction to its ethos, structure and development. Energy Policy 39, 5850-5870.

Previous equations:

s.t. CC2_DiscountingCapitalInvestmenta{y in YEAR, t in TECHNOLOGY, r in REGION}:
CapitalInvestment[y,t,r]/((1+DiscountRate[t,r])^(y-StartYear)) = DiscountedCapitalInvestment[y,t,r];

New equations:

s.t. CC2_DiscountingCapitalInvestmenta{y in YEAR, t in TECHNOLOGY, r in REGION}:
CapitalInvestment[y,t,r]/((1+DiscountRate[t,r])^(y-min{yy in YEAR} min(yy))) =
DiscountedCapitalInvestment[y,t,r];

Reasons for the change

Using “min{yy in YEAR} min(yy)” finally removes the need to include the parameter ‘StartYear’ in the overall model.

9. Model 2011/07/07

Found and corrected by Manuel Welsch

Previous equations:

s.t. SV1_SalvageValueAtEndOfPeriod1{y in YEAR, t in TECHNOLOGY, r in REGION: (y + OperationalLife[t,r]) >= (StartYear + card(YEAR))}: SalvageValue[y,t,r] =
NewCapacity[y,t,r]*1((((1+DiscountRate[t,r])^(StartYear+card(YEAR) - y))-
1)/((1+DiscountRate[t,r])^OperationalLife[t,r]-
1));

s.t. SV2_SalvageValueAtEndOfPeriod2{y in YEAR, t in TECHNOLOGY, r in REGION: (y +

OperationalLife[t,r] < (StartYear + card(YEAR)); SalvageValue[y,t,r] = 0;
 s.t. SV3_SalvageValueDiscountedToStartYear{y in YEAR, t in TECHNOLOGY, r in REGION}:
 DiscountedSalvageValue[y,t,r] = SalvageValue[y,t,r]/((1+DiscountRate[t,r])^(1+card(YEAR)));"

New equations:

s.t. SV1_SalvageValueAtEndOfPeriod1{y in YEAR, t in TECHNOLOGY, r in REGION: (y + OperationalLife[t,r]-1) > (max{yy in YEAR} max(yy)) && DiscountRate[t,r]>0}: SalvageValue[y,t,r] = CapitalCost[y,t,r]*NewCapacity[y,t,r]*(1-(((1+DiscountRate[t,r])^(max{yy in YEAR} max(yy) - y+1)-1)/((1+DiscountRate[t,r])^OperationalLife[t,r]-1)));

s.t. SV2_SalvageValueAtEndOfPeriod2{y in YEAR, t in TECHNOLOGY, r in REGION: (y + OperationalLife[t,r]-1) > (max{yy in YEAR} max(yy)) && DiscountRate[t,r]=0}: SalvageValue[y,t,r] = CapitalCost[y,t,r]*NewCapacity[y,t,r]*(1-(max{yy in YEAR} max(yy) - y+1)/OperationalLife[t,r]);

s.t. SV3_SalvageValueAtEndOfPeriod3{y in YEAR, t in TECHNOLOGY, r in REGION: (y + OperationalLife[t,r]-1) <= (max{yy in YEAR} max(yy))}: SalvageValue[y,t,r] = 0;

s.t. SV4_SalvageValueDiscountedToStartYear{y in YEAR, t in TECHNOLOGY, r in REGION}:
 DiscountedSalvageValue[y,t,r] = SalvageValue[y,t,r]/((1+DiscountRate[t,r])^(1+max{yy in YEAR} max(yy)-min{yy in YEAR} min(yy)));

Reasons for the change

SV1-4: The Card expression yielded a number +1 higher than was required. StartYear + card(YEAR), for example would give the 'end year + 1'; 1 is subtracted from OperationalLife, as new technology is already available at beginning of year;

SV1-4: Using "max{yy in YEAR} max(yy)" and "min{yy in YEAR} min(yy)" removes the need to include the parameter 'StartYear' in these equations.

SV2: allows a discount rate of 0

Previous equations:

s.t. OC4_DiscountedOperatingCostsTotalAnnual{y in YEAR, t in TECHNOLOGY, r in REGION}:
 OperatingCost[y,t,r]/((1+DiscountRate[t,r])^(y-StartYear+0.5)) = DiscountedOperatingCost[y,t,r];

New equations:

s.t. OC4_DiscountedOperatingCostsTotalAnnual{y in YEAR, t in TECHNOLOGY, r in REGION}:
 OperatingCost[y,t,r]/((1+DiscountRate[t,r])^(y-min{yy in YEAR} min(yy)+0.5)) =
 DiscountedOperatingCost[y,t,r];

Reasons for the change

OC4 min{yy in YEAR} min(yy) removes the need to include the parameter 'StartYear'

Notes

The parameter 'StartYear' can now be removed from the data file. But keeping it there does not affect the results. To do:

- Update UTOPIA example (take out StartYear)
- Update the parameter description
- Update the UTOPIA example and documentation

Previous first printf statement: printf

```
"\n" >> "SelectedResults.csv";
```

New first printf statement: printf

```
"\n" > "SelectedResults.csv";
```

Reasons for the change:

Overwrites new selected results file with each new model run instead of attaching the new results at the end of the file.

Previous equation:

s.t. $TDC1_TotalDiscountedCostByTechnology\{y \text{ in } YEAR, t \text{ in } TECHNOLOGY, r \text{ in } REGION\}$:
 $DiscountedOperatingCost[y,t,r]+DiscountedCapitalInvestment[y,t,r] +$
 $AnnualTechnologyEmissionsPenalty[y,t,r]- DiscountedSalvageValue[y,t,r] =$
 $TotalDiscountedCost[y,t,r];$

New equations and variables:

var $DiscountedTechnologyEmissionsPenalty\{y \text{ in } YEAR, t \text{ in } TECHNOLOGY, r \text{ in } REGION\} \geq 0;$

s.t. $TDC1_TotalDiscountedCostByTechnology\{y \text{ in } YEAR, t \text{ in } TECHNOLOGY, r \text{ in } REGION\}$:
 $DiscountedOperatingCost[y,t,r]+DiscountedCapitalInvestment[y,t,r]+DiscountedTechnologyEmissions$
 $Penalty[y,t,r]-DiscountedSalvageValue[y,t,r] = TotalDiscountedCost[y,t,r];$

s.t. $E5_DiscountedEmissionsPenaltyByTechnology\{y \text{ in } YEAR, t \text{ in } TECHNOLOGY, r \text{ in } REGION\}$:
 $AnnualTechnologyEmissionsPenalty[y,t,r]/((1+DiscountRate[t,r])^{(y-\min\{yy \text{ in } YEAR\} \min\{yy\}+0.5)}) =$
 $DiscountedTechnologyEmissionsPenalty[y,t,r];$

Reasons for the change:

Emissions discounting is introduced.

Found and corrected by Mark Howells:

Previous equation:

s.t. E1_AnnualEmissionProductionByMode{y in YEAR, t in TECHNOLOGY, e in EMISSION, m in MODE_OF_OPERATION, r in REGION:EmissionActivityRatio[y,t,e,m,r]<>0}: sum{l in TIMESLICE} EmissionActivityRatio[y,t,e,m,r]*TotalAnnualTechnologyActivityByMode[y,t,m,r]=AnnualTechnologyEmissionByMode[y,t,e,m,r];

New equation:

s.t. E1_AnnualEmissionProductionByMode{y in YEAR, t in TECHNOLOGY, e in EMISSION, m in MODE_OF_OPERATION, r in REGION:EmissionActivityRatio[y,t,e,m,r]<>0}: EmissionActivityRatio[y,t,e,m,r]*TotalAnnualTechnologyActivityByMode[y,t,m,r]=AnnualTechnologyEmissionByMode[y,t,e,m,r];

Reasons for the change:

E1 included a redundant sum over model year load regions. That was deleted.

10. Model 2011/01/01

- Changes and bug fixes were introduced to make the model more usable.
- Changes in nomenclature
- All sets CAPITALISED (2010 SteerCom recommendation)
- All plain English descriptions, formulations and code are coded and cross referenced
- Bug Fixes (Note that all equations are labelled (see the model code))

(1) Corrects a bug calculating the OC1 and OC2 operating cost:

s.t. OC1_OperatingCostsVariable{y in YEAR,l in TIMESLICE, t in TECHNOLOGY, r in REGION}: sum{m in MODE_OF_OPERATION} AverageAnnualTechnologyActivityByMode[y,t,m,r] *VariableCost[y,t,m,r] = VariableOperatingCost[y,l,t,r];

was changed to:

s.t. OC1_OperatingCostsVariable{y in YEAR, t in TECHNOLOGY, r in REGION}: sum{m in MODE_OF_OPERATION} TotalAnnualTechnologyActivityByMode[y,t,m,r]*VariableCost[y,t,m,r] = AnnualVariableOperatingCost[y,t,r];

s.t. OC2_OperatingCostsVariableAnnual{y in YEAR,t in TECHNOLOGY, r in REGION}: sum {l in TIMESLICE} VariableOperatingCost[y,l,t,r] = AnnualVariableOperatingCost[y,t,r]; is now deleted and the numbering is updated and changed.

New numbering and code is as follows:

s.t. OC1_OperatingCostsVariable{y in YEAR,l in TIMESLICE, t in TECHNOLOGY, r in

REGION}: $\sum\{m \text{ in } \text{MODE_OF_OPERATION}\}$
 $\text{TotalAnnualTechnologyActivityByMode}[y,t,m,r] * \text{VariableCost}[y,t,m,r] =$
 $\text{AnnualVariableOperatingCost}[y,t,r];$
s.t. OC2_OperatingCostsFixedAnnual{y in YEAR,t in TECHNOLOGY, r in REGION}:
 $\text{TotalCapacityAnnual}[y,t,r] * \text{FixedCost}[y,t,r] = \text{AnnualFixedOperatingCost}[y,t,r];$
s.t. OC3_OperatingCostsTotalAnnual{y in YEAR,t in TECHNOLOGY,r in REGION}:
 $\text{AnnualFixedOperatingCost}[y,t,r] + \text{AnnualVariableOperatingCost}[y,t,r] =$
 $\text{OperatingCost}[y,t,r];$
s.t. OC4_DiscountedOperatingCostsTotalAnnual{y in YEAR, t in TECHNOLOGY, r in
REGION}: $\text{OperatingCost}[y,t,r] / ((1 + \text{DiscountRate}[t,r])^{(y - \text{StartYear} + 0.5)}) =$
 $\text{DiscountedOperatingCost}[y,t,r];$

(2) Also,changed:

AverageAnnualTechnologyActivityByMode for
TotalAnnualTechnologyActivityByMode in all code.

(3) Corrected equation Acc3

from:

s.t. Acc3_AverageAnnualRateOfActivity{y in YEAR,l in TIMESLICE, t in TECHNOLOGY, m
in MODE_OF_OPERATION, r in REGION}: $\text{RateOfActivity}[y,l,t,m,r] * \text{YearSplit}[y,l] =$
 $\text{AverageAnnualTechnologyActivityByMode}[y,t,m,r];$

to:

s.t. Acc3_AverageAnnualRateOfActivity{y in YEAR,t in TECHNOLOGY, m in
MODE_OF_OPERATION, r in REGION}: $\sum\{l \text{ in } \text{TIMESLICE}\}$
 $\text{RateOfActivity}[y,l,t,m,r] * \text{YearSplit}[y,l] =$
 $\text{TotalAnnualTechnologyActivityByMode}[y,t,m,r];$

11. Model 2010/2/11

Added features

-- Can use different units for capacity and energy flow data

12. Model 2010/2/4

Added features:

-- Demands with no specific time-slice profile are allowed

-- Sets of capacities are now given the option of either to meeting average annual demand or peak demand in the year.

13. Model 2009

Added features:

- Any number of emissions can be accounted for
- A limitless number of activities per technology are allowed
- Different discount factors for the start of the year (investment) and middle of the year (penalties and operating costs) are added
- The activities of technologies can be limited on an annual or model period level

- A reserve margin constraint for one fuel is added
- A % production RE Target is allowed
- Total Capacity constraints are added by year, or model period
- Investment Constraints are added by year

14. Model 2008

Features:

- Ensures capacity adequacy, demand & balance of energy
 - multiple technologies allowed
 - multiple energy demands allowed (all with specified time-slice distribution)
 - Operating & capital costs of technologies included
 - Limitless time-slices allowed
 - Multiple fuels allowed
 - Availability and capacity factors can be assigned to technologies
 - Salvage Cost calculations included for technologies that outlive the model period --
- Each technology is allowed two modes of operation - or any combination thereof.