New Features in ETSAP TIMES Versions 4.0 – 4.7

Compiled by:
Antti Lehtilä (VTT)

ENERGY TECHNOLOGY SYSTEMS ANALYSIS PROGRAM

Energy Technology Network

Major New Features

New Feature	Version	Exten-sion
Dispatching and unit commitment features	4.0	– / DUC
TIMES-Micro – Elastic demand functions	4.1	MICRO
Retrofitting and life extensions of technologies	4.1	–
Support for grid computing and MCA with sensitivity analysis	4.2	SENSIS MCA
Support for demand load shifting processes	4.3	–
Ancillary Balancing Services	4.4	ABS
TIMES-Macro – Linearized Formulation	4.5	MLF
Logit market sharing mechanism	4.7	ECB

Minor new features – 1

New Feature	Version	Related attribute
Shaping of capacity transfer coefficients	4.0	NCAP_CPX
Functional depreciation rates for salvage value	4.0	NCAP_FDR
Defining fixed input flow profile for storage via COM_FR	4.1	COM_FR
Reporting of market share constraint marginals	4.2	PRC_MARK
New attribute for uncertain taxes/subsidies	4.2	S_FLO_TAX
Reporting of power levels of flows	4.3	RPT_OPT
Variable timeslice cycles and dynamic timeslice tree	4.3–4.4	TS_CYCLE
Automatic filtering off domain violations in input data	4.3	DATAGDX
Option for reporting selected flows on ANNUAL level	4.4	RPT_OPT
Defining maximum lifetime cycles for storage	4.4	STG_MAXCYC

Minor new features – 2

New Feature	Version	Related attribute
Filtering off small values in reporting parameters	4.4	RPT_OPT
Support for fixing capacities to previous solution	4.4	REG_BDNCAP
Enhanced formulation for general storage (STS)	4.5	PRC_MAP
Support for independent input/output AFs for storage	4.5	NCAP_AFC
Non-vintaged FLO_FUNC and ACT_FLO for vintaged prc	4.5–4.7	FLO_FUNC
Parameter for proportional investment subsidies	4.5	NCAP_ISPCT
Dynamic bound constraints for COMPRD / COMNET	4.5	UC_DYNBD
Shaping of capacity-related flows by age	4.6	FLO_FUNCX
Support for risk-free reference rate for hurdle rates	4.6	G_RFRIR
Shaping of seasonal availability factors	4.6	NCAP_AFSX

Major New Features

Dispatching and Unit Commitment Features

- Includes a basic set of important parameters commonly used for dispatching and unit commitment modeling, some of which available at two levels of modeling detail to choose:
  - Minimum stable load levels – ACT_MINLD(r, y, p)
  - Bounds on activity ramping rates – ACT_UPS(r, y, p, bd)
  - Ramping costs – ACT_CSTRMP(r,y,p,bd,cur)
  - Minimum up-time / down-time – ACT_TIME(r, y, p, bd)
  - Partial load efficiencies ACT_LOSPL(r, y, p, bd) / ACT_LOSSD(r, y, p, upt, bd)
  - Start-up / shut-down costs – ACT_CSTUP(r,y,p,tsl,cur) / ACT_CSTSD(r,y,p,upt,bd,cur)
  - Duration of start-up / shut-down phase – ACT_SDTIME(r, v, p, upt, bd)
  - Maximum non-operational time for start-up type – ACT_MAXNON(r, y, p, upt)
- Discrete unit commitment option available by setting $SET DUC YES
  - Either discrete (NCAP_DISC) or semi-continuous (NCAP_SEMI) on-line capacities
  - The latter option may be recommended for TIMES at this point, because only a single unit for each process vintage is currently supported when using fully discrete units
- For full documentation, see: https://www.iea-etsap.org/docs/TIMES_Dispatching_Documentation.pdf

TIMES-Micro – Elastic Demand Functions

- Originally, TIMES supported only linearized own-price elasticities for demands
- Since v.4.1.0, several generalizations have been implemented for demand functions:
  - Non-linear formulation of own-price elasticities (like in Markal)
  - Constant Elasticity of Substitution (CES) demand functions (non-linear)
  - Linearized CES demand functions approximating CES preferences
  - Nested linearized CES demand functions (excluding recursive aggregations)
- For the linearized CES demand functions, one can additionally use several variants:
  - Linearization approximating a proper non-linear CES function
  - Linearization calibrated on the basis of user-defined substitution rates
  - Volume-preserving variant of a CES function, such that substitution as such does not affect the level of the aggregate demand (unless made elastic to the aggregate price)
  - The aggregate demand may also be may be made elastic to the aggregate price
- One can even use component-specific elasticities in the linearized CES formulation
- CES demand functions may be useful e.g. for the modeling of modal shifts in travel demands
- For more details, see documentation, Part II, Appendix D: TIMES Demand Functions, and/or https://www.iea-etsap.org/docs/TIMES-Micro-Note.pdf

Retrofitting and Life Extensions

- Retrofitting and lifetime extension may be useful technical options for extending the operating lifetime or for improving the performance of existing or even new plants at a later time
- Starting from version 4.1.2, TIMES offers a simple facility for supporting
  - the modeling of retrofit (RF) options, and
  - lifetime extension (LE) options
- Some basic characteristics of the RF/LE options:
  - Each host process can be modeled to have any number of different RF/LE options
  - Each of the RF/LE options are modeled in the same way as new technologies
  - Investment costs for the RF/LE options represent the additional costs of the refurbishment
  - Fixed and variable O&M costs must cover the full costs during the RF/LE operation
  - When an RF/LE option is installed, the same capacity of the host process will be retired
  - One can also force that any capacity of host process being retired early must be retrofitted
  - Each RF/LE option can only be applied to a single host process
- The TIMES attribute introduced for the modeling of RF/LE options is **PRC_REFIT(reg,prc,p)**
- For more details, see documentation:
  https://github.com/etsap-TIMES/TIMES_Documentation/blob/master/User_Notes/TIMES-Refits-Note.pdf

Grid computing and MCA with sensitivity analysis

- TIMES now supports so-called **grid computing**, when running sensitivity analysis
  - Parallel solver processes can be asynchronously spawned without waiting for the completion of previous runs
  - This can significantly speed up the total time needed for solving a large number of cases
  - Requires using the TIMES SENSIS facility, with a single phase mode only
- A Monte Carlo Analysis (MCA) facility has also been implemented, based on the GAMS grid computing option:
  - Allows one to use large samples randomly drawn from probability distributions defined over a set of uncertain parameters, e.g. by using LHS sampling methods
  - Reasonably good selection of different uncertain parameters are supported for MCA
  - TIMES is then run over the whole sample space (or a user-defined subset)
  - The facility includes a report generator, to avoid handling vast amounts of results data
  - The standard detailed TIMES results are nonetheless also available when needed
- For full documentation for the MCA facility, see: https://www.iea-etsap.org/projects/TIMES-MCA%20Final%20Report.zip

Support for Load Shifting Processes

- A new type of process, Load Shifting Process, introduced in TIMES v4.3.0
  - The input / output flows represent demand load shifting upwards and downwards
  - The allowed shifting operation can be constrained in several ways:
    - maximum fraction of demand, by DAYNITE timeslice and/or seasonally
    - maximum demand shifting in hours of advance or delay
  - Investment costs, fixed O&M cost and variable costs can be defined for the load shifting operation, and the process activity and capacity can be referred to in user constraints
  - An additional cost can be defined per load shifting of one unit of demand load by one hour, forward (bd=UP) or backward (bd=LO)
- Load shifting processes must be defined as STG or NST, and with STG_SIFT(r,y,prc,com,ts)
- Remark: any defined maximum time of advance or delay cannot strictly prohibit cascaded load shifts, due to the linearized formulation
- For more details, see documentation:
  https://github.com/etsap-TIMES/TIMES_Documentation/blob/master/User_Notes/TIMES-Load-Shifting.pdf

Ancillary Balancing Services

- Two types of constraints in TIMES can be considered for the electricity system reserves:
  - Longer-term reserves requirements by peaking equations (EQ_PEAK)
  - Balance of supply and demand of short-term reserve capacity (new ABS extension)
  - The ABS extension requires activation by: $SET ABS YES
- With the **ABS formulation**, short-term reserve capacities become endogenous, and the traditional peak constraint can be used to implement strategic reserves
- Reserve types that can be modeled with the ABS extension (from fast to slow response):
  - Frequency Containment Reserve (FCR): activated upwards or downwards
  - Automatic Frequency Restoration Reserve (aFRR): upwards or downwards
  - Manual Frequency Restoration Reserve (mFRR) or tertiary: upwards or downwards
  - Replacement Reserve (RR): upwards or downwards
- The ABS extension allows for endogenous demand and trade for operating reserves
- Each reserve is a user-defined commodity with deterministic and/or probabilistic demand
- For process potentially providing reserves, the following parameters need to be defined:
  - Minimum stable operating level and ramping-up/down for supply processes
  - Fraction of capacity available for reserve provision, both for supply & demand processes
- For more details, see: https://www.iea-etsap.org/projects/TIMES-BS-Documentation.pdf

TIMES-Macro Linearized Formulation

- Original **TIMES-Macro** is a reduced form general equilibrium model with NLP formulation
  - Practical to use only for rather small single-region models due to NLP overhead
  - Does not work consistently for multi-region models due to its simplified methodology
  - No essential differentiation between taxes and costs
- **TIMES-Macro-MSA** provides decomposed formulation with a stand-alone Macro module
  - Improved performance but may also have notable convergence issues
  - Includes Negishi iterations for reaching a consistent multi-regional equilibrium
- **TIMES-Macro-MLF** has linearized formulation, aims at improved convergence performance
  - NLP only used for the Baseline calibration phase (very quick)
  - Policy runs are formulated as LP models solved within a Negishi iteration loop
  - Like with the MSA formulation, includes a lump-sum rebate transfer of taxes / subsidies
- Some experience from using the TIMES-Macro-MLF formulation:
  - According to tests, quality of results compare well with the MSA formulation
  - Overall performance better than under the MSA formulation (many fewer main iterations)
  - Granularity of the linearization may increase degeneracy in the energy system solution
  - However, experience only from a few test cases, so may be considered experimental
- For more details, see:
  https://github.com/etsap-TIMES/TIMES_Documentation/blob/master/User_Notes/TIMES-Macro-MLF-Note.pdf

Logit Market Sharing Formulation – experimental

- Simulation of imperfect markets can be modeled via a market sharing algorithm giving a non-zero market share to technologies that are not optimal but are to some extent judged competitive, or close enough to optimality, to deserve a market share
- The closeness to competitiveness may be determined using a levelized cost measure or the dual information for each technology, together with adjustments incorporating intangible costs and preferences
- The TIMES Logit Market Sharing formulation provides such a market sharing mechanism
  - Always based on the levelized cost (LEC) measure, calculated by TIMES
  - The heterogeneity parameter defines the sensitivity of the market share on the LEC
  - User may define optional preference weights and intangible costs as socioeconomic factors
  - User may also define a penalty cost for relaxing each market share constraint (such that there is an upper limit for the distance from optimality to be able to gain market)
- The algorithm requires re-solving the model after determining the adjusted market shares
- Implemented as a TIMES extension ECB (for Economic Choice Behavior)
  - Requires activation by: $SET ECB YES
  - Currently an experimental feature of TIMES
- For more details, see:
  https://github.com/etsap-TIMES/TIMES_Documentation/blob/master/User_Notes/TIMES-Logit-Market-Sharing.pdf

Minor New Features

Minor new storage modeling features

- Defining a fixed input flow profile for a storage process
  - May be useful e.g. for various types of hydro power plants
  - Requires that the input commodity is tracked on ANNUAL level
  - Done by defining the storage process as an NST with PRC_NSTTS(ANNUAL), and the defining the profile with COM_FR on the input commodity
- Defining maximum lifetime cycles for a storage process
  - Implemented as a side-product of the ABS extension
  - Defined by setting STG_MAXCYC(reg,year,prc) = N
- Support for independent input/output AFs for storage
  - Regular NCAP_AFC combines input+output availability (often useful as such)
  - If needed, independent input/output AFs can now be enabled, like for normal processes
- Enhanced formulation for general storage (STS)
  - Experimental, to mitigate some shortcomings of the original STS implementation
  - Enabled by using $SET STSFLX YES
  - For more info, see documentation: TIMES-Experimental-STS-Variant.pdf

Minor features for shaping of capacity availability

- Defining capacity survival profiles, by shaping capacity transfer by age
  - May be useful e.g. for defining the survival rates of vehicle stocks by technology and age
  - Can be used both for vintaged and non-vintaged processes
  - Requires defining the SHAPE index to be applied to the capacity transfer coefficients
  - Defined by setting NCAP_CPX(reg,year,prc) = index
- Defining seasonal availabilities shaped independently of NCAP_AF
  - Implemented to facilitate more flexible shaping of availability / utilization factors
  - Requires defining the SHAPE index to be applied to the seasonal availability factors
  - Defined by setting NCAP_AFSX(reg,year,prc,bd) = index
  - If NCAP_AFX has been used, by default that would be applied both to NCAP_AF and seasonal availabilities, and to all bound types used, which can now be avoided by defining NCAP_AFSX where necessary
  - A null NCAP_AFSX shape index will just prevent NCAP_AFX from being applied to the seasonal availability factors of the given bound type

Minor features related to shaping process flows

- Defining non-vintaged FLO_FUNC or ACT_FLO for a **vintaged** process
  - May be useful in some cases, to override the default vintage-specific relations
  - For FLO_FUNC(cg1,cg2), can be defined by FLO_FUNCX(reg,v,prc,cg1,cg2) = −1
  - For ACT_FLO(com), can be defined by FLO_FUNCX(reg,v,prc,'ACT',com) = −1
  - Requires that the FLO_FUNC or ACT_FLO has been defined on the ANNUAL level, which is used for defining the overriding shape, according to milestone year
  - Effectively results in the relation being defined like in the **non-vintaged** case
  - Recall that an extrapolation option may also be needed for FLO_FUNCX
- Shaping of capacity-related process flows by age
  - An age-based shape can be now specified for flows defined by NCAP_COM
  - Can be used both for vintaged and non-vintaged processes
  - SHAPE index defined by FLO_FUNCX(reg,v,prc,'CAPFLO',com) = index
  - As usual, the shape itself is defined with SHAPE(index,age)

Minor features related to the objective function

- Defining functional depreciation rates for determining salvage values
  - May be useful in some cases, to accelerate the default salvage value depreciation
  - For example, solar PV systems may be considered depreciating much faster in value than the remaining lifetime, due to the cells deteriorating in their performance
  - Can be defined by setting NCAP_FDR(reg,year,prc) = rate (e.g. 0.15 for 15% per annum)
- Defining the reference rate for risk-free real interest (G_RFRIR)
  - Sensitivity analyses with the general discount rates might become inconsistent, unless all the technology-specific **hurdle rates** are also reviewed / revised, because they are based on the relation: hurdle rate = risk free rate + risk premium
  - Usually the general discount rate should represent the risk free rate
  - The risk free reference rate for the hurdle rates can now be specified by the user, independently of the general discount rate (which is otherwise the default)
  - TIMES then calculates the risk premiums based on the reference rate, and derives the effective hurdle rates by adding the risk premiums to the current general discount rate
  - If G_RFRIR is left undefined, or it equals the general discount rate, no difference is caused compared to the original formulation

Minor new features related to taxes and subsidies

- Defining investment subsidies in proportion to investment cost
  - May be useful in some policy cases, to avoid the need to specify the absolute monetary values or knowing the currency unit used for the investment costs
  - Can be defined by setting NCAP_ISPCT(reg,year,prc) = pct (e.g. 0.3 for a 30% subsidy)
- Defining uncertain commodity taxes and/or subsidies
  - Uncertain taxes and/or subsidies can be defined in both stochastic runs and in sensitivity & tradeoff analyses
  - The tax is specified by S_COM_TAX(reg,year,com,ts,com_var,cur,stage,sow) = value, where **com_var** is the PRD/NET indicator for gross/net commodity production
  - Commodity subsidies can also be defined as a negative tax
  - Note that the S_COM_TAX values are <u>additive</u> to the corresponding deterministic parameter (COM_TAXNET / COM_TAXPRD), i.e. they do not replace the latter

Bounding new capacities to a previous solution

- Bounding new capacity additions to a previous solution may be useful for many purposes, for example:
  - A few first periods usually represent model calibration years, which can be fixed to a feasible baseline solution to reduce model size in subsequent runs
  - Capacities from a previous solution may be used as constraints in a higher resolution run with additional modeling features
- Bounding can be requested by using REG_BDNCAP(reg,bd) = year
  - Bound type **bd** tells whether the previous solution is used for fixed, upper or lower bounds
  - Year value specifies the model year up to which the bounding is applied
  - Using $SET LPOINT <gdx file> is also required (the GDX file of the previous solution)
- When capacities are by default fixed to previous solution, one can request individual processes to be bounded only with an upper or lower bound
  - These exceptions can be defined by using NCAP_BND(reg,'0',prc,'N') = ±1, where the value +1 requests an upper bound only and the value −1 requests a lower bound only

Minor new reporting features

- Reporting of market share constraint marginals
  - Marginals of market share constraints are not by default reported but can be requested
  - Request by setting RPT_OPT('COMPRD',4) = 1;
- Reporting of power levels of flows
  - Normal Var_FIn / Var_Fout reporting represents energy flows by timeslice
  - Power levels would often be also of interest, obtained by dividing flows by G_YRFR
  - Requested by RPT_OPT(nrg_type,1) = 1, where nrg_type ∈ {ELC, LTHEAT, HTHEAT}
- Option for reporting selected flows on ANNUAL level
  - Request by RPT_OPT(com_type,3) = −1, where com_type ∈ {NRG, MAT, DEM, ENV}
  - Override by RPT_OPT(nrg_type,3) = ±1, where nrg_type ∈ {ELC, LTHEAT, HTHEAT}
- Filtering off small values in reporting parameters
  - In some cases the model solution may include a large number of very small levels and/or marginals for variables (e.g. when disabling crossover after barrier)
  - To filter off small values, use RPT_OPT(grp,9) = tol, where grp ∈ {ACT, FLO, CAP, NCAP} and solution values smaller than tol will then be filtered out

Minor new dynamic constraint feature

- Dynamic bound constraints for COMPRD / COMNET
  - Simplified process-wise dynamic bound constraints, originally supported only for VAR_ACT, VAR_CAP and VAR_NCAP, now supports also VAR_COMPRD / VAR_COMNET
  - UC_DYNBND(uc_n,bd) holds the list of constraint names **uc_n** handled in this special way
  - Note: These dynamic constraints are defined on the ANNUAL level only
- Reminder for the rules of using dynamic bound constraints:
  - The input set **uc_dynbnd** must be used for flagging the pairs (uc_n,bd) to be reserved for dynamic bound constraints
  - The input parameters UC_CAP, UC_NCAP, UC_ACT, UC_COMPRD, UC_COMNET can be used for defining growth/decay coefficients (side='LHS') and RHS constants (side='RHS')
  - The growth/decay coefficients (side='LHS') are given as annual multipliers (e.g. 1.1 for a 10% annual growth)
  - The RHS constants represent annual absolute values of additional growth/decay
  - The LHS is by default interpolated using option 5. If no LHS is specified, the RHS is by default interpolated with the option 10, like other bounds. However, if the LHS is also specified, the RHS is by default interpolated by the same option as the LHS.

Minor features for flexible timeslice specification

- Custom and variable timeslice cycles
  - In TIMES, the total number of DAYNITE cycles in each year is by default assumed to be 365, such that each DAYNITE cycle has a length of one day (24 hours). Similarly, for WEEKLY level the assumed number of cycles is 52.25, each with the length of 7 days
  - While the defaults work well for most models, in some cases one might wish to define timeslices with custom cycle lengths, and even of different lengths, which is now possible
  - Defined by setting TS_CYCLE(reg,ts) = N, where N specifies the length of the cycles below timeslice ts, in the number of days
- Dynamic timeslice trees
  - Higher time-slice resolution in some periods of finer focus, while lower resolution in other
  - For example, hourly resolution in the short term, day/night/peak resolution in the longer term
  - Enabled by setting **$SET DYNTS YES**
  - TS_OFF(r,ts,y1,y2) – set defining year ranges, indicating that the timeslice ts and all finer timeslices below it are not to be used for those model periods (milestone years) falling within the year range; should be defined for ts below the ANNUAL level (typically SEASONs)
  - Requires that user defines all timeslice-specific data for each of the timeslice configurations
  - For more info, see documentation: User_Note-TIMES-Dynamic-timeslices.docx

Miscellaneous minor improvements

- The enhanced DATAGDX facility
  - In TIMES, diagnosing unexpected problems in new model versions may often require tracking down differences in the input data between recent model versions (DATAGDX)
  - In version 4.3 of TIMES, this functionality was further enhanced, and is now automatically used also for filtering out any domain violations in the input data, which eliminates some severe problems that such domain violations were earlier causing in GAMS
- Enhancements in the levelization of timeslice attributes
  - Originally in TIMES, timeslice attributes related to process transformation were all levelized onto the process timeslice level only. Because in many models, most of the demand side processes are modeled on the ANNUAL level, this has imposed a limitation for detailed process characterization, e.g. in terms of timeslice-specific efficiencies
  - Starting from TIMES v4.7.0, the levelization has been enhanced as follows:
    - ACT_FLO(r,y,p,com,s) (& equivalent VDA_FLOP) is now always levelized onto the timeslice level of the commodity **com**
    - ACT_EFF(r,y,p,cg,s), where **cg** is the efficiency group, is now always levelized onto the level of the shadow group flow variables (i.e. onto the finest level of them)
    - FLO_EFF(r,y,p,cg,com,s) and FLO_EMIS(r,y,p,cg,com,s) are now always levelized onto the level of the source flows (i.e. the process flows in the group **cg**)