IRENA
International Renewable Energy Agency

IRENA FlexTool

Methodology

IRENA 2020

Contents

Major Assumptions

Contents

Perfect vs. uncertain forecast

IRENA FlexTool

- FlexTool has **no uncertainty** – it knows what will happen and solves a perfect dispatch
- In reality:
  - uncertainty forces transmission system operators to **commit more resources than needed**
  - uncertainty means, *e.g.*, **charging and discharging of energy storages cannot be fully optimal**
- FlexTool commits
  - resources sufficient for perfect dispatch
  - single upward reserve that can cover variability within model time step as well as contingencies
- **Reserves that are used to mitigate forecast errors** (longer than model time step) can be used during dispatch – FlexTool should not keep those reserved, because then they could not be used
- **Consequences:**
  - Value of storages can be higher than can be really achieved
  - Slightly smaller costs, since the model can manage with less online units

Linear vs. mixed integer modelling

- IRENA FlexTool does **not use integer** decision variables (*e.g.,* on/off)
  - Instead, startups have been linearised (unit can start also partially)

- It matters when:
  - Running operational optimisation for **actual system operation**
  - Comparing technologies that have distinct **start-up** characteristics (*e.g.,* gas turbine vs. gas engine)
  - **Small systems** with only few units
  - **Large individual units** in comparison to system size
  - System **stability** requires some units to be online (can be endogenously forced in FlexTool)

- It matters a little in:
  - **High level** long-term planning
  - Systems with **flexible generation** portfolios

Solving the problem

- FlexTool uses **GNU MathProg** language to formulate the optimisation problem to a separate solver
- **Solver minimises** (or maximises) a system of linear equations
  - flexModel.mod is a MathProg file and contains the equations
- For linear problems, **open source solvers perform quite well**
  - Especially Clp (Coin-or linear programming) used by FlexTool
- In mixed-integer problems commercial solvers are orders of magnitude better than open source solvers
- Efficient solution algorithms are based on **primal simplex, dual simplex, and interior point methods**
- Genetic algorithms, AI, particle swarm, annealing, etc. methods also exist, but are much slower
- Linear problems are typically **solved to global optimum** (integer problems are typically not – defined by solution gap)

Objective function

Cost minimisation

Capacity
=
+ pre-existing capacity [units: capacity]
+ forced new capacity [units: invested_capacity]
+ invested new capacity [*v_invest* | *v_investTransfer*]

Operation	+ **fixed** operation and maintenance costs	[capacity × unittype: fixed_cost]
	+ **variable** operation and maintenance costs	[*v_gen* | *v_charge* | *v_convert* × unittype: O&M_cost]
	+ **fuel** costs of units	[*v_fuelUse* × fuel: fuel_price]
	+ **CO2** emission costs	[*v_fuelUse* × fuel: CO2_content × master: CO2_cost]
	+ **start-up** costs	[*v_startup* × unittype: startup_cost]
Penalties	+ penalty cost for **loss of load**	[*v_slack* × master: loss_of_load_penalty]
	+ penalty cost for **insufficient upward reserves**	[*v_reserveSlack* × master: loss_of_reserves_penalty]
	+ penalty cost for **insufficient capacity margin**	[*v_capacitySlack* × master: lack_of_capacity_penalty]
	+ penalty cost for **curtailment of VRE**	[*v_curtail* × master: curtailment_penalty]
	+ penalty cost for **insufficient inertia**	[*v_inertiaSlack* × master: lack_of_inertia_penalty]
Investment	+ **unit investment** costs	[*v_invest* × unit_type: inv.cost_kW × annuity]
	+ **storage investment** costs	[*v_investStorage* × unit_type: inv.cost_kWh × annuity
	+ **transmission line investment** costs	[*v_investTransfer* × nodeNode: inv.cost_kW × annuity]

IRENA
International Renewable Energy Agency

Building Blocks

Contents

gridNodes

- Grids, nodes, and nodeGroups are defined in
  - **nodeGroup** and
  - **gridNode** sheets

A
nodeGroup
mainLand

nodeGroup sheet

A	B	C	D	E
grid	**node**	**nodeGroup**	**nodeGroup2**	**nodeGroup3**
elec	nodeA	mainLand		
elec	nodeB	mainLand		
elec	nodeC	mainLand		
elec	nodeD			

gridNode sheet

One of the basic building blocks, 1/2

- Grids are used to label different grids (*e.g.,* electricity and natural gas)
  - No equations or constraints related only to grid
  - Used when presenting results

- Combination of **gridNodes** used in defining the model

One of the basic building blocks, 2/2

- Grids and nodes are used to model **characteristics of geographical areas**,
  - Demand,
  - Reserve requirements,
  - Non-synchronous limit
  - Inertia limit
- One node can be part of only one grid
  - They can cover the same geographical area, but need different names
- Nodes can be modelled individually or as a group of nodes
- Transfers between nodes allows sharing generation and reserves

Demand of each node

IRENA FlexTool

- Net demand in each node is a sum of demand and import
  - Annual sums are defined in **gridNode** sheet
  - Hourly values are calculated based on normalised time series

Normalised demand of one hour:

Demand (t0001) =

ts_energy(t0001) / sum_t(ts_energy)

* annual demand

	A	B	C	D	E	F	G
	grid	node	nodeGroup	nodeGroup2	nodeGroup3	demand (MWh)	import (MWh)
	elec	nodeA	mainLand			7008000	350400
	elec	nodeB	mainLand			2190000	
	elec	nodeC	mainLand			3504000	
	elec	nodeD				438000	

gridNode sheet

	A	B	C	D
1		grid	elec	elec
2		node	nodeA	node
3	time			
4	t0000		589	208
5	t0001		537	203
6	t0002		506	197
7	t0003		482	190
8	t0004		472	188
9	t0005		454	184
10	t0006		423	175

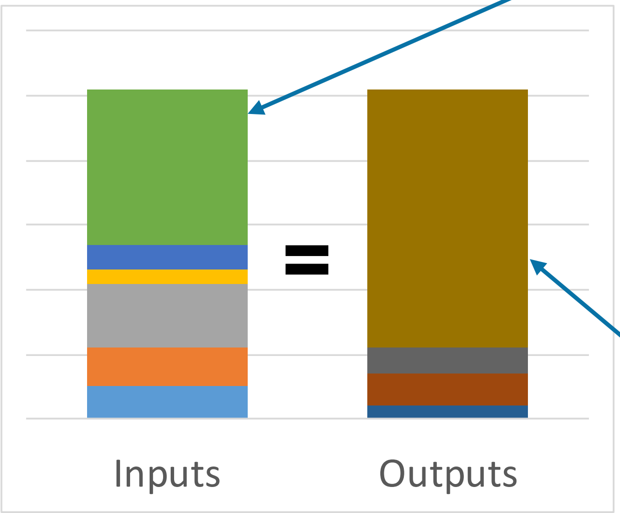
ts_energy sheet

	A	B	C
1		grid	elec
2		node	nodeA
3	time		
4	t0000		40
5	t0001		40
6	t0002		40
7	t0003		40
8	t0004		40
9	t0005		40

ts_import sheet

Energy balance equation

IRENA FlexTool



+ generation from non-VRE units	[*v_gen*]
+ generation from VRE units	[ts_cf: time series] × capacity
- curtailment of VRE units	[*v_curtail*]
+ imports*	[*v_transfer* and/or ts_import: time series]
+ energy conversions to the node*	[*v_convert*]
+ discharging of storages*	[*v_gen*]
+ loss of load	[*v_slack*]
=	
+ energy demand	[*v_charge* and/or ts_demand: time series]
+ exports*	[*v_transfer* and/or ts_import: time series]
+ energy conversions from the node	[*v_convert*]
+ charging of storages	[*v_charge*]

* can contain losses

Reserve requirements: Static reserves

IRENA FlexTool

- **Static reserves** are predefined time series that need to be activated
  - For single node or node group
- If static reserves are activated, every node and node_group requires own matching time series
- **Dynamic reserves** are calculated based on generating units (see next slide), but these need also to be activated

grid	node	nodeGroup	use ts_reserve	use dynamic reserve
elec	nodeA	mainLand	1	0
elec	nodeB	mainLand	1	0
elec	nodeC	mainLand	1	0
elec	nodeD		1	1

	A	B	C	D
1		node	nodeA	nodeB
2	Time			
3	t0000		18	6
4	t0001		16	6
5	t0002		15	6
6	t0003		14	6
7	t0004		14	6

gridNode sheet + ts_reserve_node

nodeGroup	capacity margin (MW)	non synchronous share	inertia limit (MWs)	use ts_reserve	use dynamic reserve	color in results
mainLand				1	1	

	A	B	C
1		nodeGroup	mainLand
2	Time		
3	t0000		54
4	t0001		50
5	t0002		47
6	t0003		45
7	t0004		44
8	t0005		43
9	t0006		40
10	t0007		41

nodeGroup sheet + ts_reserve_nodeGroup

Reserve requirements: Dynamic reserves

IRENA FlexTool

- **Dynamic reserve** can be defined for units
  - Reserve increase ratio in unit sheet
  - By default used with VRE generation
  - When unit generates, it increases the reserve need
  - *E.g.,* 10 MW of wind power is defined to need 1 MW reserves

- **Dynamic reserve** is not additional to static, model checks every hour (stricter requirements)

Input O

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	**unitGroup**	**unit type**	fuel	cf profile	inflow	input grid	input node	output grid	output node	capacity (MW)	invested capacity (MW)	max invest (MW)	storage (MWh)	invested storage (MWh)	max invest (MWh)	storage start	storage finish	**reserve increase ratio**
2	Coal	ST_coal	:oal					eleıde		#		0						
3	Oil	CC_oil	oil					eleıde		#		0						
4	Bio	ST_bio	ımass					eleıde		#		0						
5	Wind	wind	wind_A					eleıde		0		0						0.10
6	PV	PV	PV					eleıde		#		0						0.10
7	Battery	battery						eleıde		0			0		0			

unit sheet

Upward reserve requirement: single nodes

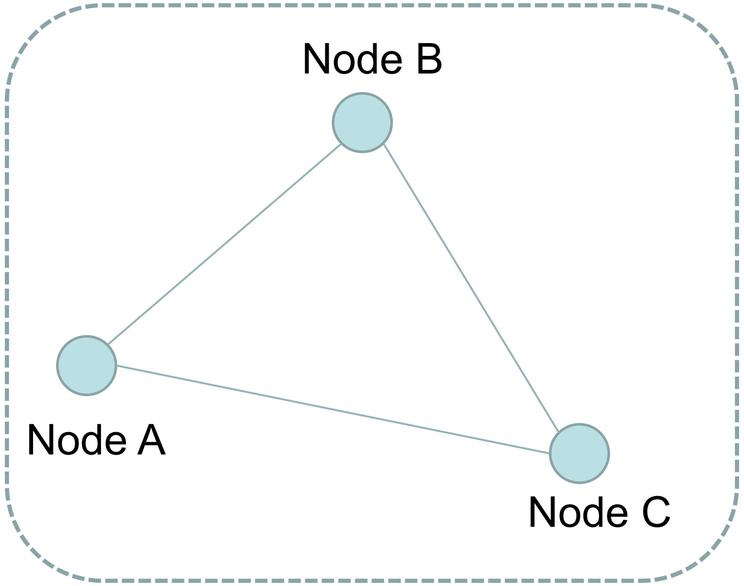
Single node reserve requirement:

+ sum of reserves from the units in the node	sum (units: [*v_reserve*(node, unit, t)]
+ reserve from VRE units	[*v_reserveVRE*(node, t)
+ lack of reserve penalty variable	[*v_reserveSlack*(node, t)
>=	
+ reserve requirement for the node	

Static
- [gridNode: use ts_reserve]
- [ts_reserve_node: time series]

Dynamic:
- [gridNode: use dynamic reserve]
- Sum (units: [*v_gen*(node, unit, t)] × [units: reserve_increase_ratio]

Upward reserve requirement: groups of nodes



NodeGroup ‘reserveNodes’
=
Union (Node A, Node B, Node C)

NodeGroup reserve requirement:

+ sum of reserves from the units in the nodes	sum (units: [*v_reserve*(node, unit, t)]
+ reserve from VRE units	[*v_reserveVRE*(node, t)
+ lack of reserve penalty variable	[*v_reserveSlack*(node, t)
>=	
+ reserve requirement for the node	

Static

- [nodeGroup: use ts_reserve]
- [ts_reserve_nodeGroup: time series]

Dynamic:

- [nodeGroup: use dynamic reserve]
- Sum (units: [*v_gen*(node, unit, t)] × [units: reserve_increase_ratio]

Further reserve limits

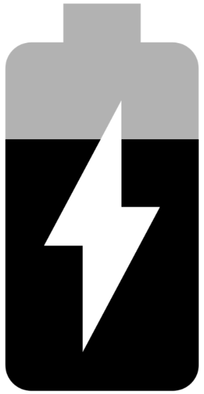
VRE upward reserve:

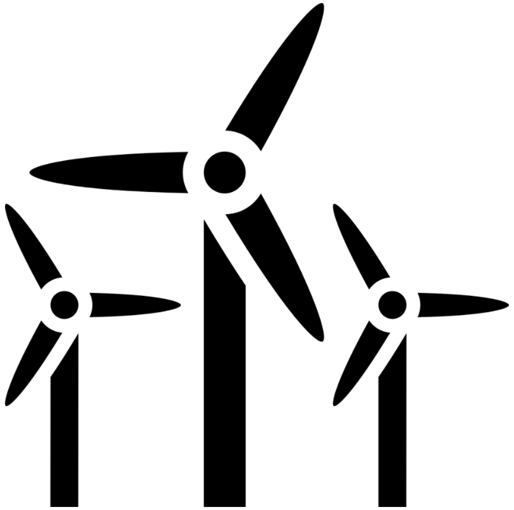
+ reserve from VRE [*v_reserveVRE*]

<=

+ VRE curtailment [*v_curtail*]

× reserve contribution [unit_type: max_reserve (0-1)]





Storage content reserve limit:

+ reserve from storage unit [*v_reserve*]

<=

+ charged energy [*v_state*]

/ duration of the reserve [master: reserve_duration]

Maximum non-synchronous share, 1/2

IRENA FlexTool

- **Maximum non-synchronous shares** activated in master sheet
- Defined for single node or node group
- Units are flagged synchronous (0) or non-synchronous (1) in unitType sheet

nodeGroup	capacity margin (MW)	non synchronous share	inertia limit (MWs)	use ts_reserve	use dynamic reserve	color in results
mainLand		0.80				

nodeGroup sheet

master sheet

parameter	value
co2_cost	10
loss_of_load_penalty	10000
loss_of_reserves_penalty	20000
lack_of_inertia_penalty	30000
curtailment_penalty	20
lack_of_capacity_penalty	5000
time_in_years	1.000
time_period_duration	60
reserve_duration	0.50
use_capacity_margin	1
use_online	1
use_ramps	0
use_non_synchronous	1
use_inertia_limit	0
mode_invest	0
mode_dispatch	1
print_duration	0
print_durationRamp	0
print_unit_results	0

grid	node	nodeGroup	G	G	demand (MWh)	import (MWh)	capacity margin (MW)	non synchronous share	use ts_reserve	use dynamic reserve	print results	color in results
elec	nodeA	mainLand					#	0.90				
elec	nodeB	mainLand					#	0.90				
elec	nodeC	mainLand					#	0.90				
elec	nodeD						5	0.80				

gridNode sheet

unit type	efficiency	min load	eff at min load	ramp up (p.u. per min)	ramp down (p.u. per min)	O&M cost /MWh	availability	max reserve	inertia constant (MWs/MW)	fixed cost /kW/year	inv.cost /kW	inv.cost /kWh	fixed kW/kWh ratio	conversion eff	startup cost	min uptime (h)	min downtime (h)	eff charge	self discharge loss	lifetime	interest	annuity	non synchronous
ST_coal																							0
Engine_gas																							0
CC_oil																							0
ST_bio																							0
Hydro_RES																							0
Hydro_ROR																							0
wind																							1
PV																							1
battery																							1

unitType sheet

Maximum non-synchronous share, 2/2

+ non-synchronous generation	[*v_gen*]
+ non-synchronous VRE generation	[ts_cf: time series] × capacity
- curtailment of VRE units	[*v_curtail*]
+ non-synchronous conversion	[*v_convert*]
+ HVDC transfer into the node	[*v_transfer*]
+ discharging of non-synch. storages	[*v_gen*]
<=	
maximum non synchronous share	[nodeGroup: non synchronous share]
× (	
+ energy demand	[*v_charge* and/or ts_demand: time series]
+ exports - imports	[*v_transfer* and/or ts_import: time series]
+ energy conversions from the node	[*v_convert*]
+ charging of storages	[*v_charge*]
- loss of load	[*v_slack*]
)	

Can be applied to
- group of nodes [nodeGroup]
- individual nodes [gridNode]

Minimum inertia limit, 1/2

IRENA FlexTool

- **Minimum inertia limit** needs to be activated from master sheet in input data (use_inertia_limit = 1)
- Defined only for node groups
- Inertia constant for each unit defined in unitType (MWs/MW)

parameter	value
co2_cost	10
loss_of_load_penalty	10000
loss_of_reserves_penalty	20000
lack_of_inertia_penalty	30000
curtailment_penalty	20
lack_of_capacity_penalty	5000
time_in_years	1.000
time_period_duration	60
reserve_duration	0.50
use_capacity_margin	1
use_online	1
use_ramps	0
use_non_synchronous	1
use_inertia_limit	0
mode_invest	0
mode_dispatch	1
print_duration	0
print_durationRamp	0
print_unit_results	0

master sheet

nodeGroup	capacity margin (MW)	non synchronous share	inertia limit (MWs)	use ts_reserve	use dynamic reserve	color in results
mainLand			100			

nodeGroup sheet

unit type	inertia constant (MWs/MW)
ST_coal	5.00
Engine_gas	2.00
CC_oil	3.00
ST_bio	5.00
Hydro_RES	3.00
Hydro_ROR	3.00
wind	
PV	
battery	

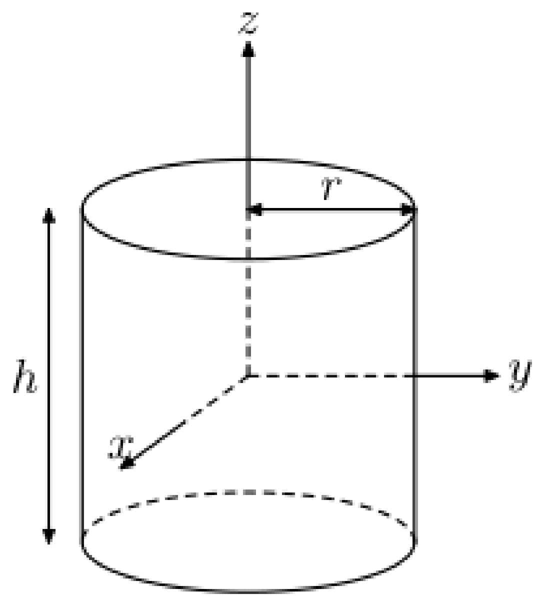
unitType sheet

Minimum inertia limit, 2/2

IRENA FlexTool

Can be applied only to group of nodes [nodeGroup]

+ online capacity of conventional online units	[*v_online*]
× inertia constant	[unit_type: inertia constant]
+ generation of conventional units without online	[*v_gen*]
× inertia constant	[unit_type: inertia constant]
+ generation from VRE units	[ts_cf: time series] × capacity
× inertia constant	[unit_type: inertia constant]
+ lack of inertia penalty variable	[*v_inertiaSlack*]
>=	
+ inertia limit in MWs	[nodeGroup: inertia limit]



Transfers, 1/2

IRENA FlexTool

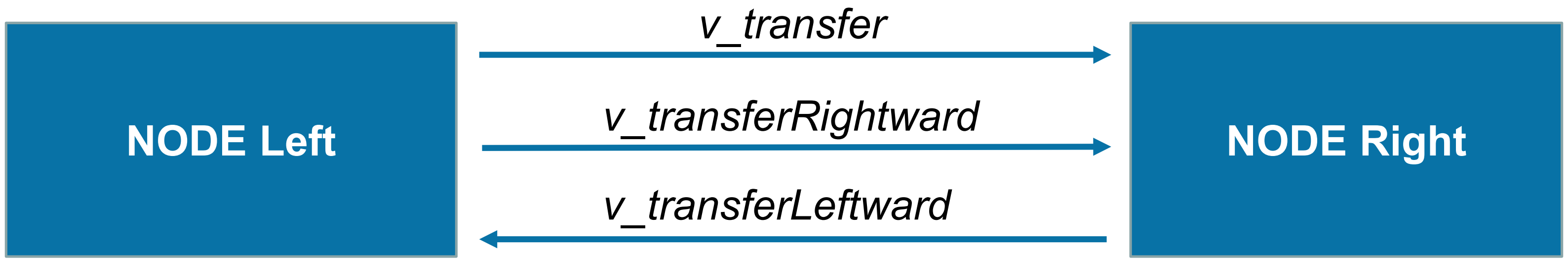
- **Transfers** between nodes are defined in nodeNode sheet
  - Both nodes have to be from the same grid
  - Existing transfer links can have different capacity to different direction
  - Future investments will always have equal capacity to both directions

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	grid	node1	node2	cap.rightward (MW)	cap.leftward (MW)	invested capacity (MW)	max invest (MW)	loss	inv.cost/kW	lifetime	interest	annuity	HVDC	color in results
2	elec	nodeA	nodeB	150	150		0	0.01	100	50	0.08	0.082	0	
3	elec	nodeB	nodeC	100	100		0	0.01	100	50	0.08	0.082	0	

nodeNode sheet

Transfers, 2/2

- **Transfer with losses** requires at least two variables
  - A linear equation with ‘loss x transfer’ would mean that in the other direction the loss is actually a gain
- The loss can be used to make the **model ‘leak’**
  - Instead of curtailing VRE, the model can dissipate energy by transferring in two directions at once
  - Can be controlled only with a binary variable (not allowed in FlexTool)
- Hence, **three variables**: transfer, transfer rightward and transfer leftward
  - Transfer does not contain loss
  - Transfer rightward allows losses and helps to limit the leakage
  - Transfer leftward helps to limit the leakage further



Transfer losses (balance equation)

- **Two nodes:** left and right (a node-node link is established with one direction only)
- When transferring from **left to right**:
  - Left node: transfer deducted from node balance
  - Right node: transfer minus loss added to the node balance
- When transferring from **right to left**:
  - Left node: transfer minus loss added to the node balance
  - Right node: transfer deducted from node balance

Balance leftward node

+ transfer	[*v_transfer*(g,n,n_right,t)]
minus loss	× (1 – [nodeNode: loss])
+ rightward transfer	[*v_transferRightward*(g,n,n_right,t)]
loss	× [nodeNode: loss]

When rightward: CANCELS EACH OTHER

Balance rightward node

+ transfer	[*v_transfer*(g,n_left,n,t)]
- rightward transfer	[*v_transferRightward*(g,n_left,n,t)]
× loss	× [nodeNode: loss]

When leftward: ZERO

Transfer constraints

IRENA FlexTool

Tie transfers together

+ transfer [*v_transfer*(g,n_left,n_right,t)]

=

+ rightward transfer [*v_transferRightward*(g,n_left,n_right,t)]

- leftward transfer [*v_transferLeftward*(g,n_left,n_right,t)]

Limit rightward transfer

+ transfer [*v_transfer*(g,n_left,n_right,t)]

<=

+ capacity [see orange box]

Limit rightward transfer again

+ transfer rightward [*v_transferRightward*(g,n_left,n_right,t)]

<=

+ capacity [see orange box]

And same for leftward transfers!!!

Rightward capacity

=

+ pre-existing leftward transfer capacity [nodeNode: cap_rightward]

+ forced new capacity [nodeNode: invested_capacity]

+ invested new capacity [*v_investTransfer*]

Concluding remarks on transfers

IRENA FlexTool

- Transfer with losses works, but the **model can leak**
- In normal circumstances, the model does not leak (why waste energy?), but the model can use leakage instead of VRE curtailment (if curtailment has a penalty cost)
  - *E.g.,*
    - Rightward transfer = 100 MW
    - Leftward transfer = -100 MW
    - → Transfer = 0
    - Loss = 5 %
    - → Leakage = 100 MW × 5% = 5 MW
- Leakage shown in Summary sheet of Results (‘Model leakage TWh/a’)

One of the basic building blocks

IRENA FlexTool

- **Units** are used to model
  - Power plants,
  - Storages,
  - Inflow units, *e.g.,* hydro power,
  - VRE units, *e.g.,* wind and solar,
  - Scheduled run units,
  - Conversion units (*e.g.,* power to heat), etc.

- Units are modelled slightly differently in dispatch and invest modes. Invest mode simplifies equations, because it is much slower to solve.

Units, dispatch, main constraints and equations

IRENA FlexTool

Main constraints and equations for units in dispatch mode are

- Upward limit of generation, reserve provision, and storage charge/discharge **(always on)**
  - Four unit categories: generating unit, inflow unit without storage, VRE unit, conversion unit
- Online variable **(activated by user)**. If activated, units can have
  - Start-up costs,
  - Minimum uptime and downtime, and
  - Efficiency loss with partial load
- Ramp constraint **(activated by user)**
- Costs related to unit operations **(always on, but parameters can have 0 value)**
  - Variable costs: fuel, variable O&M, CO_2 cost, startup costs
  - Fixed costs: fixed O&M

Unit category

- **Four unit categories:** generating unit, inflow unit without storage, VRE unit, conversion unit
  - FlexTool decides unit category based on the unit input defined in “units” sheet
  - Input options: fuel, cf profile, inflow, input grid/node, none

Input source (fuel, cf, inflow **or** model node)

Output #1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	unitGroup	unit type	fuel	cf profile	inflow	input grid	input node	output grid	output node	capacity (MW)	invested capacity (MW)	max invest (MW)	storage (MWh)
2	Coal	ST_coal	coal					elec	nodeA	##		#	
3	Oil	CC_oil	oil					elec	nodeA	##		#	
4	Bio	ST_bio	biomass					elec	nodeA	##		#	
5	Wind	wind		wind_A				elec	nodeA	##		#	
6	PV	PV		PV				elec	nodeA	##		#	
7	Battery	battery						elec	nodeA	##			200
8	Hydro	Hydro_RES			nodeB_RES			elec	nodeB	##		0	150000
9	Hydro	Hydro_ROR			nodeB_ROR			elec	nodeB	##		0	0
10	charger	EVcharger				elec	nodeB	elec	EV	##			

units sheet

- **ST_Coal:** Input from fuel -> generating unit
- **Wind**: input from cf_profile -> VRE unit
- **Battery**: no input + storage -> generating unit
- **Hydro_RES**: inflow + storage -> generating unit
- **Hydro_ROR**: inflow, but no storage -> inflow unit without storage
- **EVcharger**: input from node -> conversion unit

Upward limit: generating units

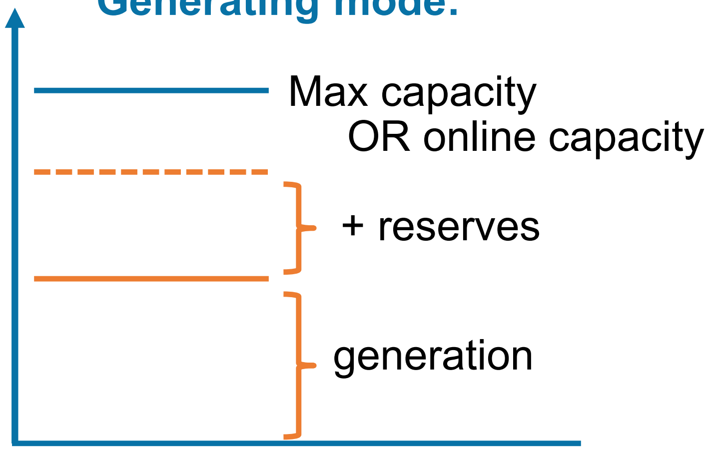
Generating units:

+ generation	[*v_gen*]
+ reserve	[*v_reserve*]
- charge	[*v_charge*]
<=	
+ capacity OR	[see orange box]
online capacity	[*v_online*]

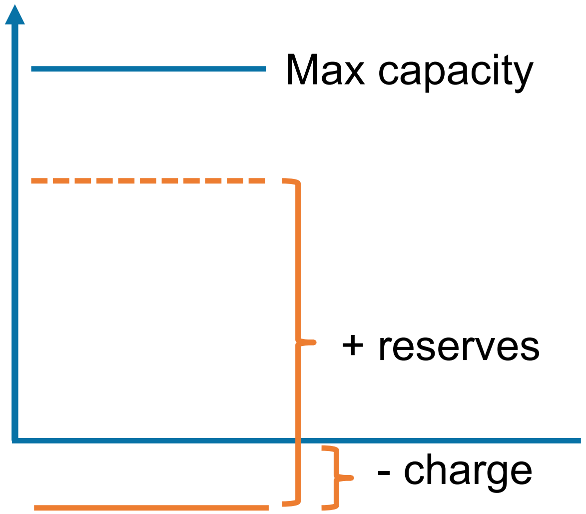
Capacity
=

+ pre-existing capacity	[units: capacity]
+ forced new capacity	[units: invested_capacity]
+ invested new capacity	[*v_invest* | *v_investTransfer*]

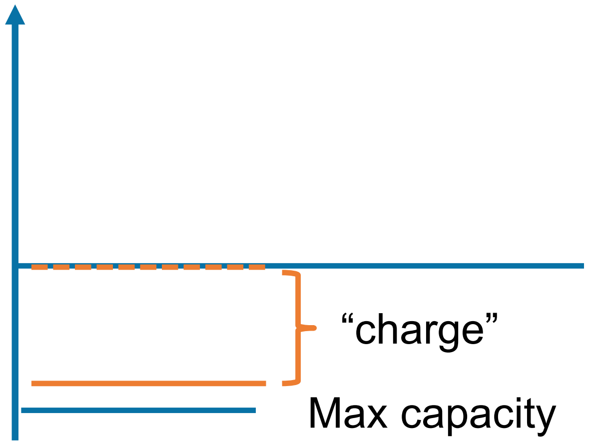
Generating mode:



Storage charging mode:



Demand increase:



Model syntax example (upward limit for online units)

```
s.t. upwardLimitOnline {(g,n,u,t) in gnut : (g,n,u) in gnu_gen && u in unit_online} :
  + v_gen[g,n,u,t]
  + (if (g,n,u) in gnu_reserve then v_reserve[g,n,u,t])
  - (if (g,n,u) in gnu_storage_charging then v_charge[g,n,u,t])
  <=
  + v_online[g,n,u,t]
;
```

Upward limit: inflow no storage, VRE

Simplified unit categories to allow faster solve time

Inflow but no storage:

+ generation [*v_gen*]
+ reserve [*v_reserve*]

<=

+ inflow time series [ts_inflow: series]

VRE units:

+ curtail [*v_curtail*]

<=

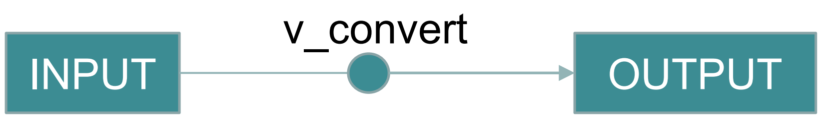
+ capacity factor [ts_cf: time series]
    × capacity [see orange box]

Capacity

=

+ pre-existing capacity [units: capacity]
+ forced new capacity [units: invested_capacity]
+ invested new capacity [*v_invest* | *v_investTransfer*]

Upward limit: conversion units

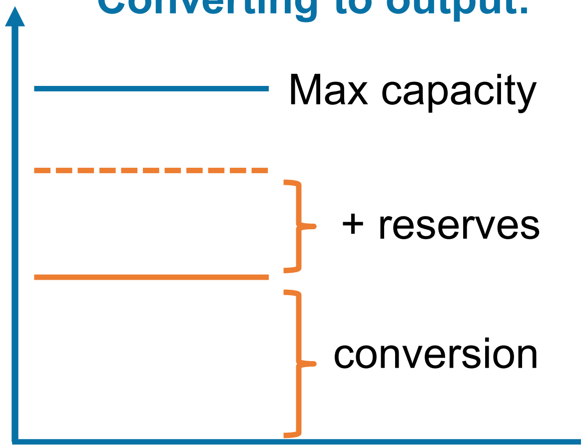


- Single variable ***v_convert*** presents both directions of the conversion
- In the input node, the **energy consumption** is equal to *v_convert* / efficiency
- The output node energy yield is just *v_convert*
- **Efficiency** can be a time series
- **Maximum capacity** is limited on the input side (units sheet capacity affects input, *e.g.*, heat pump with 100 MW capacity and 2.5 COP can generate 250 MW heat, but can be affected by efficiency time series)
- No **startups or online** for conversion units

Conversion upward limit:

+ convert	[*v_convert*]
+ reserve (to output node)	[*v_reserve*]
<=	
+ capacity	[see orange box]
OR online capacity	[*v_online*]

Converting to output:



Reserve provision

- User defines reserve capabilities of generation and conversion units in input data file, sheet unit_type

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
unit type	efficiency	min load	eff at min load	ramp up (p.u. per min)	ramp down (p.u. per min)	O&M cost/MWh	availability	**max reserve**	inertia constant (MWs/MW)	fixed cost/kW/year	inv.cost/kW	inv.cost/kWh	fixed kW/kWh ratio	conversion eff	startup cost	min uptime (h)	min downtime (h)	eff charge	self discharge loss	lifetime	interest	annuity	non synchronous
ST_coal	#	#	#	#	#	#	#	1.00	#	#	#				#	#	#			#	#	#	0
Engine_gas	#	#	#	#	#	#	#	1.00	#	#	#				#	4				#	#	#	0
CC_oil	#	#	#	#	#	#	#	1.00	#	#	#				#	5	5			#	#	#	0

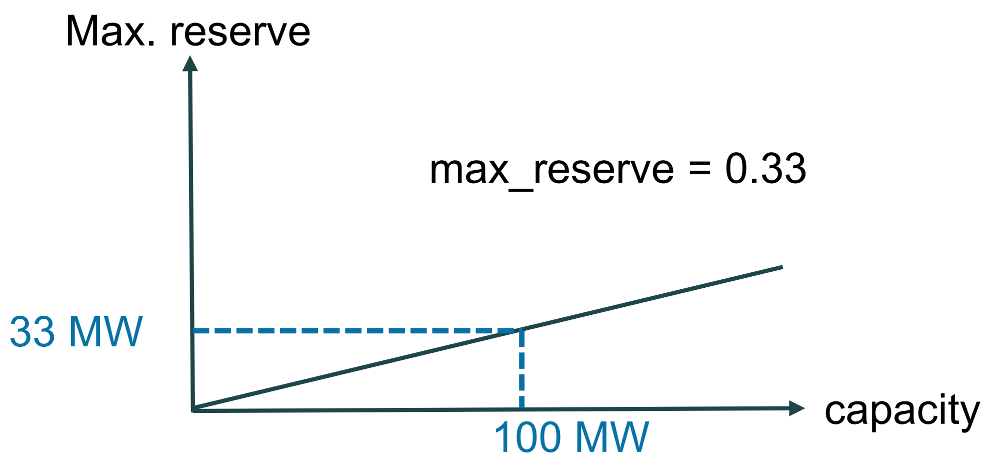
unit_type sheet

Maximum reserve provision: generating units

IRENA FlexTool

Reserve limit

+ reserve	[*v_reserve*]
<=	
+ max. reserve	[unit_type: max_reserve]
× capacity	[see orange box]
OR online	[*v_online*]

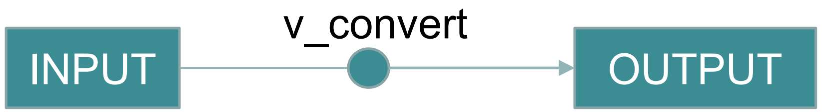


Capacity

=

+ pre-existing capacity	[units: capacity]
+ forced new capacity	[units: invested_capacity]
+ invested new capacity	[*v_invest* | *v_investTransfer*]

Maximum reserve provision: conversion units



- Conversion units **convert energy** from one type to another, *e.g.,* electricity to hydrogen or EV charger (grid electricity to car electricity)
- In g,n,u,t language, conversion unit changes energy from one grid to another
- This is different from an unit that changes energy from one node to another (transfer link)

Limit for providing reserve when converting <u>from</u> electricity:

+ reserve (input node)	[v_reserve(g,n_input,u,t)]
<=	
+ convert	[v_convert]
× max_reserve	[unit_type: max_reserve]

Limit for providing reserve when converting <u>to</u> electricity:

+ reserve (output node)	[v_reserve(g,n_output,u,t)]
<=	
+ capacity	[see orange box]
× max_reserve	[unit_type: max_reserve]

Maximum reserve provision: inflow no storage

IRENA FlexTool

Inflow time series can be used to generation or reserves

Inflow but no storage:

+ generation	[*v_gen*]
+ reserve	[*v_reserve*]
<=	
+ inflow time series	[ts_inflow: series]

Storages

IRENA FlexTool

- Deciding parameter is **‘storage (MWh)’** in units sheet
  - If ‘storage (MWh)’ has a positive value, unit has storage → Hydro_RES is storage unit and hydro_ROR is not
  - More details of storage can be given in unit_type sheet, *e.g.,* charge efficiency (eff charge), storage losses (self discharge loss), and discharge efficiency (efficiency)

A	B	C	D	E	F	G	H	I	J	K	L	M
unitGroup	**unit type**	fuel	cf profile	inflow	input grid	input node	output grid	output node	**capacity (MW)**	invested capacity (MW)	max invest (MW)	**storage (MWh)**
Battery	battery						e	n	10			10
Hydro	Hydro_RES			n			e	n	150			150000
Hydro	Hydro_ROR			n			e	n	120			0

units sheet

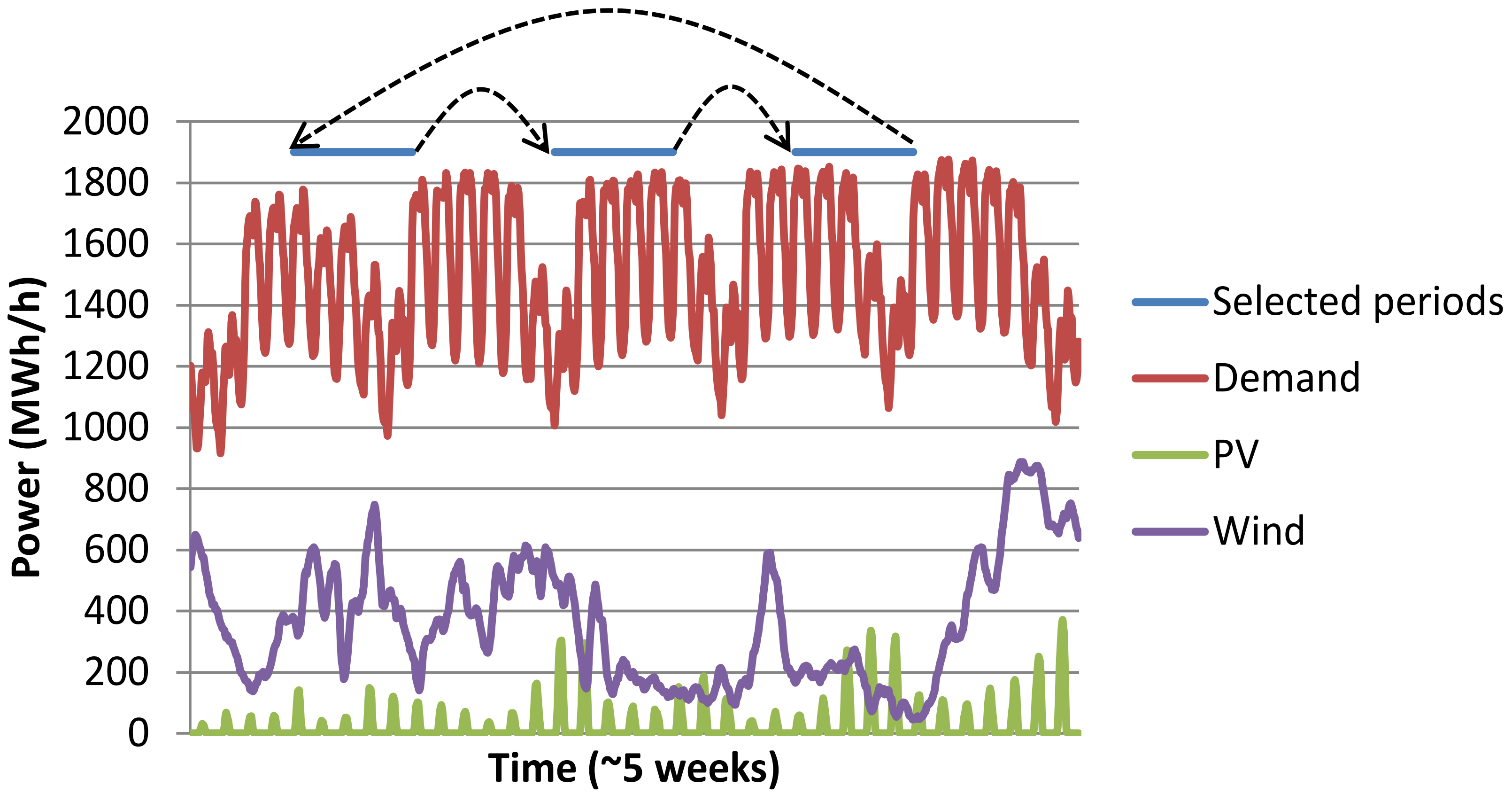
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
unit type	**efficiency**	min load	eff at min load	ramp up (p.u. per min)	ramp down (p.u. per min)	O&M cost/MWh	availability	max reserve	inertia constant (MWs/MW)	fixed cost/kW/year	inv.cost/kW	inv.cost/kWh	fixed kW/kWh ratio	conversion eff	startup cost	min uptime (h)	min downtime (h)	**eff charge**	**self discharge loss**	lifetime	interest	annuity	non synchronous
Hydro_RES	1.00			#	#		#	#	#	#													
Hydro_ROR	1.00			#	#		#	#	#	#													
battery	0.96			#	#		#	#		#		#	1					0.96	0.00040				

unit_type sheet

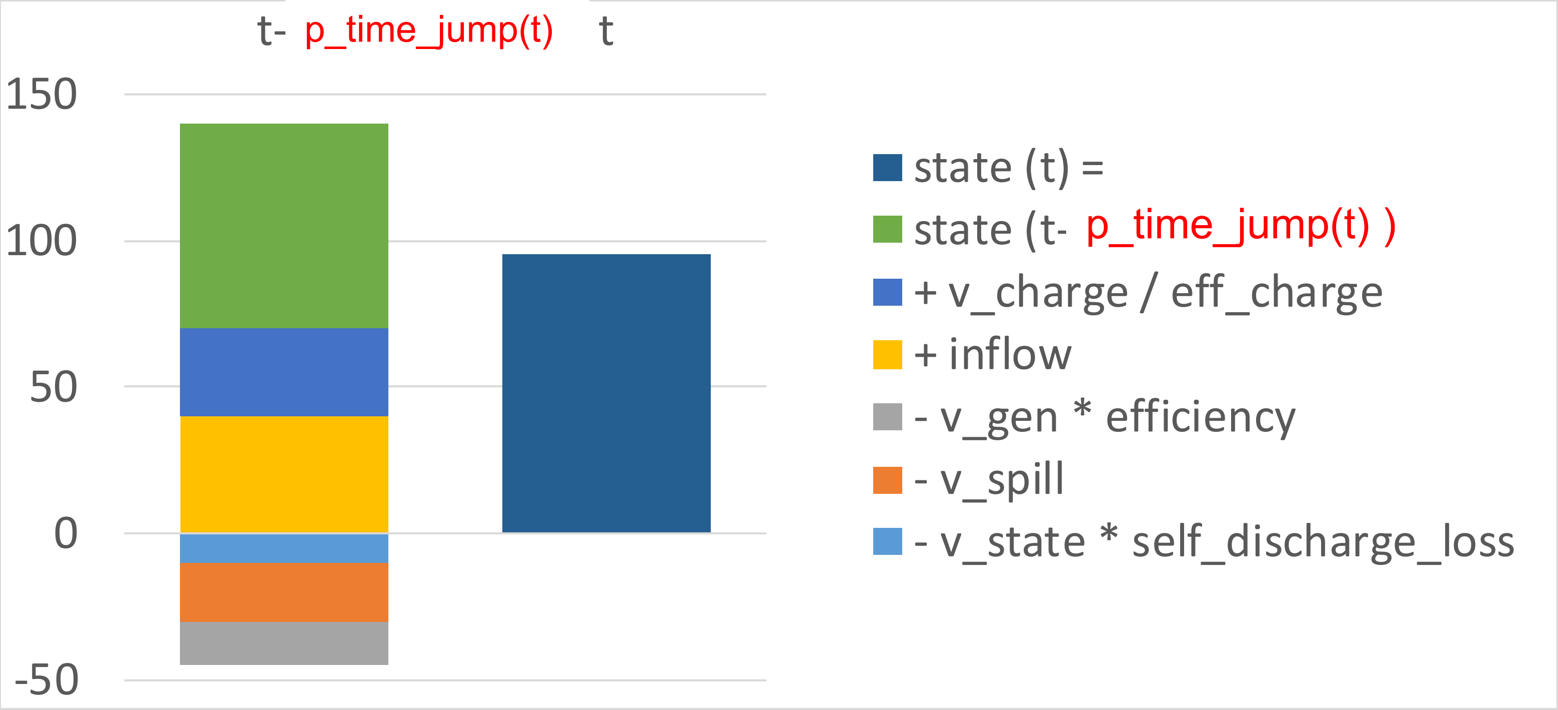
Time jump for storages

IRENA FlexTool

- It is important to maintain **storage chronology** when using jumps in time
- In full year run, hours follow other and storages are optimised correctly
- When **using jumps**, FlexTool has to follow storage values over the jumps.
- Storage value optimisation requires somewhat complicated equations when time jumps are allowed, see following slides.



Storage balance equation



Storage balance

+ storage content at time t

=

+ storage content at time t-1

+ charging

+ inflow

- discharging

- spill

- storage content * self discharge loss

Additional storage constraints

IRENA FlexTool

- Why?
  - Otherwise the model can curtail through losses without limit (linear storage model can generate and charge at the same time)

Charging limit:
+ charge [*v_charge*]
<=
+ capacity OR online [units: capacity / *v_online*]

Generating limit:
+ generation [*v_gen*]
<=
+ capacity OR online [units: capacity / *v_online*]

Storage start state
+ state in first time step [*v_state*]
=
+ parameter storage start [units: storage_start]

Storage finish state
+ state in the last time step [*v_state*]
=
+ parameter storage finish [units: storage_finish]

Activating online variable: related parameters

IRENA FlexTool

- User activates **online variables** in input data, sheet master (use_online = 1)
- Minimum load, efficiencies, startup costs, and uptime constraints are unit type parameters (input data, sheet unit_type)

parameter	value
co2_cost	10
loss_of_load_penalty	10000
loss_of_reserves_penalty	20000
lack_of_inertia_penalty	30000
curtailment_penalty	20
lack_of_capacity_penalty	5000
time_in_years	1.000
time_period_duration	60
reserve_duration	0.50
use_capacity_margin	1
use_online	1
use_ramps	0
use_non_synchronous	1
use_inertia_limit	0
mode_invest	0
mode_dispatch	1
print_duration	0
print_durationRamp	0
print_unit_results	0

master sheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	**unit type**	**efficiency**	**min load**	**eff at min load**	ramp up (p.u. per min)	ramp down (p.u. per min)	O&M cost/MWh	availability	max reserve	inertia constant (MWs/MW)	fixed cost/kW/year	inv.cost/kW	inv.cost/kWh	fixed kW/kWh ratio	conversion eff	**startup cost**	**min uptime (h)**	**min downtime (h)**	eff charge	self discharge loss	lifetime	interest	annuity	non synchronous
2	ST_coal	0.28	0.40	0.23	#	#	#	#	#	#	#	#				2.00	12	12			#	#	#	0
3	Engine_gas	0.46	0.20	0.43	#	#	#	#	#	#	#	#				0.50	4				#	#	#	0
4	CC_oil	0.40	0.50	0.35	#	#	#	#	#	#	#	#				1.00	5	5			#	#	#	0

unit_type sheet

Start-up and online variables

Startup
+ startup(unit, t) [v_startup]
>=
+ online(unit, t) [v_online]
- online(unit, *previous t*) [v_online]

Online capacity is constrained
+ online [v_online]
<=
+ capacity [see orange box]

- Activating **online variable** increases costs
  - Start up costs (default value = 0)
  - Increased fuel consumption of online units (default value = full load efficiency -> no increase in fuel consumption)

Capacity
=
+ pre-existing capacity [units: capacity]
+ forced new capacity [units: invested_capacity]
+ invested new capacity [*v_invest* | *v_investTransfer*]

Minimum uptime and downtime

IRENA FlexTool

Minimum uptime	
+ online	[*v_online*(g,n,u,t)]
>=	
- sum of capacity started up during minimum uptime	[sum (t_ >= t – unittype:min_uptime & t_ < t) *v_startup*(g,n,u,t_)]

Minimum downtime	
+ online	[*v_online*(g,n,u,t)]
<=	
+ capacity	[see orange box]
- sum of capacity started up during minimum downtime	[sum (t_ >= t + 1 & t_ <= t + 1 + unittype:min_downtime) *v_startup*(g,n,u,t_)]

- **Online variable is linear**
  - Not binary
  - These limits apply to started up quantity (in MWs)

Minimum online and generation, maximum generation

IRENA FlexTool

Minimum online

+ online [*v_online*(g,n,u,t)]

>=

+ min. online limit [capacity * ts_unit: min_online]

Minimum generation

+ generation [*v_gen*(g,n,u,t)]

>=

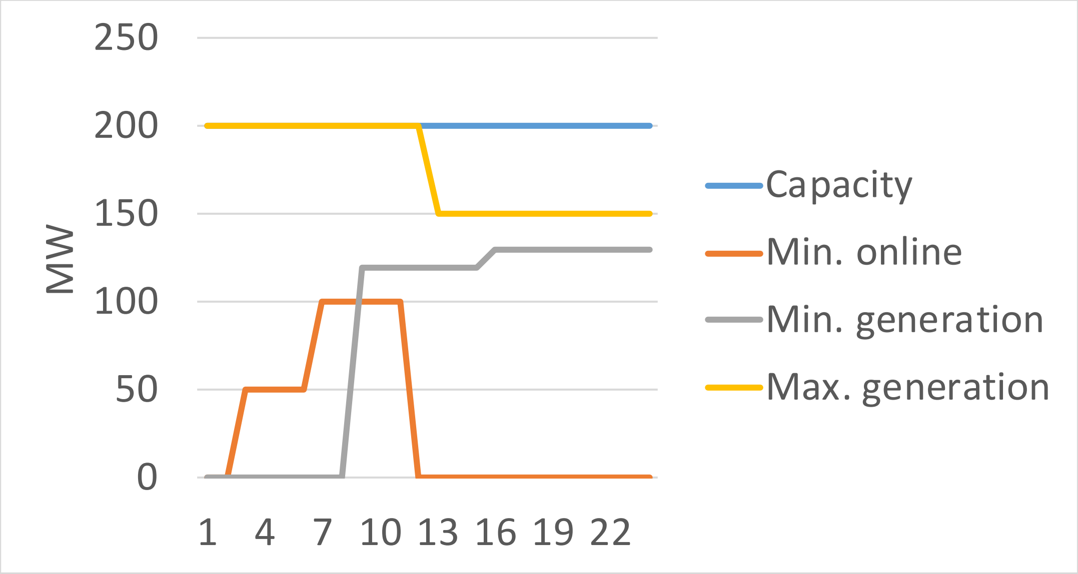
+ min. generation limit [capacity * ts_unit: min_generation]

Maximum generation

+ generation [*v_gen*(g,n,u,t)]

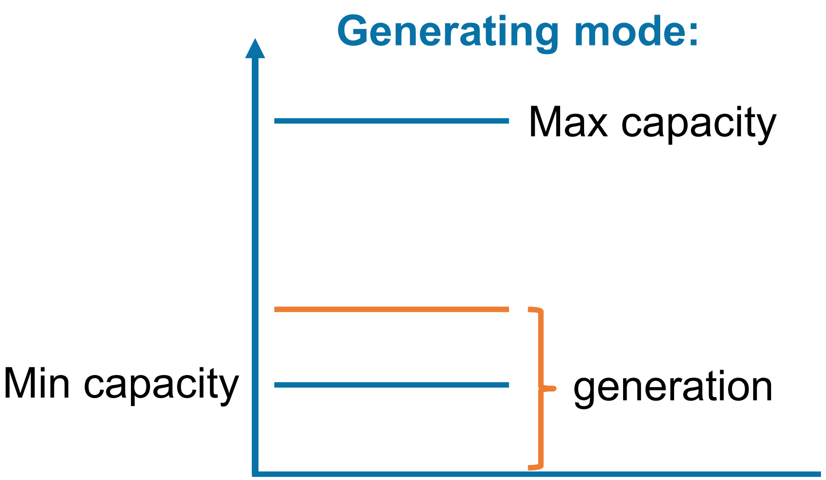
<=

+ max. generation limit [capacity * ts_unit: max_generation]



Unit constraints: minimum load

IRENA FlexTool



Minimum load:

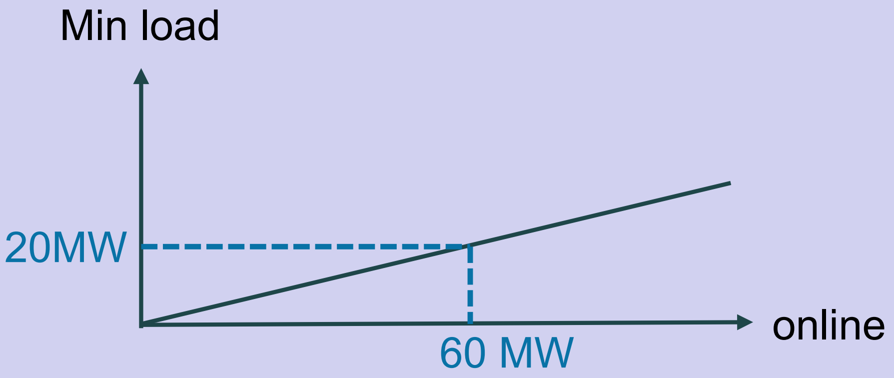
+ generation	[*v_gen*]
>=	
+ online	[*v_online*]
× min.load	[unit_type: min_load]

Example unit:

100 MW_fuel max capacity,

Min load 0.33

Efficiency 0.4, efficiency at min load 0.35



V_online	V_gen, max	Eff. (%)	V_gen, min	Eff. (%)
100 MW	100 MW	40%	33 MW	35%
60 MW	60 MW	40%	20 MW	35%
33 MW	33 MW	40%	11 MW	35%
0 MW	0 MW	-	0 MW	-

Activating ramp constraint: related parameters

IRENA FlexTool

- User activates **ramp constraint** in input data, sheet master (use_ramps = 1)
- Also adds rampRoom figures to the results

parameter	value
co2_cost	10
loss_of_load_penalty	10000
loss_of_reserves_penalty	20000
lack_of_inertia_penalty	30000
curtailment_penalty	20
lack_of_capacity_penalty	5000
time_in_years	1.000
time_period_duration	60
reserve_duration	0.50
use_capacity_margin	1
use_online	1
use_ramps	0
use_non_synchronous	1
use_inertia_limit	0
mode_invest	0
mode_dispatch	1
print_duration	0
print_durationRamp	0
print_unit_results	0

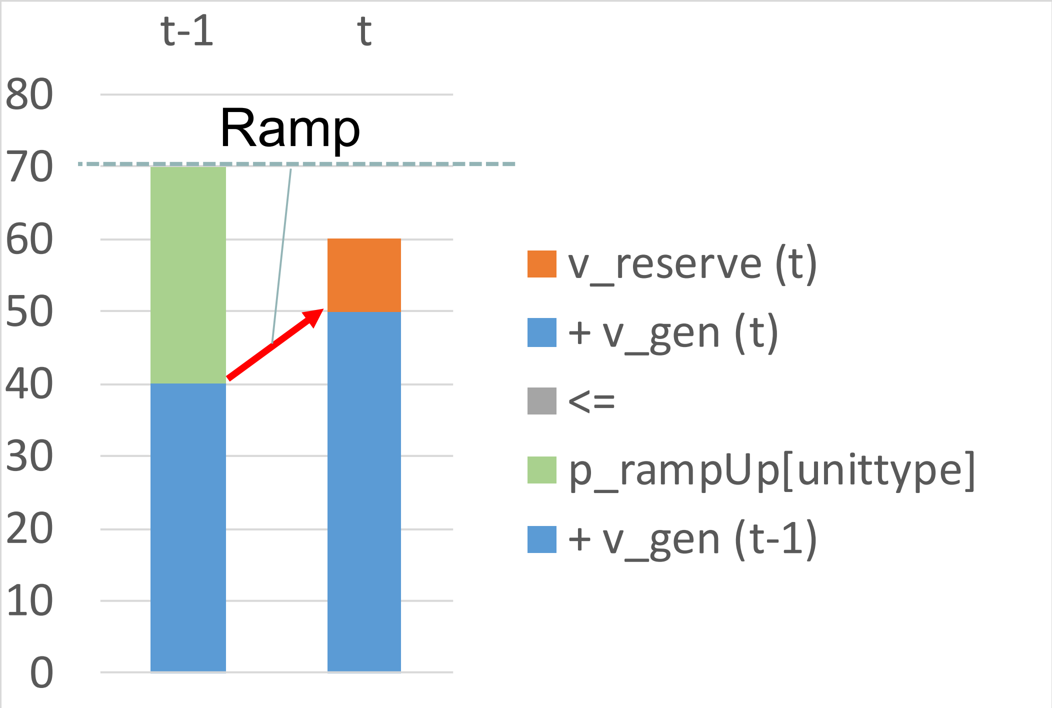
master sheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	**unit type**	efficiency	min load	eff at min load	**ramp up (p.u. per min)**	**ramp down (p.u. per min)**	O&M cost/MWh	availability	max reserve	inertia constant (MWs/MW)	fixed cost/kW/year	inv.cost/kW	inv.cost/kWh	fixed kW/kWh ratio	conversion eff	startup cost	min uptime (h)	min downtime (h)	eff charge	self discharge loss	lifetime	interest	annuity	non synchronous
2	ST_coal	#	#	#	0.02	0.02	#	#	#	#	#	#				#	#	#			#	#	#	0
3	Engine_gas	#	#	#	0.20	0.20	#	#	#	#	#	#				#	4				#	#	#	0
4	CC_oil	#	#	#	0.05	0.05	#	#	#	#	#	#				#	5	5			#	#	#	0
5	ST_bio	#	#	#	0.02	0.02	#	#	#	#	#	#				#	8	8			#	#	#	0

unit_type sheet

Unit ramp constraint:

+ reserve	[*v_reserve(t)*]
+ generation	[*v_gen(t)*]
<=	
+ generation in the previous time step	[*v_reserve(t-1)*]
+ upward ramp capability	[unit_type: ramp_up (0-1)]
× capacity	[units: capacity + *v_invest*]



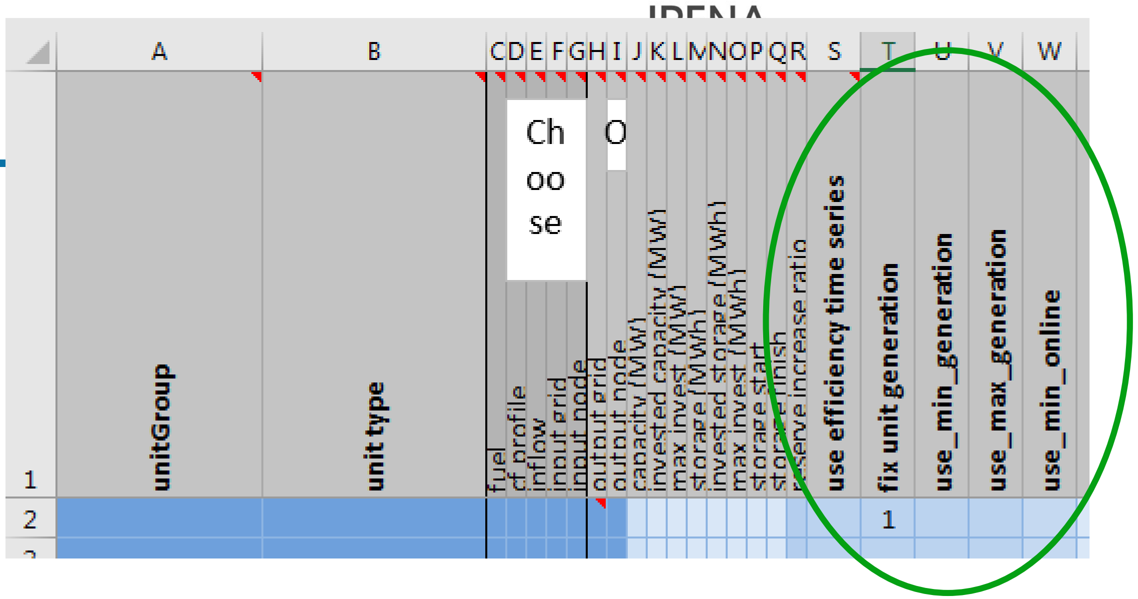
- Similar for downward ramp
- Also ramp constrained:
  - storage units
  - demand increasing units
  - conversion units
- Charging can also be ramp constrained
- For storage units maximum upward ramp could be from full charging to full discharging
  (2 ×capacity)

Time serie constraints for units

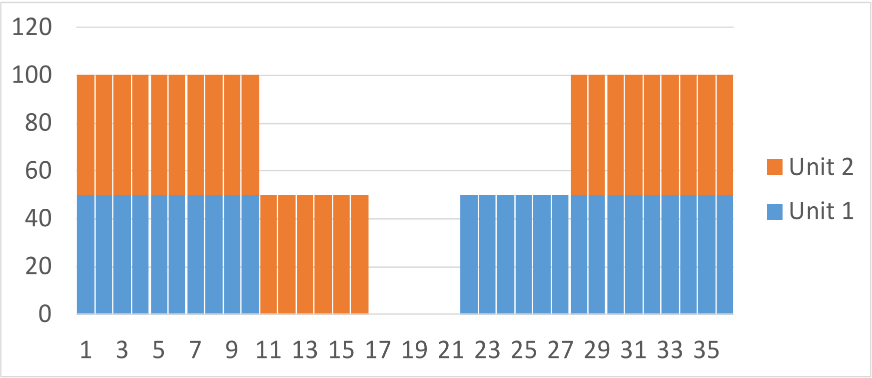
- User can give **constraints** to units as **time series**
  - Units need to be flagged to use time series in units sheet
  - Time series are given in ts_unit sheet

- Possible time series are
  - **efficiency** (works as efficiency at unit_type sheet, but has separate value for each hour)
  - **fix_generation**, **min generation, max_generation** (these fix or limit the generation. Use values from 0-1 as a share of max geration)
  - **min_online** (this sets a minimum value for unit online variable, see slide 51. Use values from 0-1.)

- The use of all these are demonstrated in template.xlsm



Example of two units with overlapping outages



Units with fixed generation:

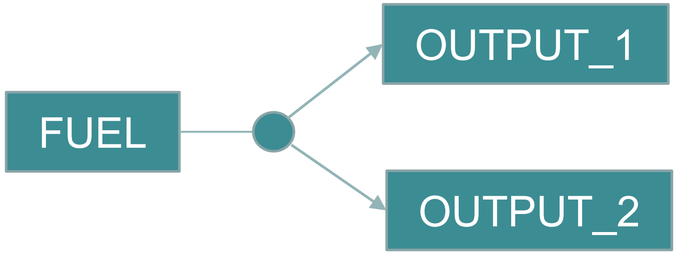
+ generation [*v_gen*]

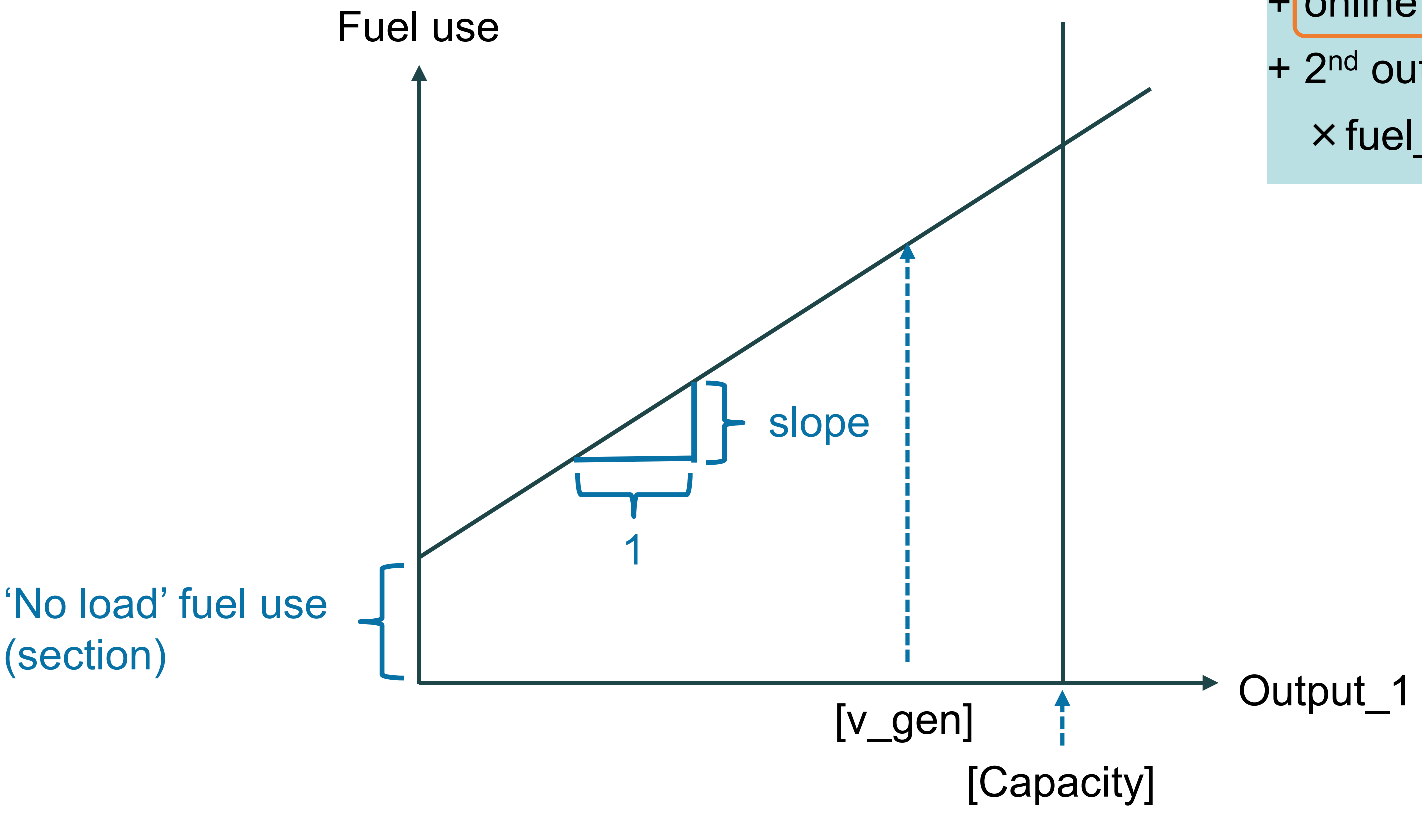
=

+ time series [ts_unit: series]

Units with two outputs

IRENA FlexTool

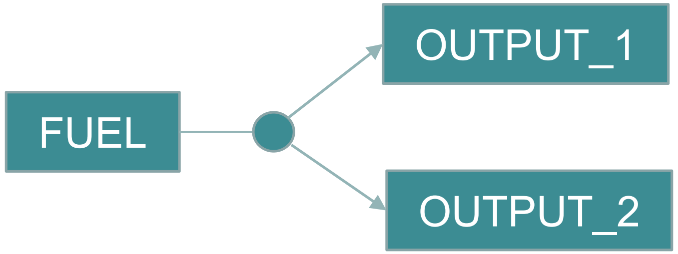




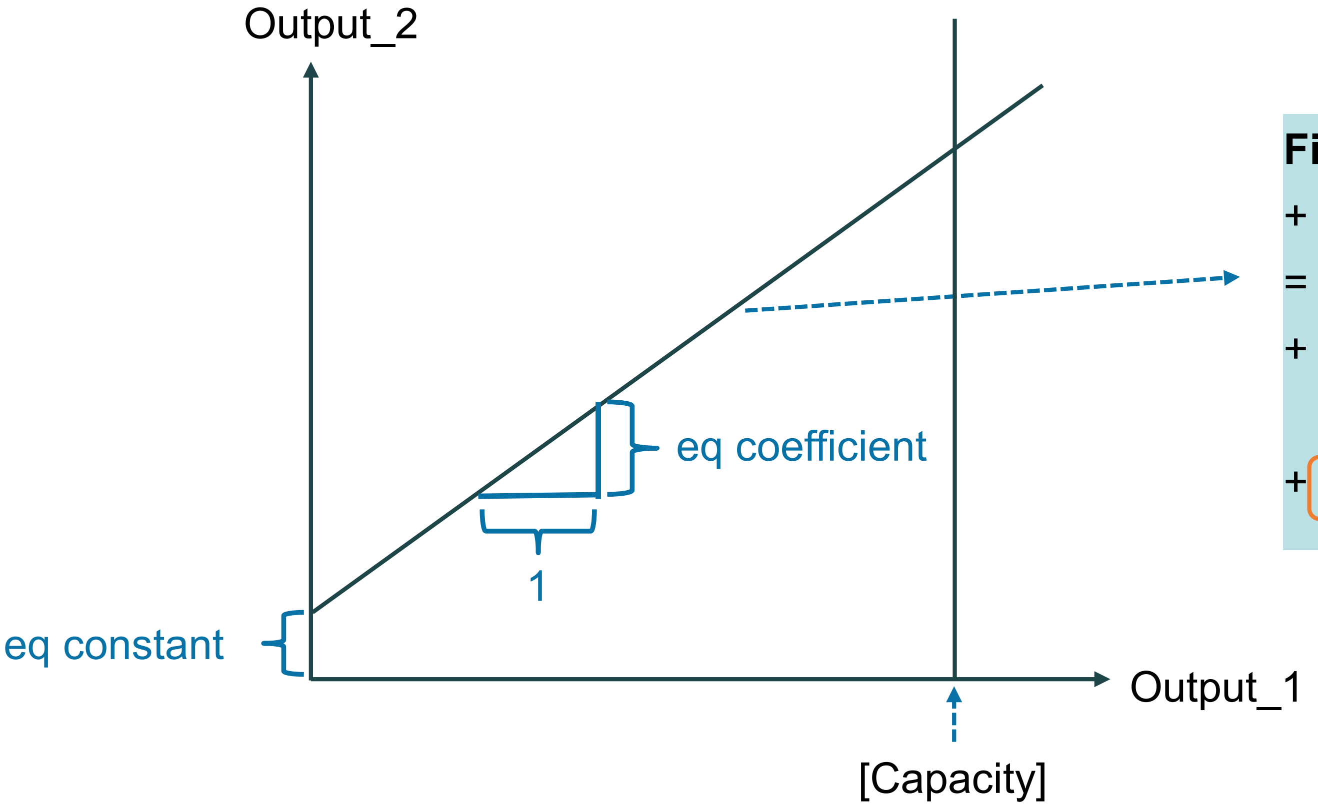
Fuel use:

+ fuel_use	[v_fueluse]
=	
+ 1st output × slope	[v_gen(g,node1,u,t)]
+ online × section	[v_online]
+ 2nd output	[v_gen(g,node2,u,t)]
× fuel_use_increase	[units: fueluse_increase]

Units with two outputs



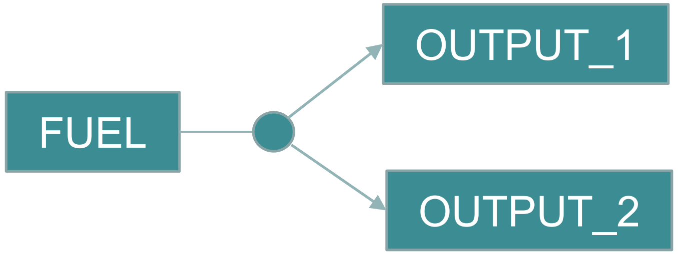
Output_2 cannot provide reserve (use output_1 for electricity)



Fixed output ratio

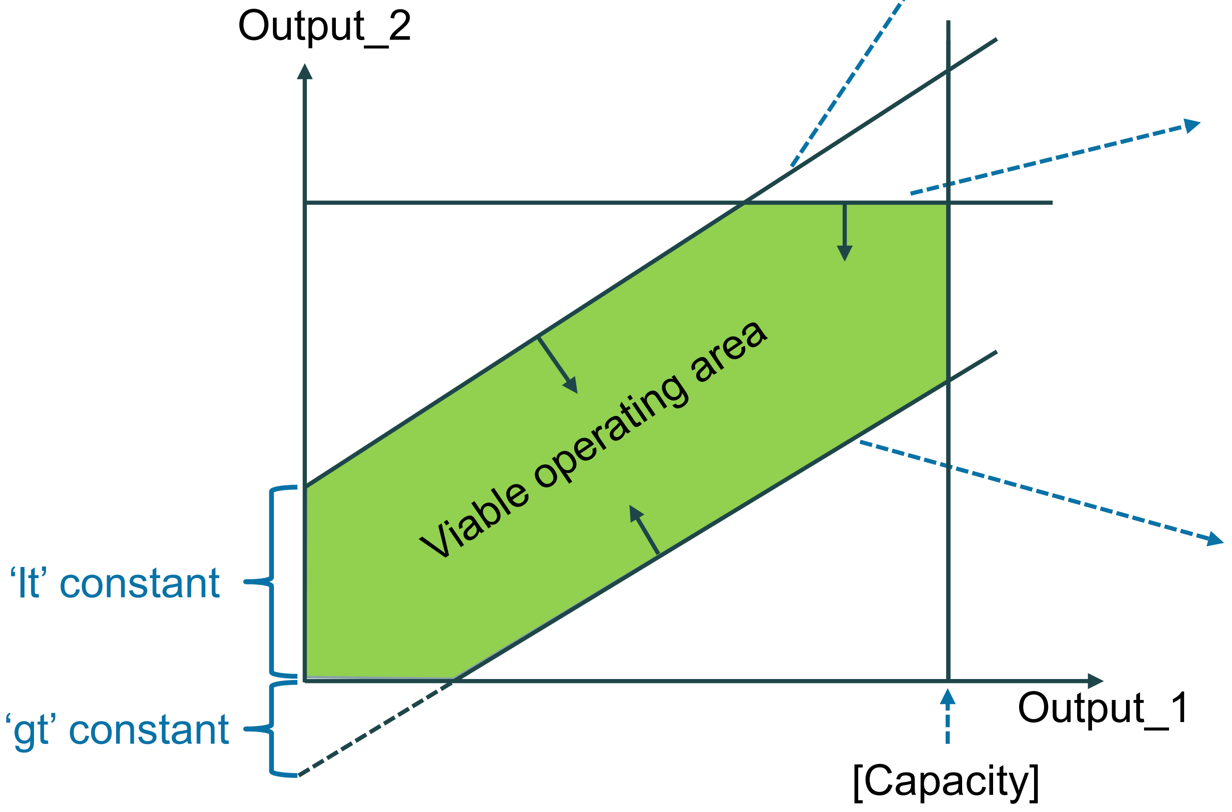
+ output_2	[*v_gen(g,node2,u,t)*]
=	
+ output_1	[*v_gen(g,node1,u,t)*]
× eq_co-efficient	[units: output2_eq_coeff]
+ eq_constant	[units: output2_eq_constant]

Units with two outputs



Less than output ratio

+ output to node 2	[v_gen(g,node2,u,t)]
<=	
+ output to node 1	[v_gen(g,node1,u,t)]
× co-efficient	[units: output2_lt_coeff
+ constant	[units: output2_lt_constant]



Upper limit for 2nd output:

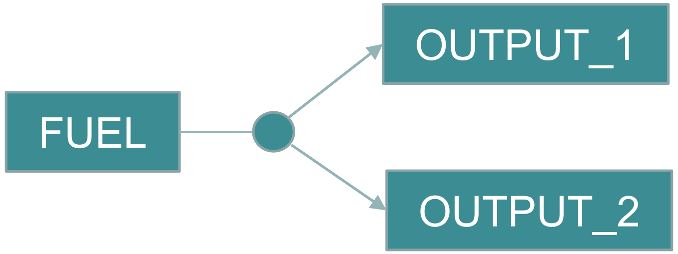
+ output to node 2	[v_gen(g,node2,u,t)]
<=	
+ ratio between outputs	[units: output2_max_capacity]
× 1st output online	[v_online(g,node1,u,t)]
OR 1st output capacity	[see orange box]

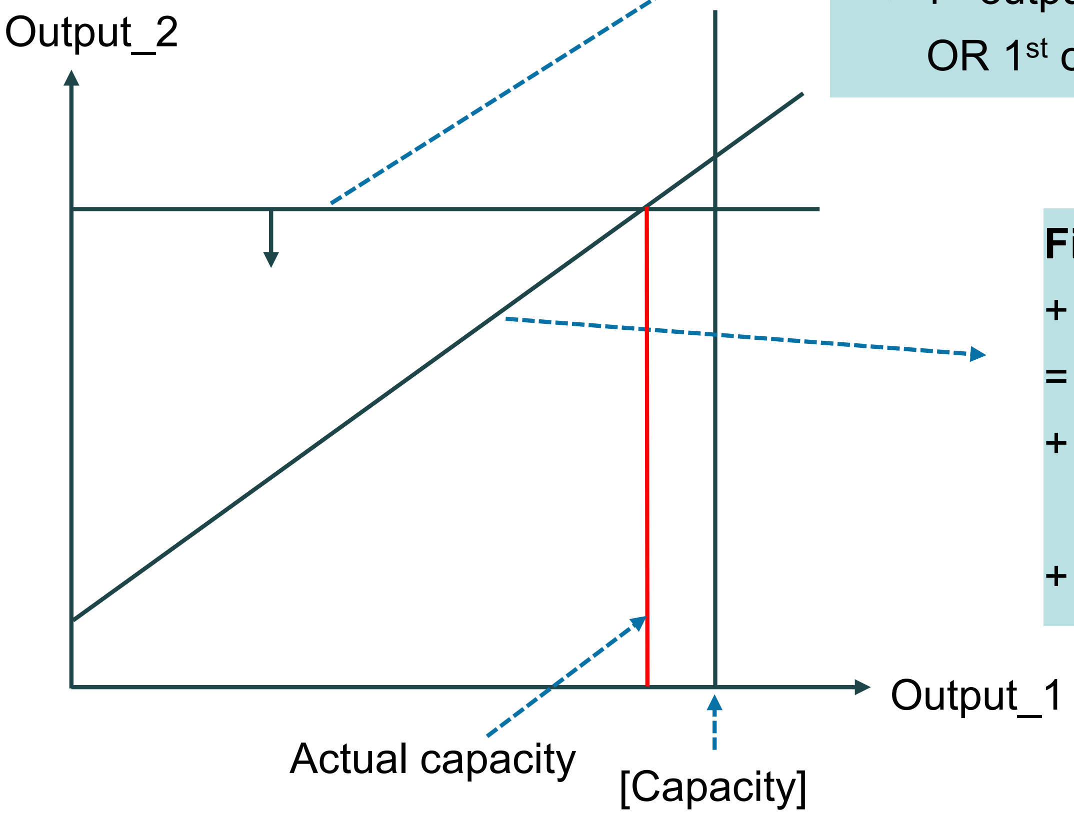
Greater than output ratio

+ output to node 2	[v_gen(g,node2,u,t)]
>=	
+ output to node 1	[v_gen(g,node1,u,t)]
× co-efficient	[units: output2_gt_coeff]
+ constant	[units: output2_gt_constant]

Units with two outputs

IRENA FlexTool





Upper limit for 2nd node:

+ output to node 2	[v_gen(g,node2,u,t)]
<=	
+ max_output_2_ratio	[units: output2_max_capacity]
× 1st output online	[v_online(g,node1,u,t)]
OR 1st output capacity	[see orange box]

Fixed output ratio

+ output to node 2	[v_gen(g,node2,u,t)]
=	
+ output to node 1	[v_gen(g,node2,u,t)]
× co-efficient	[units: output2_eq_coeff]
+ constant	[units: output2_eq_constant]

Invest Run

Back to contents

Invest run

IRENA FlexTool

- **Invest mode** is activated from master sheet in input data (mode_invest = 1)
  - Both invest and dispatch can be active, or only either
- **Capacity margin** approximates reserves during invest run

parameter	value
co2_cost	10
loss_of_load_penalty	10000
loss_of_reserves_penalty	20000
lack_of_inertia_penalty	30000
curtailment_penalty	20
lack_of_capacity_penalty	5000
time_in_years	1.000
time_period_duration	60
reserve_duration	0.50
use_capacity_margin	1
use_online	1
use_ramps	0
use_non_synchronous	1
use_inertia_limit	0
mode_invest	0
mode_dispatch	1
print_duration	0
print_durationRamp	0
print_unit_results	0

master sheet

	A	B	C	D	E	F	G
1	nodeGroup	capacity margin (MW)	non synchronous share	inertia limit (MWs)	use ts_reserve	use dynamic reserve	color in results
2	mainLand	120					

nodeGroup sheet

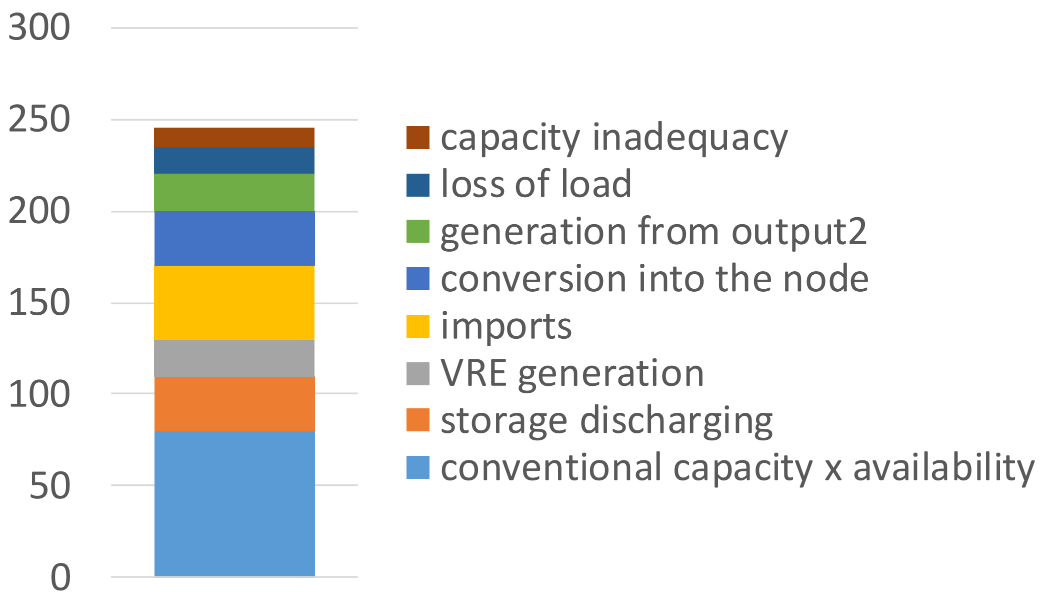
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	grid	node	nodeGroup	G	G	demand (MWh)	import (MWh)	capacity margin (MW)	non synchronous share	use ts_reserve	use dynamic reserve	print results	color in results
2	elec	nodeA	mainLand					35					
3	elec	nodeB	mainLand					10					
4	elec	nodeC	mainLand					20					
5	elec	nodeD						5					

gridNode sheet

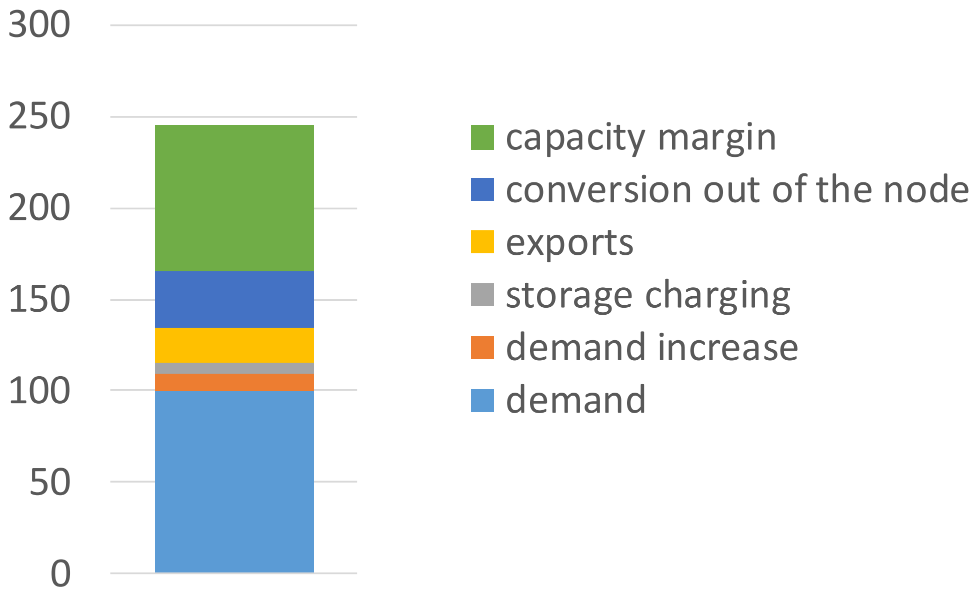
Capacity margin

IRENA FlexTool

For each time step:



>=



Capacity margin can be applied either to nodes or to groups of nodes

Limits for capacity investment

IRENA FlexTool

- FlexTool allows following **definitions to investments**
  - Max invest (MW or MWh) – maximum investment allowed, works only with invest mode
  - Min invest (MW or MWh) – minimum investment required, works only with invest mode, unit groups only
  - Invested capacity (MW) – predefined invested capacity, works both dispatch and invest mode
  - Invested storage (MWh) – predefined invested storage, works both dispatch and invest mode
- Define **fixed kW/kWh ratio** for storages in unitType sheet
  - Two of the three should be provided: [inv. cost MW], [inv. cost MWh] and [fixed kW/kWh ratio]

Multiple constraints

- User can define multiple investment constraints
- FlexTool will always follow all constraints
- However, conflicting constraints will make the model infeasible (crash)
- See slide comments for examples

	A	B	C	D	E	F	G
1	unitGroup	max invest MW	min invest MW	max invest MWh	min invest MWh	print results	color in results
2	Coal					1	
3	Oil					1	
4	Gas					1	
5	Bio					1	
6	Hydro					1	
7	Wind					1	
8	PV					1	
9	Battery					1	

unitGroup sheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	unitGroup	unit type	fuel	cf profile	inflow	input grid	input node	output grid	output node	capacity (MW)	invested capacity (MW)	max invest (MW)	storage (MWh)	invested storage (MWh)	max invest (MWh)
2	Coal	ST_coal	oal					eled				100			
3	Oil	CC_oil	oil					eled				50			
4	Bio	ST_bio	mass					eled							
5	Wind	wind	vind_A					eled				0			
6	PV	PV		PV				eled							
7	Battery	battery						eled							100

units sheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	grid	node1	node2	cap. rightward (MW)	cap. leftward (MW)	invested capacity (MW)	max invest (MW)	loss	inv.cost /kW	lifetime	interest	annuity	HVDC	color in results
2	elec	nodeA	nodeB				0							
3	elec	nodeB	nodeC				0							

nodeNode sheet

Costs

Back to contents

Objective function: minimise costs

IRENA FlexTool

- Each model run minimises costs of following equation:

Capacity
=
+ pre-existing capacity [units: capacity]
+ forced new capacity [units: invested_capacity]
+ invested new capacity [*v_invest* | *v_investTransfer*]

Operation	+ **fixed** operation and maintenance costs	[capacity × unittype: fixed_cost]
	+ **variable** operation and maintenance costs	[*v_gen* | *v_charge* | *v_convert* × unittype: O&M_cost]
	+ **fuel** costs of units	[*v_fuelUse* × fuel: fuel_price]
	+ **CO2** emission costs	[*v_fuelUse* × fuel: CO2_content × master: CO2_cost]
	+ **start-up** costs	[*v_startup* × unittype: startup_cost]
Penalties	+ penalty cost for **loss of load**	[*v_slack* × master: loss_of_load_penalty]
	+ penalty cost for **insufficient upward reserves**	[*v_reserveSlack* × master: loss_of_reserves_penalty]
	+ penalty cost for **insufficient capacity margin**	[*v_capacitySlack* × master: lack_of_capacity_penalty]
	+ penalty cost for **curtailment of VRE**	[*v_curtail* × master: curtailment_penalty]
	+ penalty cost for **insufficient inertia**	[*v_inertiaSlack* × master: lack_of_inertia_penalty]
Investment	+ **unit investment** costs	[*v_invest* × unit_type: inv.cost_kW × annuity]
	+ **storage investment** costs	[*v_investStorage* × unit_type: inv.cost_kWh × annuity
	+ **transmission line investment** costs	[*v_investTransfer* × nodeNode: inv.cost_kW × annuity]

Cost parameters

- Cost parameters are defined in:

parameter	value
co2_cost	10
loss_of_load_penalty	10000
loss_of_reserves_penalty	20000
lack_of_inertia_penalty	30000
curtailment_penalty	20
lack_of_capacity_penalty	5000

master sheet

Penalties

fuel	fuel (price/MWh)	CO2 content (t/MWh)
coal	10.00	0.34
nat_gas	15.00	0.20

fuel sheet

Operation

unit type	efficiency	min load	eff at min load	ramp up (p.u. per min)	ramp down (p.u. per min)	O&M cost/MWh	availability	max reserve	inertia constant (MWs/MW)	fixed cost/kW/year	inv.cost/kW	inv.cost/kWh	fixed kW/kWh ratio	conversion eff	startup cost	min uptime (h)	min downtime (h)	eff charge	self discharge loss	lifetime	interest	annuity	non synchronous
ST_coal	#	#	#	#	#	4.0				50	1200									30	0.08	0.089	0
Engine_gas	#	#	#	#	#	2.0				30	600									30	0.08	0.089	0
CC_oil	#	#	#	#	#	2.5				50	800									30	0.08	0.089	0
ST_bio	#	#	#	#	#	4.0				50	1200									30	0.08	0.089	0
Hydro_RES	#			#	#					20										30	0.08	0.089	0
Hydro_ROR	#			#	#					20										30	0.08	0.089	0
wind	#			#	#					20	1300									30	0.08	0.089	1
PV	#			#	#					10	700									30	0.08	0.089	1
battery	#			#	#					20		80								15	0.08	0.117	1

unitType sheet

Operation + investment

grid	node1	node2	cap.rightward (MW)	cap.leftward (MW)	invested capacity (MW)	max invest (MW)	loss	inv.cost/kW	lifetime	interest	annuity	HVDC	color in results
elec	nodeA	nodeB						100	50	0.08	0.082	0	
elec	nodeB	nodeC						100	50	0.08	0.082	0	

nodeNode sheet

Operation + Investment

Operation costs

IRENA FlexTool

- **Fixed operation costs**:
  - Fixed O&M
- **Variable operation costs**:
  - Variable O&M, fuel, CO_2 cost, startup costs
  - Scaled to annual level if running less than full year
- **Scaling to annual costs** is done based on
  - Amount of active timesteps
  - Total number of timesteps
  - Time_in_years parameter at master sheet

Fixed costs =

+ capacity x fixed O&M

Capacity =
+ pre-existing capacity [units: capacity]
+ forced new capacity [units: invested_capacity]
+ invested new capacity [*v_invest* | *v_investTransfer*]

Variable costs =

(+ generation x O&M cost
+ fuel use x fuel price
+ charge x O&M cost
+ fuel use x fuel CO2 content x CO2 cost
+ number of start ups x capacity x startup cost
) * scaled to annual

Penalties costs

IRENA FlexTool

- The model tries to avoid very high **penalty values**
  - Seeing loss of load in result is a sign of significant flexibility issue
  - Curtailment penalty should be much lower than loss of load penalty. Default value is 20, but it could also be close to zero.

	A	B
1	parameter	value
2	co2_cost	10
3	loss_of_load_penalty	10000
4	loss_of_reserves_penalty	20000
5	lack_of_inertia_penalty	30000
6	curtailment_penalty	20
7	lack_of_capacity_penalty	5000

Penalties =

+ penalty cost for **loss of load**	[*v_slack* × master: loss_of_load_penalty]
+ penalty cost for **insufficient upward reserves**	[*v_reserveSlack* × master: loss_of_reserves_penalty]
+ penalty cost for **insufficient capacity margin**	[*v_capacitySlack* × master: lack_of_capacity_penalty]
+ penalty cost for **curtailment of VRE**	[*v_curtail* × master: curtailment_penalty]
+ penalty cost for **insufficient inertia**	[*v_inertiaSlack* × master: lack_of_inertia_penalty]

Investment costs

IRENA FlexTool

In objective function:

+ **unit investment** costs
  [*v_invest*] × [unit_type: inv.cost_kW × annuity]
+ **storage investment** costs
  [*v_investStorage*] × [unit_type: inv.cost_kWh × annuity]
+ **transmission line investment** costs
  [*v_investTransfer*] × [nodeNode: inv.cost_kW × annuity]

In the [unit_type] sheet:

unit type	fixed cost/kW/year	inv.cost/kW	inv.cost/kWh	fixed kW/kWh ratio	lifetime	interest	annuity	non synchronous
ST_coal	50	1200			30	0.08	0.089	
Engine_gas	30	600			30	0.08	0.089	
CC_oil	50	800			30	0.08	0.089	
Engine_diesel	30	700			30	0.08	0.089	
Hydro_RES	20				30	0.08	0.089	
Hydro_ROR	20				30	0.08	0.089	
wind	20	1300			30	0.08	0.089	
PV	10	700			30	0.08	0.089	
battery	20		80	1.000	15	0.08	0.117	
pumpHydro	20	2500		0.020	30	0.08	0.089	

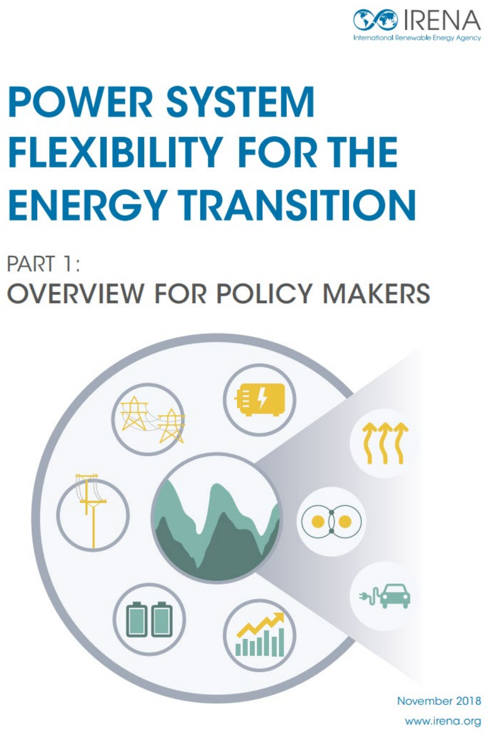
More information

Back to contents

More information

- More info from FlexTool methodology report

English



Spanish

- IRENA FlexTool Support: Flextool@irena.org www.irena.org

www.twitter.com/irena

www.facebook.com/irena.org

www.instagram.com/irenaimages

www.flickr.com/photos/irenaimages

www.youtube.com/user/irenaorg