SPD_Model_Formulation_v16.0

Model Formulation Version 16.0

SO Ref:

System Operator
23 June 2026

SYSTEM OPERATOR

TRANSPOWER

Keeping the energy flowing

NOTICE

1.1.1.1. COPYRIGHT © 2012 TRANSPOWER New Zealand LIMITED

1.1.1.2. ALL RIGHTS RESERVED

1.1.1.3. LIMITATION OF LIABILITY/DISCLAIMER OF WARRANTY

Version	Date	Change
4.4	24 June 2010	To model new CVP values for 1 May 2010 Rule Changes
5.0	24 November 2011	Major update associated with the decommissioning of pole1 and the commissioning of pole3 and removal of appendix1
6.0	21 May 2012	Update associated with implementing new PRSS, PRSL, NRSS, NRSL schedules
7.0	11 July 2012	Update associated with implementing HVDC Secondary Risk and Frequency Keeping Band into Risk
8.0	18 April 2013	Update associated with implementing the Electricity Industry Participation (Scarcity Pricing) Code Amendment 2011.
9.0	10 April 2014	Update associated with implementing Dispatchable Demand project.
10.0	1 April 2016	Update associated mainly with correction of reserve price definition and generation ramp model.
11.0	14 October 2016	Update associated with implementing National Market for Instantaneous Reserves and risk groups.
11.1	22 November 2016	Minor change to 5.4 to reflect changes to shared reserve loss modelling during DC outages
11.2	08 October 2018	No changes to the SPD software. This update aligns the formulation more closely with the software in the areas of pre-processing and post-processing.
11.3	01 July 2019	Update associated with implementing Wind Offer Arrangements
11.4	13 November 2020	Add check for $0 to replacing invalid prices after SOS1
11.5	18 May 2021	Updated for various changes associated with the Reverse Branch Limits project
11.6	24 November 2021	Added RTD load calculation. Added pre-processing for price responsive IG
12.0	15 September 2022	Changes for the release of RTP3
13.0	04 April 2023	Changes for the release of RTP4
14.0	1 November 2024	Updates to the replacement of invalid prices after SOS1
15.0	25 February 2025	Add Link Risk and account for AC Secondary Risk
16.0	27 May 2026	Add Tie Break and Battery. Update Reserve Price definition

	Position	Date
Prepared By:	System Operator	27/05/2026

TABLE OF CONTENTS

2 Introduction

This document is a mathematical description of Transpower's "Scheduling, Pricing, and Dispatch" (SPD) software used by the New Zealand Electricity Market. It has been presented using a mathematical notation designed to be rigorous but (relatively) easy to follow. There are many alternative expressions which are mathematically equivalent and will produce the same result in the market. Alternative representations may be more convenient for implementation purposes. Thus, the mathematical formulation in this document may not necessarily correspond in detail with that implemented by the developers of the SPD software. Auditing the software to verify that it is mathematically equivalent will be treated as a separate task.

The SPD software builds and solves an LP Model of the electricity market and writes out the resulting prices and quantities. After defining the Sets, Parameters and Variables, this document describes the process in three parts: 1) Pre-processing prepares the input data for use by the LP Model, 2) Constraints and the Objective Function define the LP Model, 3) Post-processing extracts the results from the solved LP Model.

3 LP Model Sets, Parameters and Variables

All variables and parameters are non-negative except where stated. There are no “soft” constraints and associated penalty variables are used to catch and help identify reasons for infeasible solutions.

3.1 Fundamental Sets

Item	**Definition**
AC Node	An ACNode is represented by an element of the set *ACNODES* and is indexed by *n*. There is a subset of *ACNODES* called *REFERENCENODES* which contains a single node from each island.
AC Line	An undirected ACLine is represented by an element of the set ACLINES indexed by k. The flow on the undirected ACLine can be positive or negative with the sign determined by a pair of ACNodes representing the $FromNode_k$ and $ToNode_k$ of the ACLine. For each undirected ACLine there is a pair of directed lines which model the directional components of the undirected flow, the directed lines are represented by elements of the set $DIRECTEDACLINES$ indexed by q.
Enode	An Electrical Node, for example a bus-section at a substation, indexed by *en*. Each Enode maps to exactly one AC Node. Enodes are included in both the Market Model and the Network Model and are used in the LP Model to map Pnodes in the Market Model to AC Nodes in the Network Model.
Pricing Node	A Pricing Node is represented by an element of the set *PNODES*, indexed by *pn*. Each Pnode maps to one or more Enodes via a $PnodeEnodeWeight_{pn}$ and this relationship is used to map prices and quantities from Pnodes to AC Nodes before the model is solved and from AC Nodes to Pnodes after the model is solved.
Island	An Island is represented by an element of the set *ISLANDS*, indexed by *i* and consisting of the two physical islands 'NI' and 'SI'.
Generation Offer	A Generation Offer is represented by an element of the set *OFFERS* and is indexed by *g*. Each offer has an associated Pnode. For reserve purposes a subset of *OFFERS* called $ISLANDRISKGENERATORS_i$ is defined. This subset is used to determine the potential risk due to generators in each island. A subset is used because not all offers represent a potential risk. For example, the total generation of a hydro station which is

Item	Definition
	represented by one offer is not a potential risk because it is generally made up of many small units that are not themselves at risk generators. For the generation ramp model a subset of *OFFERS* called *UNITGENERATORS* is defined. This subset is used because the total generation from some stations cannot be presented as one unit for ramping purpose. Jointly owned units use the ramp rate of the primary unit. For the purposes of pre-processing to restrict cleared quantities via ramp-rates, a subset of *OFFERS* called *PRICERESPONSIVEIG* is defined.
Demand Bid	A Demand Bid is represented by an element of the set *BIDS* and is indexed by *db*. *BIDS* consist of Nominated bids and Difference bids. Each bid has an associated Pnode.
Energy Scarcity Load	An Energy Scarcity Load is represented by an element of the set *ENERGYSCARCITYLOADS* and is indexed by *es*. Each Energy Scarcity Load has an associated Pnode. Energy Scarcity Load provides a price-sensitive model of *PnodeRequiredLoad,* where *PnodeRequiredLoad* is Pnode load that is not represented by a Demand Bid.
Reserve Offer	A reserve offer from a generator or interruptible load provider is represented by an element of the set *RESERVEOFFERS* and is indexed by *r*. Each reserve offer has an associated Pnode and is mapped to an ACNode using an Pnode-ACNode mapping.
HVDC Pole	An HVDC Pole represents the HVDC transmission from Benmore to Haywards including the submarine cables across the Cook Strait. Represented by the HVDCPOLES set, indexed by element *po*. Each HVDC Pole is modelled as a pair of directed HVDC Lines, one for each direction of flow.
HVDC Line	A directed HVDC Line allows for a single direction of flow on an HVDC Pole. Represented by the HVDCLINES set, indexed by element *h*. Note that unlike AC Lines, an HVDC Line does not have associated directed lines, rather the HVDC Line itself is a directed line.
HVDC Link	A directed HVDC Link represents HVDC flow between physical islands. The set consists of a NORTH link and a SOUTH link. Each HVDC Link is modelled as a pair of directed HVDC Lines that have the same direction of flow. Represented by the HVDCLINKS set, indexed by element *l*.
Reserve Type	A reserve type is represented by an element of the set *RESERVETYPES* and is indexed by element *s* *RESERVETYPES* = {*PLSR,TWD,IL*} *PLSR* is partly loaded spinning reserve which can be provided by any generator. *TWD* is tail water depressed reserve which can only be provided by hydro generators. *IL* is interruptible load which is provided by ancillary services agents.
Reserve Class	A Reserve Class is represented by an element of the set *RESERVECLASSES* and is indexed by *c*. *RESERVECLASSES* = {*Fast,Sustained*}

Item	Definition
Risk Class	A Risk Class is represented by an element of the set $RISKCLASSES_i$ and indexed by *rc*. $RISKCLASSES_i = \left\{\begin{matrix} DCCE_i, DCECE_i, ManualCE_i, ManualECE_i, \\ ACCERISKS_i, ACECERISKS_i, \\ ACCERISKGROUPS_i, ACECERISKGROUPS_i \\ ACCELINKRISKS_i, ACECELINKRISKS_i \\ HVDCSECRISKSACCE_i, HVDCSECRISKSACECE_i \\ HVDCSECRISKSMANUALCE_i, HVDCSECRISKSMANUALECE_i \end{matrix}\right\}$ $DCCE_i$ indicates the loss of a single HVDC pole. $DCECE_i$ indicates the loss of both HVDC poles. $ManualCE_i$ indicates an island's minimum CE risk. $ManualECE_i$ indicates an island's minimum ECE risk. $ACCERISKS_i$ and $ACECERISKS_i$ indicate the set of ACCE and ACECE risks associated with $g \in ISLANDRISKGENERATORS_i$ as identified in the policy statement. $HVDCSECRISKSACCE_i$ and $HVDCSECRISKSACECE_i$ indicate the HVDC secondary risks associated with an island's ACCE and ACECE risks. $HVDCSECRISKSMANUALCE_i$ and $HVDCSECRISKSMANUALECE_i$ indicate the HVDC secondary risks associated with an island's manual CE and ECE risks. $ACCERISKGROUPS_i$ and $ACECERISKGROUPS_i$ indicate the set of ACCE and ACECE risks associated with $rg \in RISKGROUPS_i$. $ACCELINKRISKS_i$ and $ACECELINKRISKS_i$ indicate the set of ACCE and ACECE risks associated with $lr \in LINKRISKS_i$
Reserve Shortfall	A Reserve Shortfall is represented by an element of the set $RESERVESHORTFALLS_{c,rc}$ indexed by Risk Class $c \in RISKCLASSES_{i,CE}$ and Reserve Class $rc \in RESERVECLASSES$.
Security Measure	A Security Measure is represented by an element of the set $SECURITY$ and is indexed by *v*. The System Operator can adjust the parameters to meet the Common Quality requirements of the Electricity Industry Participation Code.
Reserve Direction	Set indicating the direction of reserve sharing. This set is indexed by element *rd*. $RESERVEDIRECTIONS = \{Forward, Reverse\}$
Risk Group	A Risk Group represents a collection of generation and reserve offers treated as a group risk. A Risk Group is represented by an element of the set $RISKGROUPS$ and is indexed by element *rg*.
Link Risk	A Link Risk is represented by an element of the set $LINKRISKS$ indexed by element lr. A Link Risk represents a collection of one or more $ACLines_k$ whose flow in a specified direction represents export from within an area where loss of this export presents a risk to the power system outside of the area. A Link Risk definition also specifies reserve offers which are within the area, these reserves cannot be used to cover the export risk.

3.2 Derived Sets

Numerous subsets of the fundamental sets are of interest. A subscripted fundamental set represents all elements of the fundamental set having the attribute represented by the subscript.

Derived Set	Definition
$ELECTRICALISLAND_{pn}$	A group of Pnodes that are electrically connected by the AC System
$DEADACNODES_{n}$	ACNodes that are not able to make a meaningful contribution to the solution
$DEADPNODES_{pn}$	Pnodes that are not able to make a meaningful contribution to the solution
$DISCONNECTEDPNODES_{pn}$	Pnodes that are in the set of *DeadPnodes* and have a scheduled load of zero
$ISLANDPNODES_{pn}$	Pnodes belonging to physical Island i
$OFFERS_{i}$	Generation offers belonging to physical Island *i*
$OFFERS_{n}$	Generation offers associated with ACNode *n* mapped from the Pnode of the Offer via the *PnodeEnodeWeight* and *EnodeACNode* relationships
$BIDS_{n}$	Demand bids associated with ACNode *n* mapped from the Pnode of the Bid via the *PnodeEnodeWeight* and *EnodeACNode* relationships
$ACLINES_{n}$	AC lines connected to ACNode *n*
$ENERGYSCARCITYLOAD_{pn}$	Energy Scarcity Load blocks assigned to PNode *pn* by Energy Scarcity Pre-Processing
$ENERGYSCARCITYLOAD_{n}$	Energy Scarcity Load associated with ACNode *n,* mapped from $ENERGYSCARCITYLOAD_{pn}$ via the *PnodeEnodeWeight* and *EnodeACNode* relationships
$RISKGROUPS_{i}$	Risk groups in island i
$OFFERS_{rg}$	Generation offers belonging to risk group rg
$RESERVEOFFERS_{rg,c}$	Reserve offers of reserve class c belonging to risk group rg
$LINKRISKS_{i}$	Link risks in island i
$ACLINES_{lr}$	AC lines belonging to link risk lr
$RESERVEOFFERS_{lr,c}$	Reserve offers of reserve class c belonging to generators or IL associated with link risk lr
$RISKCLASSES_{i,CE}$	$RISKCLASSES_{i,CE} = \{DCCE_i, ManualCE_i, ACCERISKS_i, ACCERISKGROUPS_i, HVDCSECRISKSACCE_i, HVDCSECRISKSMANUALCE_i\}$
$RISKCLASSES_{i,ECE}$	$RISKCLASSES_{i,ECE} = \{DCECE_i, ManualECE_i, ACECERISKS_i, ACECERISKGROUPS_i,$

	$HVDCSECRISKSACECE_i$, $HVDCSECRISKSMANUALECE_i\}$
$SHORTFALLACLINES_{pn}$	ACLines which when removed from the model will reduce the capacity available to supply load at Pnode *pn*
$PNODETRANSFERPNODE_{pn}$	Pnode which has been deemed potentially suitable as a target for shortfall transfer from Pnode *pn* or as a source for price replacement transfer to Pnode *pn*
$DISPATCHABLEBIDS_{db}$	Nominated Demand Bids which are dispatchable
$DISCRETEDEMANDBIDS_{db}$	*DISPATCHABLEBIDS* which represent load that can only be on or off and will therefore be constrained to either clear fully or clear zero
$TIEBREAKOFFERS_n$	$OFFERS_n$ excluding any offer that is associated with a constraint in $SECURITY_{GenerationMaximum}$ or $SECURITY_{GenerationMinimum}$

3.3 Functions Defined on Sets

For ease of description a number of functions are defined that operate on elements of sets and return either another set or a single element. The following functions are defined:

Item	Definition
$k(\cdot)$	where the argument could be a security measure *v* or directed AC line *q* gives the undirected AC line associated with the argument.
$q(v)$	gives the directed AC line *q* of interest in security measure *v*.
$n(v)$	gives the AC node *n* of interest in security measure *v*.
$n(i)$	gives the set of AC nodes *n* located in island *i*.
$g(\cdot)$	where the argument could be a reserve offer *r* or security measure *v* gives the generation offer associated with the argument.
$db(\cdot)$	where the argument could be a reserve offer *r* or security measure *v* gives the demand bid associated with the argument.
$b(\cdot)$ and $e(\cdot)$	give the beginning and ending AC nodes respectively of a line or link where the argument could be an undirected AC line *k* or a HVDC link *l*. For the undirected AC line the conventional direction of the line is used to determine the beginning and end.
$F(k)$ and $B(k)$	give the forward and backward directed AC lines respectively associated with AC undirected line *k*.
$l(v)$	gives the HVDC link of interest in security measure *v*.

Item	Definition
$S_{AC}(n)$ and $R_{AC}(n)$	give the sets of AC directed lines for which *n* is the sending AC node or receiving AC node respectively.
$S_{HVDC}(n)$ and $R_{HVDC}(n)$	give the sets of HVDC links for which *n* is the sending AC node or receiving AC node respectively.

3.4 Generation and Load

3.4.1 *Parameters*

Item	Definition
$GenerationOfferBlocks_g$	The number of blocks in generation offer $g \in OFFERS$.
$GenerationOfferMW_{g,j}$	The MW element of the j^{th} block of the offer.
$GenerationOfferPrice_{g,j}$	The price element of the j^{th} block of the offer. *The parameter is unrestricted.*
$RampRateUp_g$	The ramping up rate in MW per hour associated with generation offer $g \in OFFERS$.
$RampRateDown_g$	The ramping down rate in MW per hour associated with generation offer $g \in OFFERS$.
$Dispatchable_g$	If this binary input flag is False then the generation offer $g \in OFFERS$ is excluded from the model
$DemandBidBlocks_{db}$	The number of blocks in demand bid $db \in BIDS$
$DemandBidMW_{db,j}$	The MW element of the j^{th} block of the bid. *The parameter is unrestricted.*
$DemandBidPrice_{db,j}$	The price element of the j^{th} block of the bid.
$Dispatchable_{db}$	If this binary input flag is False then the demand bid $db \in BIDS$ is excluded from the model
$PnodeRequiredLoad_{pn}$	Forecast MW load at Pnode *pn*. The source of *PnodeRequiredLoad* is dependent on Schedule Type
$RequiredLoad_n$	Forecast MW load at AC Node *n*, assigned from the associated Pnode.

$EnergyScarcityBlockCount_{es}$	Number of blocks in EnergyScarcityLoad $es \in ENERGYSCARCITYLOADS$
$EnergyScarcityLimit_{es,j}$	MW limit of the j^{th} block of $es \in ENERGYSCARCITYLOADS$
$EnergyScarcityPrice_{es,j}$	Price component of the j^{th} block of $es \in ENERGYSCARCITYLOADS$
$PenaltyForMWChangeOf5MinDispatch$	Small non-zero penalty price to prevent dispatch from flip-flopping between same-priced generation from one result to the next in the RTD schedule
$TieBreakSlackPrice$	Penalty price applied in the objective to the tie-break slack variables $TieBreakSlack1$ and $TieBreakSlack2$ which relax the tie-break constraint (6.1.1.11) so that the tie-break constraint only binds if it has a negligible associated cost
$CappedOfferBlockMW_{g,j}$	The MW quantity of block j of offer g capped to account for any limits that may prevent the block from fully clearing. Used by the tie-break constraint (6.1.1.11). Calculated by Tie-Break Pre-Processing.
$BatteryPair_{pn,pn'}$	Binary parameter used to map a pair of Pnodes pn and pn' that represent the same physical battery, where pn represents the battery when it is charging and pn' represents the battery when it is discharging. Used by the battery charging mode constraints 6.1.1.12 and 6.1.1.13. Calculated by Battery Pre-Processing.

3.4.2 *Variables*

Item	**Definition**
$Generation_{g}$	The scheduled part of MW generation corresponding to offer $g \in OFFERS$.
$Generation_{pn}$	$Generation_{g}$ mapped to the associated Pnode pn
$GenerationBlock_{g,j}$	The scheduled part of MW generation corresponding to the j^{th} block of the offer.
$Demand_{db}$	The scheduled part of MW load corresponding to bid $db \in BIDS$. The variable is *unrestricted*.
$Demand_{db}$	$Demand_{pn}$ mapped to the associated Pnode pn

$DemandBlock_{db,j}$	The scheduled part of MW load corresponding to the j^{th} block of the bid. The variable is *unrestricted*.
$BlockMustFullyClear_{db,j}$	Binary variable which determines whether Demand Block db,j clears fully or clears zero
$EnergyScarcityLoadCleared_{es}$	The scheduled part of MW load corresponding to $es \in ENERGYSCARCITYLOADS$
$EnergyScarcityCleared_{es,j}$	The scheduled part of MW load corresponding to the j^{th} block of $es \in ENERGYSCARCITYLOADS$
$GenerationChange1_{g}$	Use in conjunction with $GenerationChange2_{g}$ to prevent unnecessary changes in MW generation corresponding to offer $g \in OFFERS$ in the RTD schedule
$GenerationChange2_{g}$	Use in conjunction with $GenerationChange1_{g}$ to prevent unnecessary changes in MW generation corresponding to offer $g \in OFFERS$ in the RTD schedule
$TieBreakSlack1_{g,j,g',j'}$	Used in conjunction with $TieBreakSlack2_{g,j,g',j'}$ to ensure that the tie-break constraint (6.1.1.11) is only enforced if it has a negligible associated cost
$TieBreakSlack2_{g,j,g',j'}$	Used in conjunction with $TieBreakSlack1_{g,j,g',j'}$ to ensure that the tie-break constraint (6.1.1.11) is only enforced if it has a negligible associated cost
$BatteryChargingMode_{pn,pn'}$	Binary variable used by constraints 6.1.1.12 and 6.1.1.13 to enforce mutual exclusivity between charging and discharging for pnodes pn and pn' that are paired by parameter $BatteryPair_{pn,pn'}$

3.5 HVDC Transmission System

3.5.1 *Parameters*

Item	**Definition**
$HVDCLinkCapacity_{l}$	The MW capacity of HVDC link $l \in HVDCLINKS$.
$HVDCLinkFixedLosses_{l}$	The fixed losses of the link. The losses attributed to each *link* are half the fixed losses of the *pole*.
$HVDCBreakpoint$ $MWFlow_{l,bp}$	Value of power flow at the break point *bp* of HVDC Link *l*.
$HVDCBreakpoint$ $MWLoss_{l,bp}$	Value of variable (non-fixed) loss at the breakpoint *bp* in the loss curve of HVDC Link *l*.

$HVDCBreakpoints_l$	The number of breakpoints in the loss curve of HVDC Link l.

3.5.2 *Index*

Item	**Definition**
bp	Index of the break points from 1 to $HVDCBreakpoints_l$

3.5.3 *Variables*

Item	**Definition**
$HVDCLinkFlow_l$	The MW flow at the sending end scheduled for HVDC link $l \in HVDCLINKS$.
$HVDCLinkLosses_l$	The MW losses for the link.
$Lambda_{l,\ bp}$	Non-negative weight applied to breakpoint *bp* of $HVDC\ Link_l$.

3.6 AC Transmission System

3.6.1 *Parameters*

Item	**Description**
$ACLineCapacity_{k(q)}$	The MW capacity of AC line $k \in ACLINES$ in the direction associated with directed line $q \in DIRECTEDACLINES$
$ACLineAdmittance_k$	The admittance of the line. It is really the susceptance but the use of “admittance” seems to be widespread. The admittance of a line is a complex number $G - iB$ where G is the conductance and B is the susceptance. It is the susceptance which is used in the power flow that is implemented by the LP Model.
$ACLineLossBlockCount_k$	The number of blocks in the loss curve of AC line $k \in ACLINES$.
$ACLineLossBlockMW_{k(q),\ j}$	The MW limit of the j^{th} block of the loss curve of AC line $k \in ACLINES$ in the direction associated with directed line $q \in DIRECTEDACLINES$
$ACLineLossBlockFactor_{k(q),\ j}$	The loss factor of the j^{th} block of the loss curve of AC line $k \in ACLINES$ in the direction associated with directed line $q \in DIRECTEDACLINES$
$ACLineFixedLosses_k$	The fixed losses of AC line $k \in ACLINES$.

3.6.2 *Variables*

Item	Description
$ACNodeNetInjection_n$	The MW injection at node $n \in ACNODES$. *The variable is unrestricted.*
$ACNodeAngle_n$	The voltage angle at the node. *The variable is unrestricted.*
$ACLineFlow_k$	The MW flow scheduled for line $k \in ACLINES$. *The variable is unrestricted.*
$ACLineFlow_q^{Directed}$	The MW flow scheduled for directed line $q \in DIRECTEDACLINES$.
$ACLineFlowBlock_{q,j}^{Directed}$	The MW flow corresponding to the j^{th} block of the loss curve of directed line $q \in DIRECTEDACLINES$.
$ACLineLosses_q^{Directed}$	The MW losses for the directed line $q \in DIRECTEDACLINES$.
$ACLineLossesBlock_{q,j}^{Directed}$	The MW losses corresponding to the j^{th} block of the loss curve of directed line $q \in DIRECTEDACLINES$.
$ACLineFlowBinary_q^{Directed}$	Binary variable applied to flow on directed line $q \in DIRECTEDACLINES$ in order to remove circulating flows

3.7 Risk and Reserve

A generic reserve offer structure is used. Differentiation between types of reserve is achieved by using the fundamental set *RESERVETYPES* to create subsets of *RESERVEOFFERS*. Reserve is assumed to be available while ramping.

3.7.1 *Parameters*

Item	Description
$ReserveOfferBlocks_r$	The number of blocks in reserve offer $r \in RESERVEOFFERS$.
$ReserveOfferProportion_{r,j}$	The incremental MW percentage of the j^{th} block of offer $r \in RESERVEOFFERS_{PLSR}$.
$ReserveOfferPrice_{r,j}$	The price element of the j^{th} block of the offer. *The parameter is unrestricted.*
$ReserveOfferMaximum_{r,j}$	The maximum MW reserve available from the j^{th} block of the offer.
$ReserveGenerationMaximum_g$	The maximum MW generation and reserve capability associated with

Item	Description
	generation offer $g \in OFFERS$. Reference to a generation offer because maximum capability is advised via the generation offer.
$ReserveMaximumFactor_{g,\,c}$	The factor to adjust the maximum reserve of class $c \in RESERVECLASSES$ associated with generation offer $g \in OFFERS$.
$IslandRiskAdjustmentFactor_{i,\,c,\,rc}$	The risk adjustment factor for island $i \in ISLANDS$, reserve class $c \in RESERVECLASSES$ and risk class $rc \in RISKCLASSES_i$.
$IslandMinimumRisk_{i,rc}$	The minimum MW risk level for island $i \in ISLANDS$
$RiskOffsetParameter_{i,c,rc}$	Input from RMT, which accounts for HVDC frequency sharing, net free reserve, AUFLS, non-compliant generation, secondary generators risk, for island $i \in ISLANDS,$ reserve class $c \in RESERVECLASSES$ *and risk class* $rc \in RISKCLASSESS_i/\{DCCE_i\,, DCECE_i\}$
$RampupMax_i$	Input from RMT, which accounts for maximum remaining HVDC capacity following an HVDC contingency event (DCCE) for island $i \in ISLANDS$
$NetFreeReserve_{i,c,rc}$	Input from RMT, which accounts for AUFLS and non-compliant generation, for island $i \in ISLANDS$ reserve class $c \in RESERVECLASSES$ *and* risk class $rc \in \{DCCE_i, DCECE_i\}$
$HVDCSecondaryRiskSubtractor_i$	Available capacity on the HVDC pole that is not the secondary risk, for island $i \in ISLANDS$ and is available from RMT
$FKBand_g$	Frequency keeping band, set at risk generators $g \in ISLANDRISKGENERATORS_i$
$ResShareReceivedEffectiveness_{i,c,rc}$	Effectiveness factor applied to the shared reserve received in the risk island $i \in ISLANDS$ for reserve class $c \in RESERVECLASSES$ and for $rc \in \left\{\begin{matrix} ACCERISKS_i, ACECERISKS_i, ManualCE_i, ManualECE_i \\ ACCERISKGROUPS_i, ACECERISKGROUPS_i \end{matrix}\right\}$
$SharedNFRMaxLimit_{i,c}$	Limit on the SharedNFRMax$_{i,c}$ parameter for non-risk island *i* for reserve class *c*. A non-zero value is only applicable for reserve class $c \in \{Fast\}$.

Item	Description
$SharedNFRFactor$	Proportion of load in the non-risk island that provides damping which can be considered for reserve sharing.
$SharedNFRLoadOffset_i$	Load in non-risk island $i \in ISLANDS$ that does not provide load damping for shared reserves.
$ResShareControlBand_{rd}$	Reserve sharing limit on the HVDC due to HVDC modulation limit or HVDC monopole limit during reduced voltage operation mode.
$ModulationRisk_{i,rc}$	HVDC modulation risk due to frequency keeping for $i \in ISLANDS$ and $rc \in \begin{Bmatrix} DCCE_i, DCECE_i, HVDCSECRISKSAC_i, \\ HVDCSECRISKSMANUAL_i \end{Bmatrix}$
$HVDCMax_i$	HVDC transfer capability from island $i \in ISLANDS$.
$RoundPowerZoneExit_c$	MW value above which cannot guarantee the HVDC ability to reduce below the monopole minimum fast enough to provide reserve sharing in the reverse direction for reserve class $c \in RESERVECLASSES$.
$MonopoleMin$	HVDC minimum monopole transfer.
$HVDCSentFlowBreakPoint_{bp}$	Value of power flow at the breakpoint *bp* of the HVDC sent flow.
$HVDCSentLossBreakPoint_{bp}$	Value of the variable loss at the breakpoint *bp* of the HVDC sent flow.
$HVDCAfterResShareFlowBreakPoint_{rsbp}$	Value of power flow at the breakpoint *rsbp* of the HVDC flow after reserve sharing.
$HVDCAfterResShareLossBreakPoint_{rsbp}$	Value of the variable loss at the breakpoint *rsbp* of the HVDC flow after reserve sharing.
M	Large positive number, for use by the integer constraints that model reserve sharing.
$ExcessSharedNFRPenaltyPrice$	Small non-zero penalty price to prevent excess scheduling of the net free reserve variable.
$ExcessSharedIslandResPenaltyPrice$	Small non-zero penalty price to prevent excess scheduling of the island reserve sharing variable.
$ExcessResShareEffectivePenaltyPrice$	Small non-zero penalty price to prevent excess scheduling of the effective reserve variable.
$RoundPowerDisabled_c$	Binary input flag indicating if roundpower is disabled for reserve class $c \in$

Item	Description
	$RESERVECLASSES$. If roundpower is disabled then $RoundPowerDisabled_c$ = 1 otherwise $RoundPowerDisabled_c$ = 0.
$ReserveScarcityBlockCount_{i,c}$	Number of Reserve Scarcity blocks for $i \in ISLANDS$ and reserve class $c \in RESERVECLASSES$
$ReserveScarcityLimit_{i,c,j}$	MW limit of Reserve Scarcity block j for $i \in ISLANDS$ and reserve class $c \in RESERVECLASSES$
$ReserveScarcityPrice_{i,c,j}$	Block price of Reserve Scarcity block j for $i \in ISLANDS$ and reserve class $c \in RESERVECLASSES$
$ECEReserveDeficitPrice_{c}$	Price associated with using $ECEReserveDeficit_{i,c}$ to cover risk for $i \in ISLANDS$ and $c \in RESERVECLASSES$ for risk classes $rc \in RISKCLASSES_{i,ECE}$
$TowardsNode_{lr,k}$	ACNode indicating which direction of flow on ACLine k represents a risk in LinkRisk lr
$DirectionalRiskFactor_{lr,k}$	Factor associated with $k \in ACLINES$ and $lr \in LINKRISKS$ determined by pre-processing to be +1 when $TowardsNode_{lr,k}$ corresponds to $ToNode_k$ or else -1 when $TowardsNode_{lr,k}$ corresponds to $FromNode_k$
$ACSecondaryRiskMW_{g,rc}$ $ACSecondaryRiskMW_{rg,rc}$ $ACSecondaryRiskMW_{lr,rc}$	Secondary Risk MW for risk class $rc \in RISKCLASSES$ associated with generator g and any risk group rg or link risk lr associated with generator g. Included as a subtractor in the risk calculation associated with g, rg or lr because $RiskOffsetParameter_{i,c,rc}$ already accounts for secondary risk.

3.7.2 *Index*

Item	Definition
rrz	Index indicating the operational range of the HVDC for providing shared reserves in the reverse direction. $rrz \in \{RoundPowerZone, NoReverseZone, ReverseZone\}$

$rsbp$	Index of the reserve sharing break points from 1 to $ReserveShareBreakpoints$ which are used to model the HVDC flow and losses after reserve sharing.

3.7.3 Variables

Item	Description
$Reserve_r$	The reserve scheduled corresponding to reserve offer $r \in RESERVEOFFERS$.
$Reserve_{pn}$	$Reserve_r$ mapped to the associated Pnode pn
$ReserveBlock_{r,j}$	The reserve scheduled corresponding to j^{th} block of the offer.
$IslandRisk_{i,c,rc}$	The MW risk for island $i \in ISLANDS$, reserve class $c \in RESERVECLASSES$ and risk class $rc \in RISKCLASSES_i$. The variable is unrestricted (due to the risk offset the island risk variable may become negative).
$HVDCRec_i$	The total net pre-contingent HVDC flow received at island i. The variable is unrestricted, i.e. negative for export.
$RiskOffset_{i,c,rc}$	The risk offset applies for island $i \in ISLANDS$, reserve class $c \in RESERVECLASSES$ and risk class $rc \in \{DCCE_i, DCECE_i\}$
$Rampup_i$	HVDC pole rampup contribution to covering the DCCE risk.
$IslandReserve_{i,c}$	The reserve of class $c \in RESERVECLASSES$ scheduled at island $i \in ISLANDS$
$HVDCSent_i$	HVDC energy sent from island i. Calculated as the sum of flow at the sending nodes.
$SharedNFR_{i,c}$	Component of shared reserves that is provided by load damping in the non-risk island i for reserve class c.
$SharedIslandReserve_{i,c}$	Reserve cleared in the non-risk island i that could be shared for reserve class c.
$ResShareEffective_{i,c,rc}$	Reserve sharing received after adjustment for effectiveness in island i, reserve class c and risk class rc.
$ResShareSent_{i,c,rd}$	Reserve sharing provided by the non-risk island i for reserve class c in reserve sharing direction rd.
$ResShareReceived_{i,c,rd}$	Reserve sharing received by the risk island i for reserve class c in reserve sharing direction rd.
$IsSendingHVDC_i$	Binary variable indicating if island i is the sending end of the HVDC flow. 1 = Yes.
$InZone_{i,c,rrz}$	Binary variable (1 = Yes) indicating if the HVDC flow is in a zone (rrz) that facilitates the appropriate

	quantity of shared reserves in the reverse direction to the HVDC sending island *i* for reserve class *c.*
$HVDCSentLoss_{i}$	For the purposes of reserve sharing the HVDC variable losses for island *i* associated with the $HVDCSent_{i}$ flow.
$LambdaHVDCSent_{i,bp}$	Non-negative weight applied to breakpoint *bp* for island *i* sent flow and loss.
$HVDCSentAfterResShare_{i,c,rd}$	HVDC sent flow after the activation of reserves provided by reserve sharing for island *i,* reserve class *c* and reserve sharing direction *rd.* This variable is unrestricted.
$HVDCLossAfterResShare_{i,c,rd}$	HVDC sent losses associated with sent flow after the activation of reserves provided by reserve sharing for island *i,* reserve class *c* and reserve sharing direction *rd.*
$LambdaResShare_{i,c,rd,rsbp}$	Non-negative weight applied to breakpoint *rsbp* for island *i* sent flow and loss after activation of reserves provided by reserve sharing for reserve class *c* and reserve sharing direction *rd*.
$ExcessResSharePenalty$	Small non-zero penalty cost to prevent sharing of reserves in excess of benefit (“over-sharing”). This “over-sharing” occurs when the incremental cost of sharing reserves is zero.
$ReserveScarcityCleared_{c,rc,j}$	Cleared MW quantity for the j^{th} block of reserve scarcity for reserve class $c \in RESERVECLASSES$ and risk class $rc \in RISKCLASSES_{i,CE}$
$ReserveShortfall_{c,rc}$	MW quantity of risk that is not covered by a cleared Reserve Offer for $c \in RESERVECLASSES$ and $rc \in RISKCLASSES_{i,CE}$
$ECEReserveDeficit_{i,c}$	MW quantity of risk that is not covered by a cleared Reserve Offer for $i \in ISLANDS$ and $c \in RESERVECLASSES$. Applies to $rc \in RISKCLASSES_{i,ECE}$
$HVDCSentMustBeZero_{i}$	Binary variable that if 1 forces the HVDC transfer to zero and allows the HVDC Secondary Risk to not be covered. Only active when HVDC Secondary Risk is active.

3.8 Security

The System Operator may impose generation, reserve and purchase limits, and flow limits on AC and DC transmission equipment for security reasons, using the constraint forms defined in Section 6.6 to meet the requirements of the Grid Operating Security Policy.

3.8.1 *Sets*

Item	Description
$SECURITY_{GenerationMaximum}$	The set of maximum generation offer security constraints.
$SECURITY_{GenerationMinimum}$	The set of minimum generation offer security constraints.
$SECURITY_{ACLineCapacity}$	The set of all directed AC transmission line flow security constraints.
$SECURITY_{HVDCLinkCapacity}$	The set of HVDC link flow security constraints.
$SECURITY_{GroupACLinesFlow}$	The set of group AC transmission line flow security constraints.
$SECURITY_{GroupACNodesNetInjection}$	The set of all group AC node net injection security constraints.
$SECURITY_{GroupMarketNodes}$	The set of group market node security constraints on generation, demand and reserve.
$SECURITYACLINESGROUP_v$	The set of AC directed transmission lines used in a group flow security constraint for security measure *v*.
$SECURITYACNODESGROUP_v$	The set of AC node net injections used in market node group constraint for security measure *v*.
$SECURITYMARKETDEMNODESGROUP_v$	The set of demand bids used in market node group constraint for security measure *v*
$SECURITYMARKETGENNODESGROUP_v$	The set of generation offers used in market node group constraint for security measure *v*
$SECURITYMARKETRESNODESGROUP_v$	The set of reserve offers used in market node group constraint for security measure *v*

3.8.2 *Parameters*

Item	Description
$SecurityGenerationMaximum_v$	The MW generation maximum associated with security measure $v \in SECURITY_{GenerationMaximum}$ imposed on a generation offer by the System Operator for security reasons.

Item	Description
$\mathit{SecurityGenerationMinimum}_v$	The MW generation minimum associated with $v \in \mathit{SECURITY}_{\mathit{GenerationMinimum}}$.
$\mathit{SecurityACLineCapacity}_v$	The MW directed AC line capacity associated with $v \in \mathit{SECURITY}_{\mathit{ACLineCapacity}}$.
$\mathit{SecurityHVDCLinkCapacity}_v$	The MW HVDC link capacity associated with $v \in \mathit{SECURITY}_{\mathit{HVDCLinkCapacity}}$.
$\mathit{SecurityGroupACLineFlow}_v$	The MW maximum total flow of a group of directed AC lines, associated with security measure $v \in \mathit{SECURITY}_{\mathit{GroupACLinesFlow}}$. *The parameter is unrestricted.*
$\mathit{SecurityGroupACLineWeight}_q$	The weight associated with directed line $q \in \mathit{SECURITYACLINESGROUP}_v$. *The parameter is unrestricted.*
$\mathit{SecurityGroupACNodesNetInjection}_v$	The MW maximum total AC node net injection of a group of AC nodes, associated with security measure $v \in \mathit{SECURITY}_{\mathit{GroupACNodesNetInjection}}$. *The parameter is unrestricted*.
$\mathit{SecurityGroupACNodeWeight}_n$	The weight associated with AC node $n \in \mathit{SECURITYACNODESGROUP}_v$. *The parameter is unrestricted*.
$\mathit{MarketNodeDemWeight}_{db}$	The weight associated with demand bid $db \in \mathit{SECURITYMARKETDEMNODEGROUP}_v$. *The parameter is unrestricted.*
$\mathit{MarketNodeGenWeight}_g$	The weight associated with generation offer $g \in \mathit{SECURITYMARKETGENNODEGROUP}_v$. *The parameter is unrestricted.*
$\mathit{MarketNodeResWeight}_r$	The weight associated with reserve offer $r \in \mathit{SECURITYMARKETRESNODEGROUP}_v$. *The parameter is unrestricted.*
$\mathit{MarketNodeSecurityLimit}_v$	The limit associated with security measure $v \in \mathit{SECURITY}_{\mathit{GroupMarketNodes}}$. *The parameter is unrestricted.*

3.9 Mixed Constraints

This facility allows the System Operator to impose mixed constraints on any existing variables. It also provides a facility for the creation of new models. Approval for the creation of new mixed constraints is required to go through the consultation process used for Code amendments. Such consultation (and subsequent approval) relates to the form of the mixed constraints and may include specification of permanent conditions with respect to the level of, or relationships between, parameters in the constraint. Other parameters may be adjusted by specified processes. Any formulation constraint involving mixed constraint variables may also be implicitly involved.

3.9.1 *Sets*

Item	**Description**
$MIXEDCONSTRAINTS_{Type1}$	The set of all Type 1 mixed constraints. Each constraint, *m,* will normally define one new variable, $MixedConstraintVariable_m$, and can link it to any combination of existing model variables.
$MIXEDCONSTRAINTS_{Type2}$	The set of all Type 2 mixed constraints. Each Type 2 constraint is a group constraint creating links between the new variables created by Type 1 constraints.
$MIXEDVARGROUP_b$	The set of Type 1 mixed constraints whose new variables are linked by Type 2 mixed constraint *b.*
$MIXEDDEMNODEGROUP_m$	The set of demand bids used in Type1 mixed constraint *m*.
$MIXEDGENNODEGROUP_m$	The set of generation offers used in Type1 mixed constraint *m*.
$MIXEDRESNODEGROUP_m$	The set of reserve offers used in Type1 mixed constraint *m*.
$MIXEDDIRACLINEGROUP_m$	The set of AC lines whose flow used in Type1 mixed constraint *m*.
$MIXEDDIRACLINELOSSGROUP_m$	The set of AC line whose losses are used in Type1 mixed constraint *m*.
$MIXEDACFIXLOSSGROUP_m$	The set of AC lines whose fixed losses used in Type1 mixed constraint *m*.
$MIXEDDCLINEGROUP_m$	The set of DC lines whose flow is used in Type1 mixed constraint *m*.
$MIXEDDCLNLOSSGROUP_m$	The set of DC lines whose losses are used in Type1 mixed constraint *m*.
$MIXEDDCFIXLOSSGROUP_m$	The set of DC lines whose fixed are losses used in Type1 mixed constraint *m*.

3.9.2 *Parameters*

Item	Description
$MixedConstVarWeight1_{m}$	The weight associated with mixed security constraint variable $m \in MIXEDCONSTRAINTS_{Type1}$ *The parameter is unrestricted.*
$MixedConstDemWeight_{db,\ m}$	The weight associated with demand bid $db \in MIXEDDEMNODEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstGenWeight_{g,\ m}$	The weight associated with generation offer $g \in MIXEDGENNODEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstResWeight_{r,\ m}$	The weight associated with reserve offer $r \in MIXEDRESNODEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstACLineWeight_{q,\ m}$	The weight associated with the flow in directed AC line $q \in MIXEDDIRACLINEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstACLineLossWeight_{q,\ m}$	The weight associated with the variable loss in directed AC line $q \in MIXEDDIRACLINEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstACLineFixedLossWeight_{k,\ m}$	The weight associated with the fixed loss in undirected AC line *k* $l \in MIXEDACLINEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstDCLinkWeight_{l,\ m}$	The weight associated with the flow in HVDC link $l \in MIXEDDCLINEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstDCLinkLossWeight_{l,\ m}$	The weight associated with the variable loss in HVDC link $l \in MIXEDDCLINEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstDCLinkFixedLossWeight_{l,\ m}$	The weight associated with the fixed loss in HVDC link $l \in MIXEDDCLINEGROUP_{m}$. *The parameter is unrestricted.*
$MixedConstraintLimit1_{m}$	The limit associated with mixed security constraint $m \in MIXEDCONSTRAINTS_{Type1}$. *The parameter is unrestricted.*
$MixedConstVarWeight2_{m.b}$	The weight associated with mixed security constraint variable $m \in MIXEDVARGROUP_{b}$ in constraint $b \in$

Item	Description
	$MIXEDCONSTRAINTS_{Type2}$. *The parameter is unrestricted.*
$MixedConstraintLimit2_b$	The limit associated with mixed security constraint $b \in MIXEDCONSTRAINTS_{Type2}$. *The parameter is unrestricted.*

3.9.3 *Variables*

Item	Description
$MixedConstraintVariable_m$	Mixed security constraint variable defined by constraint $m \in MIXEDCONSTRAINTS_{Type1}$. *The variable is unrestricted.*

4 Pre-Processing

4.1 Overview

Pre-processing prepares the input data for the LP Model as shown by the overview in Figure 1. The following subsections describe the pre-processing steps.

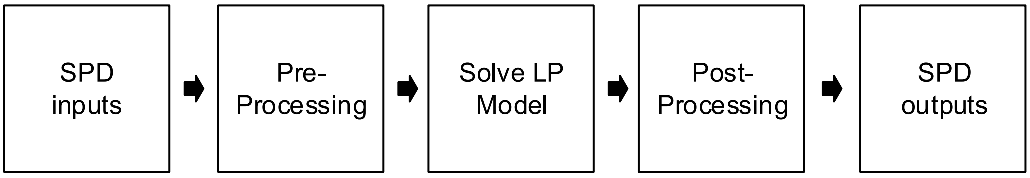


Figure 1: SPD process flow

4.2 Connectivity Pre-Processing

4.2.1 *Network Model, Market Model, LP Model*

The LP Model implements a power flow that transports electricity from generation to load as shown in Figure 2.

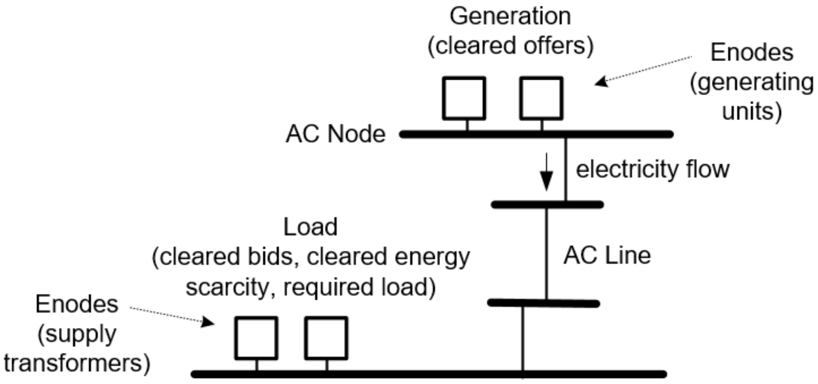


Figure 2: LP Model power flow

External topology processing produces an interval-specific Network Model by applying planned outages to a static model. The power flow in the LP Model uses the ACNodes and ACLines from this Network Model. The Market Model provides prices and quantities at Pnodes.

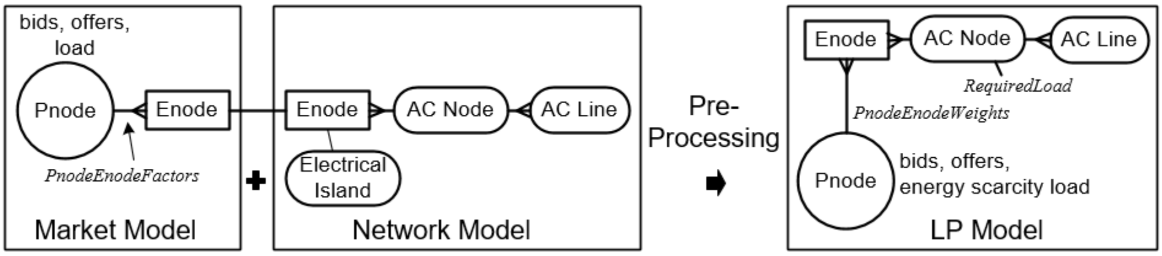


Figure 3: Connectivity pre-processing combines the Market Model and the Network Model

As shown in Figure 3, connectivity pre-processing combines inputs from the Network Model and the Market Model to create the LP Model. Within the LP Model the bid, offer and energy scarcity load quantities are mapped from Pnode to AC Node via *PnodeEnodeWeights*, while Required Load is assigned directly to the AC Node.

4.2.2 *Electrical Islands*

The external topology processing that creates the Network Model assigns an Electrical Island (represented by a number) to every Enode and ACNode. An Electrical Island is a group of Enodes that are electrically connected by the AC System, see examples in Figure 4. An ACNode is a group of Enodes that are electrically connected at a substation, i.e., electrically connected but not by an AC Line.

The topology processing defines an Electrical Island as "live" if it contains at least one generator Enode and one load Enode, thereby enabling the possibility of a result which has generation scheduled to meet load.

Live Electrical Island 1 is the NI and live Electrical Island 2 is the SI. There may be other live Electrical Islands, these will be numbered from 3 upwards (referred to as Electrical Islands > 2).

ACNodes and Enodes that are not in a live Electrical Island are assigned an island number of 0 and defined as dead. Dead AC Nodes are included in the set $DeadACNodes_n$. A Pnode is defined as dead if all of the Enodes that it maps to are dead.

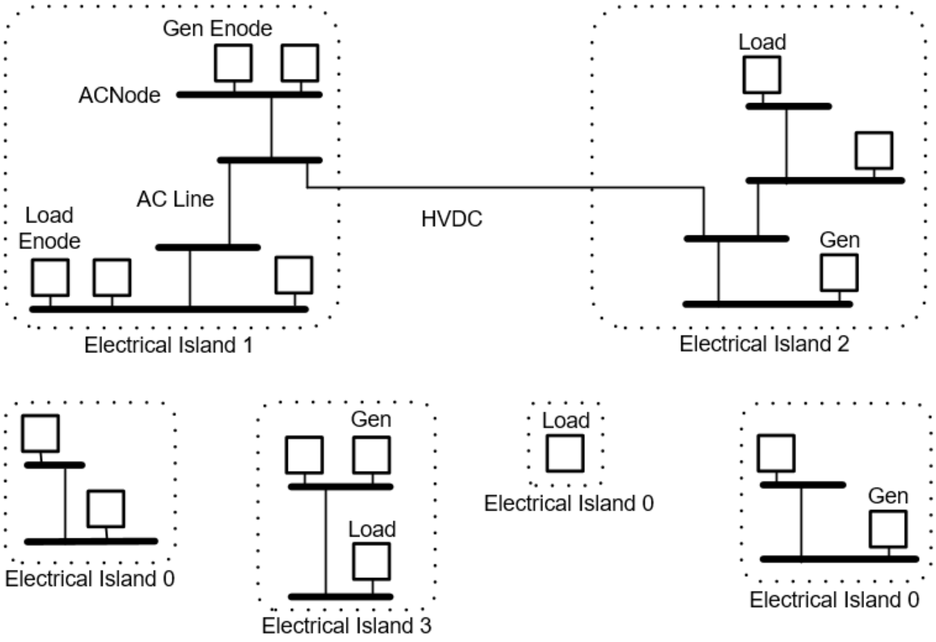


Figure 4: Electrical Islands example

4.2.3 *Input Parameters*

Parameter	**Definition**
$PnodeEnodeFactor_{pn,en}$	Market Model mapping between a Pnode and an Enode. The factor represents the capacity of the Enode relative to other Enodes that map to the Pnode. For example, a Pnode that represents a location with 20MW and 30MW supply transformers would map to two Enodes, with factors of 20 and 30 respectively (because the factors represent the relative size of the Enodes, these Enodes could also be represented by factors of 2 and 3).
$EnodeACNode_{en,n}$	Network Model mapping between an Enode and an ACNode.

4.2.4 *Calculated Parameters*

Parameter	**Definition**
$AdjustedFactor_{pn,en}$	Pnode-Enode Factors adjusted to account for the connectivity status of the Enode
$PnodeEnodeWeight_{pn,en}$	Pnode-Enode Weights calculated from the adjusted Pnode-Enode Factors

4.2.5 *Pnode-Enode-ACNode Mapping*

Pre-processing uses the $PnodeEnodeFactor_{pn,en}$ inputs from the Market Model to calculate $PnodeEnodeWeight_{pn,en}$ parameters which are used to map Pnodes to

ACNodes via Enodes. Before the factors are used in the calculation they are adjusted based on the following rules.

Rule for adjusting Pnode-Enode factor	**Reasoning (referring to examples in Figure 5 and Figure 6)**
If a Pnode maps to some dead Enodes and some live Enodes, then the Pnode-Enode Factor is set to zero for the dead Enodes.	Where a Pnode has a mix of live and dead Enodes, zeroing the factors for the dead Enodes will result in the Pnode quantities being apportioned among only the live Enodes. See Pnode F in the example. If a Pnode has a non-zero load and all the Enodes that it maps to are dead then the Enode factors will not be zeroed. The resulting uncleared load will serve to indicate a discrepancy between load and connectivity. See Pnode G in the example.
If a Pnode is dead and has no load then all of its factors will be zeroed.	See Pnode K in the example.
If a Pnode has a non-zero mapping to one or more Enodes in Electrical Island 1 or Electrical Island 2 and also maps to one or more Enodes in Electrical Island > 2, then the Pnode-Enode Factor is set to zero for the Enodes in Electrical Island > 2.	This is to prevent a Pnode from being split between two Electrical Islands, which would result in the Pnode price being set by the combination of unrelated prices from different Electrical Islands. See Pnode J in the example.

After the above rules are applied, the Pnode-Enode Weights are calculated as follows:

4.2.5.1. $$PnodeEnodeWeight_{pn,en} = \frac{AdjustedFactor_{pn,en}}{\sum_{pn,en} AdjustedFactor_{pn,en}}$$

$$\forall pn \in Pnodes, \forall en \in Enodes$$

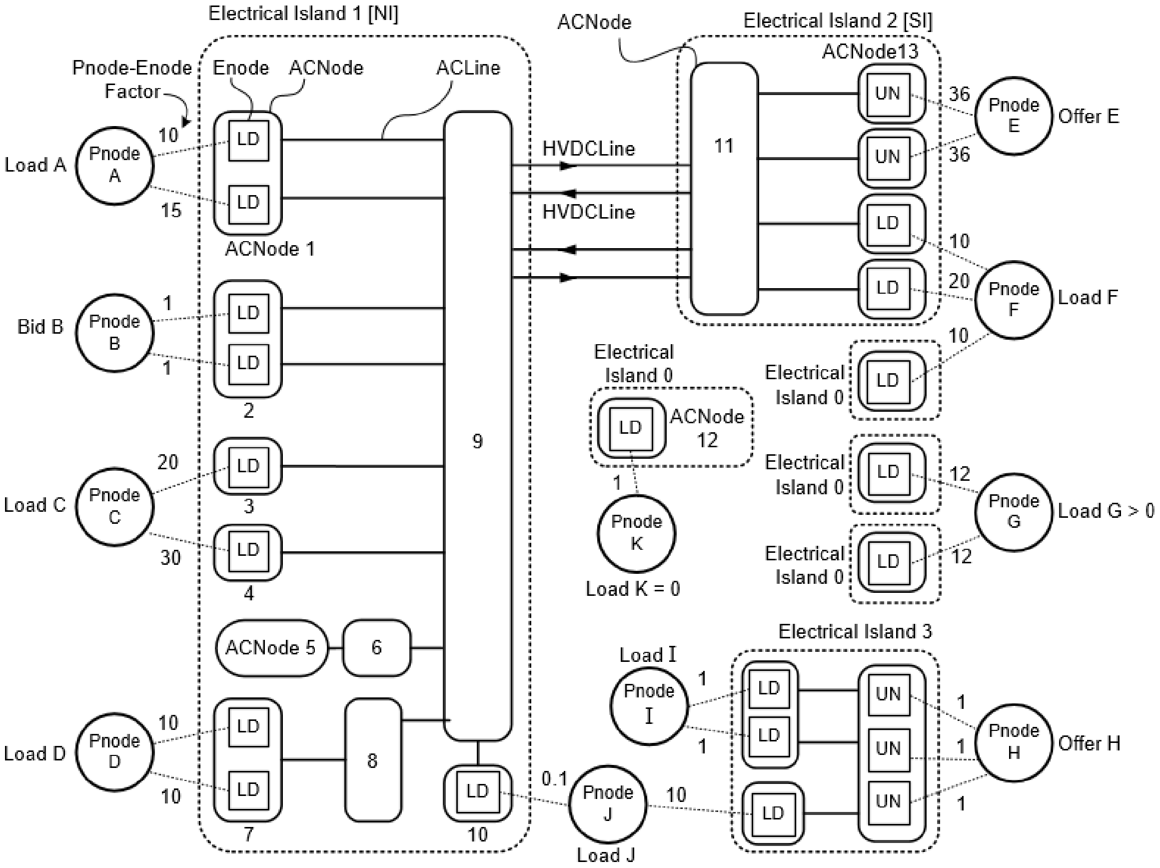


Figure 5: Example of SPD inputs

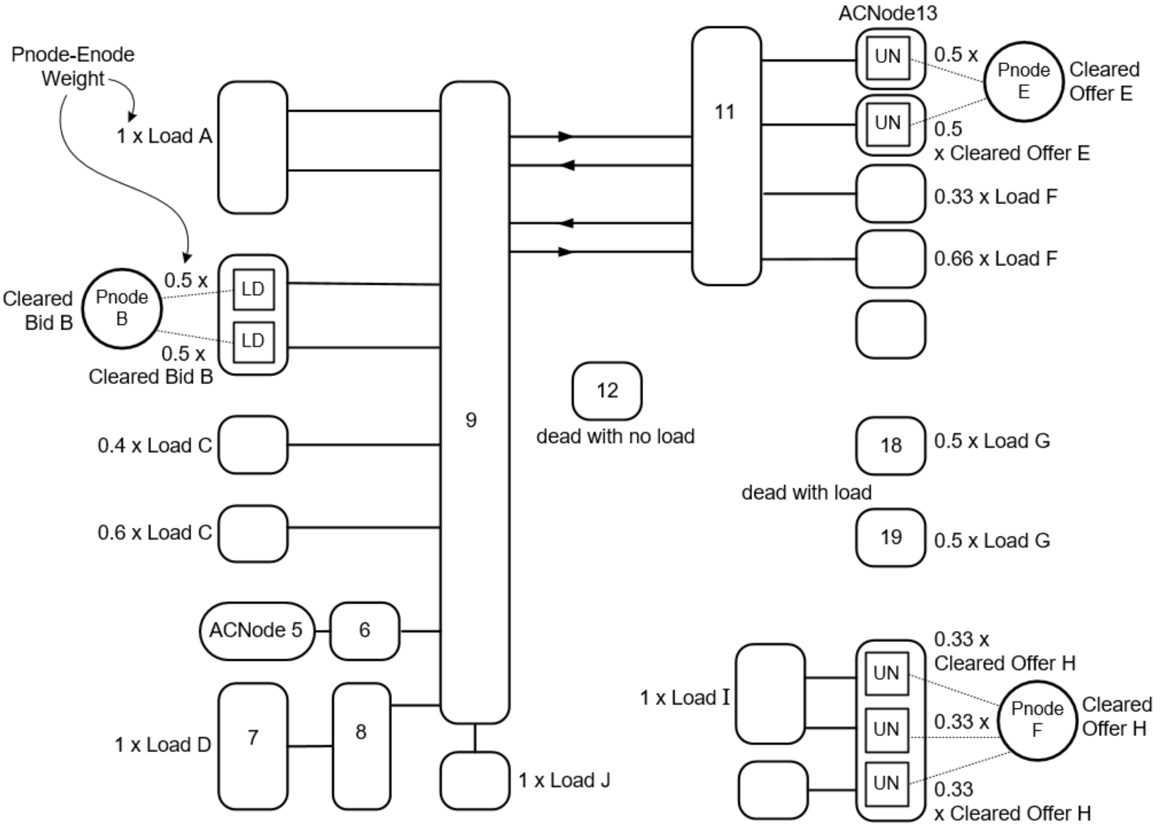


Figure 6:Example of LP Model data prepared by connectivity pre-processing

4.3 Dead Node Pre-Processing

4.3.1 *Zero cleared quantities*

For each dead Pnode the cleared bid and cleared offer quantities are fixed at zero thus preventing a non-zero cleared bid or offer quantity at the dead Pnode.

4.3.2 *Remove ACLines*

If either end of an ACLine connects to a dead ACNode then the ACLine is excluded from the LP Model. Hence a dead ACNode will not have electrical connectivity to any other ACNode.

4.4 HVDC Pre-Processing

4.4.1 *Both poles operating*

This is considered the normal situation.

4.4.2 *One or more poles not operating*

4.4.2.1. $HVDCLinkCapacity_l = 0$

$\forall l \in HVDCLinks_{POLEOUT}$ where $POLESOUT$ is the set of poles not operating

4.4.2.2. $HVDCLinkFixedLosses_l = 0$

$\forall l \in HVDCLinks_{POLEOUT}$

4.5 Reserve Sharing Pre-Processing

4.5.1 *Parameters*

Item	Description
$SharedNFRMax_{i,c}$	The maximum net free reserve provided by load damping in the non-risk island *i* that could be shared across the HVDC for reserve class *c*. A non-zero value is only applicable for reserve class $c \in \{Fast\}$.
$ResShareMaxLessMR_{i,rd}$	Maximum reserve sharing at island *i* in reserve direction *rd* due to the HVDC modulation limit less the modulation risk.
$HVDCMaxLessMR_i$	HVDC transfer capability from island *i* less the modulation risk.
$MonopoleMinPlusMR$	HVDC minimum monopole transfer increased by the modulation risk.

4.5.2 *Calculations*

4.5.2.1. $$SharedNFRMax_{i,c} = \min\left(SharedNFRMaxLimit_{i,c}, SharedNFRFactor \times \left(\sum_{n(i)} LoadForecast_n + \sum_{p \in NOMINATEDBIDS_{n(i)}} \sum_{j=1}^{DemandBidBlocks_p} DemandBidMW_{p,j} - SharedNFRLoadOffset_i\right)\right)$$

$$\forall i \in ISLANDS \quad \forall c \in \{Fast\}$$

4.5.2.2. $$SharedNFRMax_{i,c} = 0$$

$$\forall i \in ISLANDS \quad \forall c \in \{Sustained\}$$

4.5.2.3. $$RiskOffsetParameter_{i,c,rc} = RiskOffsetParameter_{i,c,rc} - SharedNFRMax_{j,c}$$

$$\forall i, j \in ISLANDS \text{ } and \text{ } i \neq j, \qquad \forall c \in \{Fast\}$$

$$\forall rc \in \left\{\begin{matrix} ACCERISKS_i, ACECERISKS_i, ManualCE_i, ManualECE_i \\ ACCERISKGROUPS_i, ACECERISKGROUPS_i \end{matrix}\right\}$$

4.5.2.4. $$HVDCMaxLessMR_i = \max\left(0, HVDCMax_i - \max\left(ModulationRisk_{i,DCCE_i}, ModulationRisk_{i,DCECE_i}\right)\right)$$

$$\forall i \in ISLANDS$$

4.5.2.5. $$ResShareMaxLessMR_{i,rd} = \max\left(0, ResShareControlBand_{rd} - \max\left(ModulationRisk_{i,DCCE_i}, ModulationRisk_{i,DCECE_i}\right)\right)$$

$$\forall i \in ISLANDS, \qquad \forall rd \in RESERVEDIRECTIONS$$

4.5.2.6. $$MonopoleMinPlusMR = MonopoleMin + max_i\left(ModulationRisk_{i,DCCE_i}, ModulationRisk_{i,DCECE_i}\right)$$

4.6 Secondary Risk Pre-Processing

To cover the possibility of reserve inadvertently offered on a secondary risk the reserve offers are set to zero for any generator that has all of its capacity modelled as a secondary risk.

4.7 Intermittent Generation Pre-Processing

4.7.1 *Parameter*

Item	Description
$PotentialMW_g$	The maximum output capability for generators in the PRICERESPONSIVEIG subset

4.7.2 *Calculation*

4.7.2.1. For generators in the $PRICERESPONSIVEIG$ subset, if the $PotentialMW_g$ value is less than $ReserveGenerationMaximum_g$ then pre-processing sets the $ReserveGenerationMaximum_g$ parameter to the $PotentialMW_g$ value, otherwise if the $PotentialMW_g$ value is greater than or equal to the $ReserveGenerationMaximum_g$ then the $ReserveGenerationMaximum_g$ value is unchanged.

4.8 Generation Ramp Pre-Processing

A generator has limits on its ability to move from one level of generation to another. SPD uses a MW-based ramping model, assuming that generators ramp instantaneously. In the case of jointly owned units the ramp rate of the primary market node unit is applied to the total generation output of the primary and secondary node units. In the case of price responsive Intermittent Generation (IG) such as wind generation, in the interests of system security a cap is applied to the ramp rate in order to limit the maximum increase within a 5-minute interval.

4.8.1 *Parameters*

Item	Description
$SolutionIntervalLength$	The length, in minutes, of the interval covered by the LP Model
$CoefficientForRampRate$	A coefficient that is used in the ramp rate constraints to convert the time frame of the ramp rate constrained quantity from $SolutionIntervalLength$ to the 60-minute time fame associated with the offered ramp rate. Calculated by Pre-Processing.
$IGIncreaseLimitForRTD$	For price responsive Intermittent Generation (IG) the 5-minute ramp-up is capped using the IGIncreaseLimitForRTD parameter which represents the maximum MW increase over 5-minutes

4.8.2 *Calculations*

4.8.2.1. $CoefficientForRampRate = \frac{60}{SolutionIntervalLength}$

4.8.2.2. For generators in the *PRICERESPONSIVEIG* subset, when *SolutionIntervalLength* is five minutes then *RampRateUp*$_g$ is capped:

$$RampRateUp_g = MIN\left(RampRateUp_g, IGIncreaseLimitForRTD \times \frac{60}{SolutionIntervalLength}\right)$$

$$\textbf{where}: SolutionIntervalLength = 5$$

$$\forall g \in PRICERESPONSIVEIG$$

4.9 Generation Tie-Break Pre-Processing

4.9.1 *Calculation*

4.9.1.1. The parameter $CappedOfferBlockMW_{g,j}$ is calculated as $GenerationOfferMW_{g,j}$ capped by $ReserveGenerationMaximum_g$ (accounting the for the $PotentialMW_g$ limit applied by 4.7.2.1), and also capped by the upper ramp rate limit, i.e., $Generation_g^{Start} + \frac{RampRateUp_g}{CoefficientForRampRate}$

4.10 Battery Pre-Processing

4.10.1 *Calculation*

The parameter $BatteryPair_{pn,pn'}$ is assigned a value of 1 when all of the following conditions are satisfied:

1. Pnode pn has either:
   a. a Demand Bid db; or
   b. a Reserve Offer r of Reserve Type IL.
2. Pnode pn' has a Generation Offer g.
3. The relevant Bids and Offers at Pnodes pn and pn' are submitted by the same Trader.
4. The associated Pnode names of pn and pn' reference the same truncated Pnode name.
5. The associated Enode names of pn and pn' reference the same battery name. See Note 1 below.

Otherwise, $BatteryPair_{pn,pn'} = 0$

For example, the load Pnode named HLY2201 HLY91 with associated Enode Key3 of 33KV_BESS11_LD, and the generation Pnode named HLY2201 HLY11 with associated Enode Key3 of BESS11, both reference the same truncated Pnode name, HLY2201 HLY, and the same battery name, BESS11. If the load Pnode has either a Demand Bid or an IL Reserve Offer, and the generation Pnode has a Generation Offer, and these are submitted by the same Trader, then these two Pnodes are defined as a battery pair.

Note 1:

To account for historical modelling where battery Enode names do not exactly match, the facility is provided for condition 5 to be excluded for specified Pnodes. For example, the load Pnode named BRB0331 RUK99 which has an Enode Key3 of 33KV_BESS_LOAD and the generation Pnode named BRB0331 RUK0 which has Enode Key3 of BESS1 will be explicitly excluded from the condition 5 requirement.

Note 2:

Each Pnode is expected to participate in at most one battery pair. As a safeguard, if application of the above conditions results in any Pnode participating in more than one pair, then $BatteryPair_{pn,pn'} = 0$ for those pairs.

4.11 Line Loss Pre-Processing

4.11.1 *Fixed and Variable Losses*

For each AC line and HVDC link, the variable loss part of the loss curve is approximated by a piecewise linear curve with a pre-determined number of segments. The approximation does not include the fixed losses, which are handled separately.

4.11.2 *Loss Approximation*

For an AC line or HVDC link the variable losses are approximated by the calculation $R \times F^2$ where R is the resistance per unit for an AC line, or the HVDC link resistance scaled to allow for using MW instead of current (assuming a constant voltage) and F is the MW flow in the line.

For the HVDC link the loss approximation calculation uses an amalgamated representation based on the in-service HVDC components.

4.11.3 *Parabola Approximation*

The variable loss approximation calculation represents a parabola. In order to be included in the linear model the parabola is approximated as a sequence of “breakpoints”, consisting of MW-loss pairs. These breakpoints are used to create loss-segments for the AC lines and lambda-loss blocks for the HVDC links.

4.12 Required Load Pre-Processing

4.12.1 *Schedule Type*

The different schedule types are modelled as follows:

Schedule Type	**Solution Interval Length (minutes)**	**Source of *PnodeRequiredLoad***
Price-Responsive	30	Load forecast
Non-Response	30	Load forecast
Real Time Dispatch (RTD and RTDP)	5	Calculated by SPD pre-processing as described below

4.12.2 *Dispatchable Pnodes*

If the Pnode associated with a Dispatchable Demand Bid is not a dead Pnode then $PnodeRequiredLoad_{pn}$ is set to zero. The Pnode load will be determined by clearing the Pnode's Dispatchable Demand Bid when the LP Model is solved.

If the Pnode associated with a Dispatchable Demand Bid is a dead Pnode then $PnodeRequiredLoad_{pn}$ is not set to zero. If the load is not zero then post-processing will set the $DeadWithLoad_{pn}$ parameter to True which indicates a discrepancy between load and connectivity.

4.12.3 *Assigning Required Load*

Non-negative *PnodeRequiredLoad* is not assigned to ACNodes, instead it is converted to price-sensitive Energy Scarcity Blocks, as described in Energy Scarcity Pre-Processing section.

Any *PnodeRequiredLoad* that is negative is assigned to ACNodes as follows:

4.12.3.1.

$$RequiredLoad_n = \sum_{en,pn\ \textbf{where}\ (EnodeACNode_{en,n}=1)\ \text{and}\ (PnodeRequiredLoad_{pn}<0)} PnodeEnodeWeight_{pn,en} \times PnodeRequiredLoad_{pn} \qquad \forall n \in ACNodes$$

4.13 Required Load for Real Time Dispatch

4.13.1 *Overview*

PnodeRequiredLoad for the Real Time Dispatch schedule types (RTD and RTDP) is calculated by applying a Pre-Solve Deviation (PSD) to the real-time generation requirement. This produces a generation forecast, which is converted to a load forecast by subtracting the network losses parameter, *LoadCalcLosses* which is initially populated by a loss estimate input. After the model is solved, the initial loss estimate is improved by repopulating *LoadCalcLosses* with the scheduled losses and there is a loop to recalculate *PnodeRequiredLoad* and then re-solve the model. An overview of the main components of the RTD load calculation process is shown in Figure 7. Note that the parameters *DidShortfallTransfer* and *ShortfallDisabledScaling* which are populated by the Energy Shortfall Check covered in 7.2 can have an impact on the second solve loop.

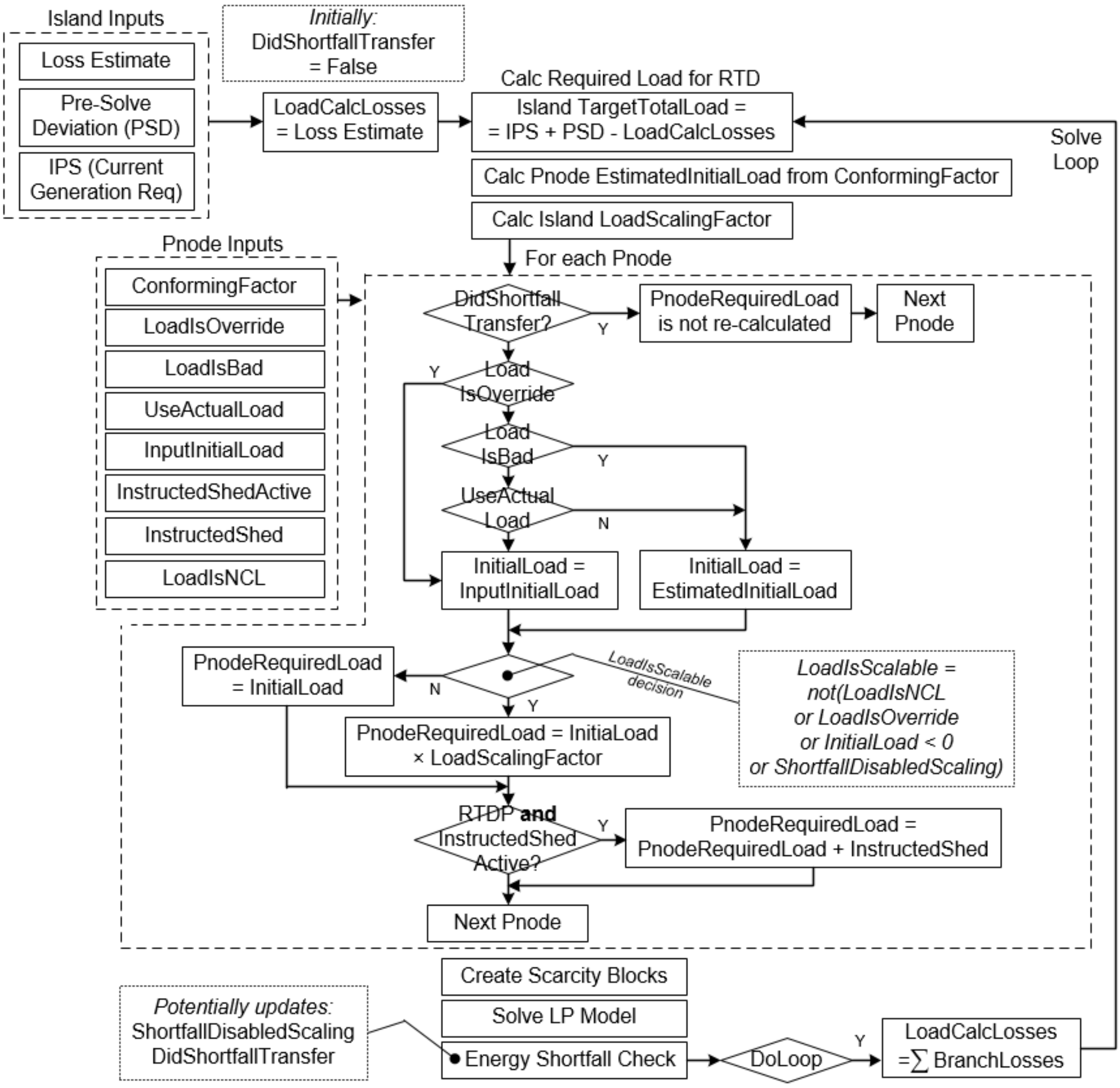


Figure 7 Overview of RTD load calculation

4.13.2 *Instructed Load Shedding*

When instructed load shedding is active the RTDP schedule is run in parallel with the RTD schedule. The RTDP schedule will produce prices that reflect the impact of the instructed load shedding quantity.

4.13.3 *Input Parameters*

The following parameters apply to RTD and RTDP schedule types:

Item	Description
$InputInitialLoad_{pn}$	If the *LoadIsOverride* flag is set True then this value represents a load MW override applied by the operator, otherwise this value represents actual load MW
$LoadIsOverride_{pn}$	Binary value. When set True indicates that the *InputInitialLoad* value represents a load MW override. When set False indicates that the *InputInitialLoad* value represents actual load MW

$UseActualLoad$	Binary value. When set False then any *InputInitialLoad* that represents actual load MW will be replaced by an estimated initial load
$LoadIsBad_{pn}$	Binary value. When set True indicates that if the associated *InputInitialLoad* represents an actual load MW value then that value is of bad quality and will be replaced by an estimated initial load
$LoadIsNCL_{pn}$	Binary value. When set True indicates that the associated Pnode is a non-conforming Pnode, i.e., Pnode load is not correlated to Island-level load. When set False indicates that the associated Pnode is a conforming Pnode, i.e., Pnode load can be estimated based on Island-level load
$ConformingFactor_{pn}$	For a conforming Pnode the *ConformingFactor* MW value is used to calculate an estimated initial load
$NonConformingLoad_{pn}$	For a non-conforming pnode the *NonConformingLoad* MW value is used to calculate an estimated initial load
$InputIPS_{i}$	The Island Power System (IPS) MW value represents the total generation required to meet the Island load, this value will be equal to the Island-level load requirement + Island-level losses
$InputIPS_{i}$	The input Island Power System (IPS) MW value is a snapshot of the monitored generation that is meeting the Island load requirement
$IslandPSD_{i}$	Pre-Solve Deviation (PSD) is an offset applied to the *InputIPS* in order to forecast the Island-level generation requirement at the end of the solution interval
$InitialLosses_{i}$	Initial estimate of Island-level losses in MW
$InstructedShedActive_{pn}$	Binary value. When set True indicates that there is an active load shedding instruction associated with this Pnode
$DispatchedLiteGen_{pn}$	Dispatched generation MW value corresponding to the unmonitored (DispatchLite) generation associated with pnode $pn \in PNODES$
$DispatchedLoad_{pn}$	Dispatched load MW value corresponding to the dispatchable load associated with pnode $pn \in PNODES$

The following parameter applies to the RTDP Schedule Type:

Item	**Description**
$InstructedShed_{pn}$	The MW quantity of the active load shedding instruction associated with this Pnode

4.13.4 *Calculated Parameters*

Item	Description
$InitialLoad_{pn}$	Value that represents the Pnode load MW at the start of the solution interval. Depending on the inputs this value will be either actual load, an operator applied override or an estimated initial load
$LoadIsScalable_{pn}$	Binary value. If True then the Pnode *InitialLoad* will be scaled in order to calculate *PnodeRequiredLoad,* if False then Pnode *InitialLoad* will be directly assigned to *PnodeRequiredLoad*
$LoadScalingFactor_{i}$	Island-level scaling factor applied to *InitialLoad* in order to calculate *PnodeRequiredLoad*
$TargetTotalLoad_{i}$	Island-level MW load forecast
$LoadCalcLosses_{i}$	Island-level MW losses used to calculate the Island-level load forecast from the *InputIPS* and the *IslandPSD*. For the first solve loop *LoadCalcLosses* will be *InitialLosses,* for the second solve loop *LoadCalcLosses* will be *SystemLosses* for the Island as calculated in section 8
$EstimatedInitialLoad_{pn}$	Calculated estimate of initial MW load, available to be used as an alternative to *InputInitialLoad*
$EstScalingFactor_{i}$	Scaling applied to *ConformingFactor* load MW in order to calculate *EstimatedInitialLoad*
$EstLoadIsScalable_{pn}$	Binary value. If True then *ConformingFactor* load MW will be scaled in order to calculate *EstimatedInitialLoad.* If False then *EstNonScalableLoad* will be assigned directly to *EstimatedInitialLoad*
$EstNonScalableLoad_{pn}$	For a non-conforming Pnode this will be the *NonConformingLoad* MW input, for a conforming Pnode this will be the *ConformingFactor* MW input if that value is negative, otherwise it will be zero
$EstScalableLoad_{pn}$	For a non-conforming Pnode this value will be zero. For a conforming Pnode this value will be the *ConformingFactor* if it is non-negative, otherwise this value will be zero
$ShortfallDisabledScaling_{pn}$	Binary value with an initial value of False. May be set to True by the Energy Shortfall Check, in which case the calculation does not scale $InitialLoad_{pn}$
$DidShortfallTransfer_{pn}$	Binary value with an initial value of False. May be set to True by the Energy Shortfall Check, in which case $PnodeRequiredLoad_{pn}$ retains its calculated value from the previous loop

4.13.5 *Derived Sets*

Item	Description
$ISLANDPNODES_i$	Pnodes in Island i
$ISLANDNONSCALABLEPNODES_i$	Pnodes in Island i with *LoadIsScalable* flag set to False, indexed by npn_i
$ISLANDSCALABLEPNODES_i$	Pnodes in Island i with *LoadIsScalable* flag set to True, indexed by spn_i

4.13.6 *Calculations*

Note that for binary values: 1= True, 0 = False.

4.13.6.1.

$$PnodeRequiredLoad_{pn} = \begin{cases} InitiaLoad_{pn} \times LoadScalingFactor_i, & \textbf{if } LoadIsScalable_{pn} = 1 \\ InitialLoad_{pn}, & \textbf{otherwise} \end{cases}$$

$$+ \begin{cases} InstructedShed_{pn}, & \textbf{if } ScheduleType = \mathrm{RTDP} \textbf{ and } InstructedShedActive_{pn} = 1 \\ 0, & \textbf{otherwise} \end{cases}$$

$$\forall i \in ISLANDS, \forall pn \in ISLANDPNODES_i$$

$$\textbf{where } DidShortfallTransfer_{pn} = 0$$

The components of 4.13.6.1 are determined as follows:

4.13.6.2.

$$InitialLoad_{pn} = \begin{cases} PnodeRequiredLoad_{pn}, \textbf{if } DidShortfallTransfer_{pn} = 1 \\ EstimatedInitialLoad_{pn}, \ \textbf{if} LoadIsOverride_{pn} = 0 \\ \textbf{and } (UseActualLoad = 0 \textbf{ or } LoadIsBad_{pn} = 1) \\ InputIntialLoad_{pn}, \ \textbf{otherwise} \end{cases}$$

$$\forall pn \in PNODES$$

4.13.6.3.

$$LoadScalingFactor_i = \frac{TargetTotalLoad_i - \sum InitialLoad_{npn_i}}{\sum InitialLoad_{spn_i}}$$

$$\forall i \in ISLANDS \textbf{ where } npn_i \in ISLANDNONSCALABLEPNODES_i$$

$$spn_i \in ISLANDSCALABLEPNODES_i$$

4.13.6.4.

$$LoadIsScalable_{pn} = \begin{cases} 0, \ \textbf{if } LoadIsNCL_{pn} = 1 \textbf{ or } LoadIsOverride_{pn} = 1 \textbf{ or } \ InitialLoad_{pn} < 0 \\ \textbf{or } ShortfallDisabledScaling_{pn} = 1 \ \textbf{ or } DidShortfallTransfer_{pn} = 1 \\ 1, \ \textbf{otherwise} \end{cases}$$

$$\forall pn \in PNODES$$

4.13.6.5.

$$\begin{aligned} TargetTotalLoad_i = {} & InputIPS_i + IslandPSD_i - LoadCalcLosses_i \\ & + \textstyle\sum_{pn \in ISLANDPNODES_i} Dispatched\text{Lite}Gen_{pn} \\ & - \textstyle\sum_{pn \in ISLANDPNODES_i} DispatchedLoad_{pn} \end{aligned}$$

$$\forall i \in ISLANDS$$

4.13.6.6.

$$EstimatedInitialLoad_{pn_i} = \begin{cases} ConformingFactor_{pn_i} \times EstScalingFactor_i, \textbf{if } EstLoadIsScalable_{pn_i} = 1 \\ EstNonScalableLoad_{pn_i}, \textbf{otherwise} \end{cases}$$

$$\forall i \in ISLANDS, \forall pn_i \in ISLANDPNODES_i$$

4.13.6.7.

$$EstLoadIsScalable_{pn} = \begin{cases} 0,\ \textbf{if } LoadIsNCL_{pn} = 1 \textbf{ or } ConformingFactor_{pn} < 0 \\ 1,\ \textbf{otherwise} \end{cases}$$

$$\forall pn \in PNODES$$

4.13.6.8.

$$EstNonScalableLoad_{pn} = \begin{cases} 0, \textbf{if } EstLoadIsScalable_{pn} = 1 \\ NonConformingLoad_{pn}, \textbf{if } LoadIsNCL_{pn} = 1 \\ ConformingFactor_{pn}, \textbf{otherwise} \end{cases}$$

$$\forall pn \in PNODES$$

4.13.6.9.

$$EstScalingFactor_i = \frac{InputIPS_i - LoadCalcLosses_i - \sum EstNonScalableLoad_{pn_i}}{\sum EstScalableLoad_{pn_i}}$$

$$\forall i \in ISLANDS \textbf{ where } pn_i \in ISLANDPNODES_i$$

4.13.6.10.

$$EstScalableLoad_{pn} = \begin{cases} ConformingFactor_{pn}, \textbf{if } EstLoadIsScalable_{pn} = 1 \\ 0, \textbf{otherwise} \end{cases}$$

$$\forall pn \in PNODE$$

4.14 Energy Scarcity Pre-Processing

4.14.1 *Input Parameters*

Item	Description
$PnodeEnergyScarcityLimit_{pn,j}$	Optional parameter that if present will set an $EnergyScarcityLimit_{es,j}$ for Energy Scarcity Load block *j* associated with a Pnode *pn*
$PnodeEnergyScarcityFactor_{pn,j}$	Optional factor that if present will be applied to $PnodeRequiredLoad_{pn}$ in order to calculate $EnergyScarcityLimit_{es,j}$ for Energy Scarcity Load block *j* associated with a Pnode *pn* that does not have any $PnodeEnergyScarcityLimit_{pn,j}$ parameters supplied

$NationalEnergyScarcityFactor_j$	National factor that will be applied to $PnodeRequiredLoad_{pn}$ in order to calculate $EnergyScarcityLimit_{es,j}$ for Energy Scarcity Load block *j* associated with Pnode *pn* that does not have any $PnodeEnergyScarcityLimit_{pn,j}$ or $PnodeEnergyScarcityFactor_{pn,j}$ parameters supplied

4.14.2 *Calculations*

4.14.2.1. $EnergyScarcityLimit_{es,j} =$

$$\begin{cases} PnodeEnergyScarcityLimit_{pn,j}, \textbf{ if exists } PnodeEnergyScarcityLimit_{pn,j} \\ PnodeRequiredLoad_{pn} \times PnodeEnergyScarcityFactor_{pn,j}, \textbf{ if exists } PnodeEnergyScarcityFactor_{pn,j} \\ PnodeRequiredLoad_{pn} \times NationalEnergyScarcityFactor_j, \quad \textbf{otherwise} \end{cases}$$

$$\forall es \in ENERGYSCARCITYLOAD_{pn} \textbf{ where } PnodeRequiredLoad_{pn} > 0$$

5 Objective Function

The *NetBenefit* is maximised.

5.1.1.1. $NetBenefit =$

$$\sum_{db \in BIDS} \sum_{j=1}^{DemandBidBlocks_{db}} DemandBlock_{db,j} \times DemandBidPrice_{db,j}$$

$$-\sum_{g \in OFFERS} \sum_{j=1}^{GenerationOfferBlocks_g} GenerationBlock_{g,j} \times GenerationOfferPrice_{g,j}$$

$$-\sum_{r \in RESERVEOFFERS} \sum_{j=1}^{ReserveOfferBlocks_r} ReserveBlock_{r,j} \times ReserveOfferPrice_{r,j}$$

$$-ExcessResSharePenalty$$

$$+\sum_{es \in ENERGYSCARCITYLOADS} \sum_{j=1}^{EnergyScarcityBlockCount_{es}} EnergyScarcityCleared_{es,j} \times EnergyScarcityPrice_{es,j}$$

$$-\sum_{rc \in RISKCLASSES_{i,CE}} \sum_{c \in RESERVECLASSES} \sum_{j=1}^{ReserveScarcityBlockCount_{i,c}} ReserveScarcityCleared_{c,rc,j} \times ReserveScarcityPrice_{i,c,j}$$

$$-\sum_{i \in ISLANDS} \sum_{c \in RESERVECLASSES} ECEReserveDeficit_{i,c} \times ECEReserveDeficitPrice_c$$

$$-TieBreakSlackPrice \times \sum_{g \in OFFERS} \sum_{j=1}^{GenerationOfferBlocks_g} TieBreakSlack1_{g,j} + TieBreakSlack2_{g,j}$$

$$\begin{aligned} &- \textbf{\textit{if}}\ ScheduleType = \ Real\ Time\ Dispatch\ \textbf{\textit{then}} \\ &\qquad \sum_{g \in OFFERS} PenaltyForMWChangeOf5MinDispatch \\ &\qquad \times (GenerationChange1_g + GenerationChange2_g) \end{aligned}$$

6 Constraints

6.1 Generation and Load

6.1.1.1. $GenerationBlock_{g,j} \leq GeneratorOfferMW_{g,j}$

$$j = 1, \dots GenerationOfferBlocks_g \quad \forall\, g \in OFFERS$$

6.1.1.2. $GenerationChange1_g - GenerationChange2_g$

$$= \left(Generation_g - Generation_g^{Start}\right)$$

$\forall\, g \in OFFERS$ **where** $ScheduleType =$ Real Time Dispatch

6.1.1.3. $Generation_g = \sum_{j=1}^{GenerationOfferBlock_g} GenerationBlock_{g,j}$

$$\forall\, g \in OFFERS$$

6.1.1.4. $Generation_g \le PotentialMW_g$

$$\forall\, g \in WINDGENERATORS$$

6.1.1.5. $0 \le DemandBlock_{db,j} \le DemandBidMW_{db,j}$

if $DemandBidMW_{db,j} \ge 0, \quad j = 1, \dots DemandBidBlocks_{db}\ \forall db \in BIDS$

6.1.1.6. $0 \ge DemandBlock_{db,j} \ge DemandBidMW_{db,j}$

if $DemandBidMW_{db,j} \le 0, \quad j = 1, \dots DemandBidBlocks_{db}\ \forall db \in BIDS$

6.1.1.7. $DemandBlock_{db,j} = BlockMustFullyClear_{db,j} \times DemandBidMW_{db,j}$

$$j = 1, \dots DemandBidBlocks_{db}\ \forall db \in DISCRETEDEMANDBIDS$$

6.1.1.8. $Demand_{db} = \sum_{j=1}^{DemandBidBlocks_{db}} DemandBlock_{db,j} \quad \forall\, db \in BIDS$

6.1.1.9. $EnergyScarcityCleared_{es,j} \le EnergyScarcityLimit_{es,j}$

$$j = 1, \dots EnergyScarcityBlockCount_{es} \quad \forall\, es \in ENERGYSCARCITYLOADS$$

6.1.1.10. $EnergyScarcityLoad_{es} = \sum_{j=1}^{EnergyScarcityBlockCount_{es}} EnergyScarcityCleared_{es,j}$

$$\forall\, es \in ENERGYSCARCITYLOADS$$

6.1.1.11. $\frac{GenerationBlock_{g,j}}{CappedOfferBlockMW_{g,j}} - \frac{GenerationBlock_{g',j'}}{CappedOfferBlockMW_{g',j'}}$

$$= TieBreakSlack1_{g,j,g',j'} - TieBreakSlack2_{g,j,g',j'}$$

$$\forall\, n \in ACNodes, \qquad \forall g, g' \in TIEBREAKOFFERS_n$$

where $g \ne g'$ **and** $GenerationOfferPrice_{g,j} = GenerationOfferPrice_{g',j'}$
and $CappedOfferBlockMW_{g,j} > 0$ **and** $CappedOfferBlockMW_{g',j'} > 0$

6.1.1.12. $Demand_{pn} + Reserve_{IL,pn} <= BatteryChargingMode_{pn,pn'} \times M$

$\forall pn, pn' \in PNODES$ **where** $BatteryPair_{pn,pn'}$

6.1.1.13. $Generation_{pn'} <= (1 - BatteryChargingMode_{pn,pn'}) \times M$

$\forall pn, pn' \in PNODES$ **where** $BatteryPair_{pn,pn'}$

6.2 Ramping

6.2.1.1. $\left(Generation_g - Generation_g^{Start}\right) \leq \frac{RampRateUp_g}{CoefficientForRampRate}$

$\forall g \in OFFERS$

6.2.1.2. $\left(Generation_g^{Start} - Generation_g\right) \leq \frac{RampRateDown_g}{CoefficientForRampRate}$

$\forall g \in OFFERS$

6.3 HVDC Transmission

6.3.1.1. $HVDCLinkFlow_l \leq HVDCLinkCapacity_l \quad \forall l \in HVDCLINKS$

6.3.1.2. $HVDCLinkLosses_l = \sum_{bp=1}^{HVDCBreakpoints_l} HVDCBreakpointMWLosses_{l,bp} \times Lambda_{l,bp}$

6.3.1.3. $HVDCLinkFlow_l = \sum_{bp=1}^{HVDCBreakpoints_l} HVDCBreakpointMWFlow_{l,bp} \times Lambda_{l,bp}$

6.3.1.4. $\sum_{bp=1}^{HVDCBreakpoints_l} Lambda_{l,bp} = 1$

6.4 AC Transmission

6.4.1.1. $ACNodeNetInjection_n = \sum_{q \in S_{AC}(n)} ACLineFlow_q^{Directed} - \sum_{q \in R_{AC}(n)} ACLineFlow_q^{Directed}$

$\forall n \in ACNODES$

6.4.1.2. $ACNodeNetInjection_n = \sum_{g \in OFFERS_n} Generation_g$

$- \sum_{db \in BIDS_n} Demand_{db} - RequiredLoad_n$

$- \sum_{es \in ENERGYSCARCITYLOADS_n} EnergyScarcityLoadCleared_{es}$

$- \sum_{l \in S_{HVDC}(n)} HVDCLinkFlow_l$

$+ \sum_{l \in R_{HVDC}(n)} (HVDCLinkFlow_l - HVDCLinkLosses_l)$

$- \sum_{l \in HVDCLINKS_n} \frac{1}{2} \times HVDCLinkFixedLosses_l$

$- \sum_{q \in R_{AC}(n)} ACLineLosses_q^{Directed}$

$- \sum_{k \in ACLINES_n} \frac{1}{2} \times ACLineFixedLosses_k$

$$\forall n \in ACNODES$$

6.4.1.3. $ACLineFlow_{q}^{Directed} \leq ACLineCapacity_{k(q)}$

$$\forall\, q \in DIRECTEDACLINES$$

6.4.1.4. $ACLineFlow_{k} = ACLineFlow_{F(k)}^{Directed} - ACLineFlow_{B(k)}^{Directed}$

$$\forall\, k \in ACLINES$$

6.4.1.5. $ACLineFlow_{k} = ACLineAdmittance_{k} \times \left(ACNodeAngle_{b(k)} - ACNodeAngle_{e(k)}\right)$

$$\forall\, k \in ACLINES$$

6.4.1.6. $ACLineFlowBlock_{q.j}^{Directed} \leq ACLineLossBlockMW_{k(q)}$

$$j = 1, \dots, ACLineLossBlockCount_{k} \;\; \forall q \in DIRECTEDACLINES$$

6.4.1.7. $ACLineFlow_{q}^{Directed} = \sum_{j=1}^{ACLineLossBlockCount_{k}} ACLineFlowBlock_{q,j}^{Directed}$

$$\forall q \in DIRECTEDACLINES$$

6.4.1.8. $ACLineLossesBlock_{q,j}^{Directed} = ACLineFlowBlock_{q,j}^{Directed} \times ACLineLossBlockFactor_{k(q),j}$

$$j = 1, \dots, ACLineLossBlockCount_{k} \;\; \forall q \in DIRECTEDACLINES$$

6.4.1.9. $ACLineLosses_{q}^{Directed} = \sum_{j=1}^{ACLineLossBlockCount_{k}} ACLineLossesBlock_{q,j}^{Directed}$

$$\forall q \in DIRECTEDACLINES$$

6.4.1.10. $ACNodeAngle_{n} = 0 \;\; \forall n \in REFERENCENODES$

6.5 Risk and Reserve

6.5.1 *Risk*

6.5.1.1. $RiskOffset_{i,c,rc} - Rampup_{i} = NetFreeReserve_{i,c,rc}$

$$\forall c \in RESERVECLASSES \;\; \forall i \in ISLANDS \;\; rc = DCCE_{i}$$

6.5.1.2. $Rampup_{i} = RampupMax_{i} \quad \forall i \in ISLANDS$

6.5.1.3. $RiskOffset_{i,c,rc} = NetFreeReserve_{i,c,rc}$

$$\forall c \in RESERVECLASSES \;\; \forall i \in ISLANDS \;\; rc = DCECE_{i}$$

6.5.1.4. $HVDCRec_{i} = \sum_{n(i)} \begin{pmatrix} -\sum_{l \in S_{HVDC}(n)} HVDCLinkFlow_{l} \\ +\sum_{l \in R_{HVDC}(n)} [\![(HVDCLinkFlow]\!]_{l} - HVDCLinkLoss_{l}) \end{pmatrix}$

$$\forall i \in ISLANDS$$

6.5.1.5. $IslandRisk_{i,c,rc} = IslandRiskAdjustmentFactor_{i,c,rc} \times \left(HVDCRec_i - RiskOffset_{i,c,rc} + ModulationRisk_{i,rc}\right)$

$$\forall c \in RESERVECLASSES \;\; \forall i \in ISLANDS \;\; rc \in \{DCCE_i, DCECE_i\}$$

6.5.1.6. $IslandRisk_{i,c,rc} = IslandRiskAdjustmentFactor_{i,c,rc} \times \left(Generation_g - -ACSecondaryRiskMW_{g,rc} - RiskOffsetParameter_{i,c,rc} + FKBand_g + \sum_{r \in RESERVEOFFERS_{g,c}} Reserve_r\right)$

$$\forall g \in ISLANDRISKGENERATORS_i \;\; \forall c \in RESERVECLASS$$
$$\forall i \in ISLANDS \;\; rc = \{ACCERISKS_i, ACECERISKS_i\}$$

6.5.1.7. $IslandRisk_{i,c,rc} = IslandRiskAdjustmentFactor_{i,c,rc} \times \left(IslandMinimumRisk_{i,rc} - RiskOffsetParameter_{i,c,rc}\right)$

$$\forall c \in RESERVECLASS \;\; \forall i \in ISLANDS \;\; rc \in \{ManualCE_i, ManualECE_i\}$$

6.5.1.8. $HVDCSent_i \leq M \times (1 - HVDCSentMustZero_i)$

$\forall i \in ISLANDS$

6.5.1.9. $IslandRisk_{i,c,rc} = IslandRiskAdjustmentFactor_{i,c,rc} \times \left(Generation_g - RiskOffsetParameter_{i,c,rc} + HVDCRec_i + ModulationRisk_{i,rc} - HVDCSecondaryRiskSubtractor_i + FKBand_g + \sum_{r \in RESERVEOFFERS_{g,c}} Reserve_r\right) - M \times HVDCSentMustZero_j$

$$\forall g \in ISLANDRISKGENERATORS_i \;\; \forall c \in RESERVECLASS$$
$$\forall i,j \in ISLANDS \; and \; i \neq j \;\; rc \in \{HVDCSECRISKSACCE_i, HVDCSECRISKSACECE_i\}$$

6.5.1.10. $IslandRisk_{i,c,rc} = IslandRiskAdjustmentFactor_{i,c,rc} \times \left(IslandMinimumRisk_{i,rc} - RiskOffsetParameter_{i,c,rc} + HVDCRec_i + ModulationRisk_{i,rc} - HVDCSecondaryRiskSubtractor_i\right) - M \times HVDCSentMustZero_j$

$$\forall c \in RESERVECLASS \;\; \forall i,j \in ISLANDS \; and \; i \neq j$$
$$rc \in \{HVDCSECRISKSMANUALCE_i, HVDCSECRISKSMANUALECE_i\}$$

6.5.1.11. $IslandRisk_{i,c,rc} = IslandRiskAdjustmentFactor_{i,c,rc} \times \left(\sum_{g \in OFFERS_{rg}} (Generation_g + FKBand_g) + \sum_{r \in RESERVEOFFERS_{rg,c}} Reserve_r - ACSecondaryRiskMW_{rg,rc} - RiskOffsetParameter_{i,c,rc}\right)$

$$\forall i \in ISLANDS \;\; \forall rg \in RISKGROUPS_i \;\; \forall c \in RESERVECLASS$$
$$rc \in \{ACCERISKGROUPS_i, ACECERISKGROUPS_i\}$$

6.5.1.12. $IslandRisk_{i,c,rc} = IslandRiskAdjustmentFactor_{i,c,rc} \times \left(\sum_{k \in ACLINES_{lr}} ACLineFlow_k \times DirectionalRiskFactor_{lr,k} + \sum_{r \in RESERVEOFFERS_{lr,c}} Reserve_r - ACSecondaryRiskMW_{lr,rc} - RiskOffsetParameter_{i,c,rc}\right)$

$$\forall i \in ISLANDS \;\; \forall lr \in LINKRISKS_i \;\; \forall c \in RESERVECLASS$$

$$rc \in \{ACCELINKRISKS_i, ACECELINKRISKS_i\}$$

6.5.2 *Reserve sharing*

6.5.2.1. $ResShareEffective_{i,c,rc} \leq \sum_{rd} ResShareReceivedEffectiveness_{i,c,rc} \times ResShareReceived_{i,c,rd}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS$$

$$\forall rc \in \begin{Bmatrix} ACCERISKS_i, ACECERISKS_i, ManualCE_i, ManualECE_i \\ ACCERISKGROUPS_i, ACECERISKGROUPS_i \end{Bmatrix}$$

6.5.2.2. $SharedIslandReserve_{i,c} \leq IslandReserve_{i,c}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS$$

6.5.2.3. $SharedNFR_{i,c} \leq SharedNFRMax_{i,c}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS$$

6.5.2.4. $ResShareSent_{i,c,rd} \leq SharedIslandReserve_{i,c} + SharedNFR_{i,c}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in RESERVEDIRECTIONS$$

6.5.2.5. $ResShareSent_{i,c,rd} \leq ResShareMaxLessMR_{i,rd}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in RESERVEDIRECTIONS$$

6.5.2.6. $HVDCSent_i + ResShareSent_{i,c,rd} \leq HVDCMaxLessMR_i$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{FORWARD\}$$

6.5.2.7. $ResShareSent_{i,c,rd} \leq M \times (1 - IsSendingHVDC_i)$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{REVERSE\}$$

6.5.2.8. $ResShareReceived_{i,c,rd} \leq ResShareMaxLessMR_{i,rd} \times IsSendingHVDC_i$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{REVERSE\}$$

6.5.2.9. $ResShareReceived_{i,c,rd} \leq M \times (1 - IsSendingHVDC_i)$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{FORWARD\}$$

6.5.2.10. $HVDCSent_i - ResShareReceived_{i,c,rd} \geq MonopoleMinPlusMR - M \times \left(1 - InZone_{i,c,rrz}\right)$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{REVERSE\}$$

$$\forall rrz \in \{REVERSEZONE\}$$

6.5.2.11. $ResShareReceived_{i,c,rd} \leq M \times \left(1 - InZone_{i,c,rrz}\right)$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{REVERSE\}$$
$$\forall rrz \in \{NOREVERSEZONE\}$$

6.5.2.12. $\sum_{i \in ISLANDS} \sum_{rrz} InZone_{i,c,rrz} = 1$

$$\forall c \in RESERVECLASS$$

6.5.2.13. $HVDCSent_i \leq M \times \sum_{rrz} InZone_{i,c,rrz}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS$$

6.5.2.14. $HVDCSent_i \leq RoundPowerZoneExit_c + M \times \left(1 - InZone_{i,c,rrz}\right)$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rrz \in \{ROUNDPOWERZONE\}$$

6.5.2.15. $IsSendingHVDC_i = \sum_{rrz} InZone_{i,c,rrz}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS$$

6.5.2.16. $InZone_{i,c,rrz} \leq (1 - RoundPowerDisabled_c)$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rrz \in \{ROUNDPOWERZONE\}$$

6.5.2.17. $InZone_{i,c,rrz} \leq RoundPowerDisabled_c$

$$\forall i \in ISLANDS, \quad \forall c \in \{Sustained\}, \quad \forall rrz \in \{NOREVERSEZONE\}$$

6.5.2.18. $ResShareSent_{i,c,rd} \leq M \times \left(2 - InZone_{i,c,rrz} - RoundPowerDisabled_c\right)$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{FORWARD\}$$
$$\forall rrz \in \{NOREVERSEZONE\}$$

6.5.2.19. $\sum_{i \in ISLANDS} IsSendingHVDC_i = 1$

6.5.2.20. $HVDCSent_i = \sum_{n(i)} \sum_{l \in S_{HVDC}(n)} HVDCLinkFlow_l$

$$\forall i \in ISLANDS$$

6.5.2.21. $ResShareSent_{i,c,rd} = HVDCSentAfterResShare_{i,c,rd} - HVDCSent_i$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{FORWARD\}$$

6.5.2.22. $ResShareReceived_{i,c,rd} = ResShareSent_{j,c,rd} - HVDCLossAfterResShare_{j,c,rd} + HVDCSentLoss_j$

$$\forall i,j \in ISLANDS \text{ and } i \neq j, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{FORWARD\}$$

6.5.2.23. $ResShareReceived_{i,c,rd} = HVDCSent_i - HVDCSentAfterResShare_{i,c,rd}$

$$\forall i,j \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{REVERSE\}$$

6.5.2.24. $ResShareReceived_{i,c,rd} = ResShareSent_{j,c,rd} - HVDCLossAfterResShare_{i,c,rd} + HVDCSentLoss_i$

$$\forall i,j \in ISLANDS \text{ and } i \neq j, \quad \forall c \in RESERVECLASS, \quad \forall rd \in \{REVERSE\}$$

6.5.2.25. $\sum_{bp} LambdaHVDCSent_{i,bp} = 1$

$$\forall i \in ISLANDS$$

6.5.2.26. $HVDCSent_i = \sum_{bp} HVDCSentFlowBreakPoint_{i,bp} \times LambdaHVDCSent_{i,bp}$

$$\forall i \in ISLANDS$$

6.5.2.27. $HVDCSentLoss_i = \sum_{bp} HVDCSentLossBreakPoint_{i,bp} \times LambdaHVDCSent_{i,bp}$

$$\forall i \in ISLANDS$$

6.5.2.28. $\sum_{rsbp} LambdaResShare_{i,c,rd,rsbp} = 1$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in RESERVEDIRECTIONS$$

6.5.2.29. $HVDCSentAfterResShare_{i,c,rd} = \sum_{rsbp} HVDCAfterResShareBreakPoint_{i,rsbp} \times LambdaResShare_{i,c,rd,rsbp}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in RESERVEDIRECTIONS$$

6.5.2.30. $HVDCLossAfterResShare_{i,c,rd} = \sum_{rsbp} HVDCAfterResShareLossBreakPoint_{i,rsbp} \times LambdaResShare_{i,c,rd,rsbp}$

$$\forall i \in ISLANDS, \quad \forall c \in RESERVECLASS, \quad \forall rd \in RESERVEDIRECTIONS$$

6.5.2.31. $ExcessResSharePenalty = \sum_{i,c,rc} ExcessResShareEffectivePenaltyPrice \times ResShareEffective_{i,c,rc} + \sum_{i,c} ExcessSharedNFRPenaltyPrice \times SharedNFR_{i,c} + \sum_{i,c} ExcessSharedIslandResPenaltyPrice \times SharedIslandReserve_{i,c}$

6.5.3 *Reserve*

6.5.3.1. $ReserveBlock_{r,j} \leq ReserverOfferProportion_{r,j} \times Generation_{g(r)}$

$$j = 1, \dots, ReserveOfferBlocks_r \quad \forall r \in RESERVEOFFERS_{PLSR}$$

6.5.3.2. $ReserveBlock_{r,j} \leq ReserveOfferMaximum_{r,j}$

$$j = 1, \dots, ReserveOfferBlocks_r \quad \forall r \in RESERVEOFFERS$$

6.5.3.3. $\sum_{j \in RESERVEOFFERBLOCKS} ReserveBlock_{r,j} \leq Demand_{db(r)}$

$$\forall db \in DISPATCHABLEBIDS \quad \forall r \in RESERVEOFFERS_{IL}$$

6.5.3.4. $Reserve_r = \sum_{j=1}^{ReserveOfferBlocks_r} ReserveBlock_{r,j}$

$$\forall r \in RESERVEOFFERS$$

6.5.3.5. $Generation_g + ReserveMaximumFactor_{g,c} \times \sum_{r \in RESERVEOFFERS_{g,c}} Reserve_r \leq ReserveGenerationMaximum_g$

$$\forall g \in OFFERS \quad \forall c \in RESERVECLASSES$$

6.5.4 *Reserve Scarcity*

6.5.4.1. $ReserveScarcityCleared_{c,\mathrm{rc},j} \leq ReserveScarcityLimit_{i,c,j}$

$$\forall \mathrm{c} \in RESERVECLASSES \quad \forall rc \in RISKCLASSES_{i,CE}$$

$$j = 1, \ldots ReserveScarcityBlockCount_{i,c}$$

6.5.4.2. $ReserveShortfall_{c,rc} = \sum_{j=1}^{ReserveScarcityBlockCount_{i,c}} ReserveScarcityCleared_{c,rc,j}$

$$\forall \mathrm{c} \in RESERVECLASSES \quad \forall rc \in RISKCLASSES_{i,CE}$$

6.5.5 *Reserve covers Risk*

6.5.5.1. $IslandReserve_{i,c} = \sum_{r \in RESERVEOFFERS_{i,c}} Reserve_r$

$$\forall c \in RESERVECLASSES \quad \forall i \in ISLANDS$$

6.5.5.2. $IslandReserve_{i,c} + ResShareEffective_{i,c,rc} + ReserveShortfall_{c,\mathrm{rc}} \geq IslandRisk_{i,c,rc}$

$$\forall \mathrm{i} \in \mathrm{ISLANDS} \quad \forall \mathrm{c} \in RESERVECLASSES \quad \forall rc \in \{ACCERISKS_i, ManualCE_i, ACCERISKGROUPS_i, ACCELINKRISKS_i\}$$

6.5.5.3. $IslandReserve_{i,c} + ReserveShortfall_{c,\mathrm{rc}} \geq IslandRisk_{i,c,rc}$

$$\forall \mathrm{i} \in \mathrm{ISLANDS} \quad \forall \mathrm{c} \in RESERVECLASSES \quad rc = DCCERISKS_i$$

6.5.5.4. $IslandReserve_{i,c} + ResShareEffective_{i,c,rc} + ECEReserveDeficit_{\mathrm{i},c} \geq IslandRisk_{i,c,rc}$

$$\forall \mathrm{i} \in \mathrm{ISLANDS} \quad \forall \mathrm{c} \in RESERVECLASSES \quad \forall rc \in \{ACECERISKS_i, ManualECE_i, ACECERISKGROUPS_i, ACCELINKRISKS_i\}$$

6.5.5.5. $IslandReserve_{i,c} + ECEReserveDeficit_{\mathrm{i},c} \geq IslandRisk_{i,c,rc}$

$$\forall \mathrm{i} \in \mathrm{ISLANDS} \quad \forall \mathrm{c} \in RESERVECLASSES \quad rc = DCECERISKS_i$$

6.6 Security

6.6.1.1. $Generation_{g(v)} \leq SecurityGenerationMaximum_v$

$$\forall v \in SECURITY_{GenerationMaximum}$$

6.6.1.2. $Generation_{g(v)} \geq SecurityGenerationMinimum_v$

$$\forall v \in SECURITY_{GenerationMinimum}$$

6.6.1.3. $ACLineFlow_{q(v)}^{Directed} \leq SecurityACLineCapacity_v$

$$\forall v \in SECURITY_{ACLineCapacity}$$

6.6.1.4. $HVDCLinkFlow_{l(v)} \leq SecurityHVDCLinkCapacity_v$

$$\forall v \in SECURITY_{HVDCLinkCapacity}$$

6.6.1.5. $\sum_{q \in SECURITYACLINESGROUP_v} ACLineFlow_q^{Directed} \times SecurityGroupACLineWeight_q \leq SecurityGroupACLinesFlow_v$

$$\forall v \in SECURITY_{GroupACLinesFlow}$$

6.6.1.6. $\sum_{n \in SECURITYACNODESGROUP_v} ACNodeNetInjection_n \times SecurityGroupACNodeWeight_n \leq SecurityGroupACNodesNetInjection_v$

$$\forall v \in SECURITY_{GroupACNodesNetInjection}$$

6.6.1.7. $\sum_{p \in SECURITYMARKETDEMNODEGROUP_v} Demand_p \times MarketNodeDemWeight_p + \sum_{g \in SECURITYMARKETGENNODEGROUP_v} Generation_g \times MarketNodeGenWeight_g + \sum_{r \in SECURITYMARKETRESNODEGROUP_v} Reserve_r \times MarketNodeResWeight_r = \leq \geq MarketNodeSecurityLimit_v$

$$\forall v \in SECURITY_{GroupMarketNodes}$$

6.7 Mixed Constraints

Note that in the following constraints the sign = ≤ ≥ is taken as meaning three constraint types in one formula, these being =, ≤ and ≥.

6.7.1.1. $MixedConstraintVariable_m \times MixedConstVarWeight1_m + \sum_{p \in MIXEDDEMNODEGROUP_m} Demand_p \times MixedConstDemWeight_{p,m} + \sum_{g \in MIXEDGENNODEGROUP_m} Generation_g \times MixedConstGenWeight_{g,m} + \sum_{r \in MIXEDRESNODEGROUP_m} Reserve_r \times MixedConstResWeight_{r,m} + \sum_{q \in MIXEDDIRACLINEGROUP_m} ACLineFlow_p^{Directed} \times MixedConstACLineWeight_{q,m} + \sum_{q \in MIXEDDIRACLINEGROUP_m} ACLineLosses_q^{Directed} \times MixedConstACLineLossWeight_{q,m} +$

$\sum_{k \in MIXEDACLINEGROUP_m} ACLineFixedLosses_k \times MixedConstACLineFixedLossWeight_{k,m} + \sum_{l \in MIXEDDCLINKGROUP_m} HVDCLinkFlow_l \times MixedConstDCLinkWeight_{l,m} + \sum_{l \in MIXEDDCLINKGROUP_m} HVDCLinkLosses_l \times MixedConstDCLinkLossWeight_{l,m} + \sum_{l \in MIXEDDCLINKGROUP_m} HVDCLinkFixedLosses_l \times MixedConstDCLinkFixedLossWeight_{l,m} = \leq \geq MixedConstraintLimit1_m$

$$\forall m \in MIXEDCONSTRAINTS_{Type1}$$

6.7.1.2. $\sum_{m \in MIXEDVARGROUP_b} MixedConstraintVariable_m \times MixedConstVarWeight2_{m,b} = \leq \geq MixedConstraintLimit2_b$

$$\forall b \in MIXEDCONSTRAINTS_{Type2}$$

6.8 Integer Constraints

The Mathematical model presented in this document has a linear objective function with integer variables used in some constraints thus resulting in a mixed integer linear programming (MILP) formulation.

Constraints 6.4.1.6 - 6.4.1.9 are used to linearise a non-linear equality constraint, a technique which is only reliably applicable for convex optimisation problems. Hence, when the effective cost of losses is negative this approximation will not produce the correct result.

When post-processing detects circulating flows on AC Lines or HVDC Poles, additional integer constraints specifically designed to remove circulating flows are activated and the problem is re-solved. The process and the constraints are described in Post Processing section 7.

6.8.1 *Integer Constraints for non-continuous limits*

When the limits on a set of circuits, transformers, and/or market nodes is dependent on the sign of another variable (indicating a direction of flow, for example), then a decision must be made as to what sign that variable will have, and hence which limits shall apply. Where appropriate an integer optimisation may be used to determine the most appropriate sign and set limits accordingly. These integer constraints existed only in the mixed constrains section 3.9 and will be subject to approval in accordance with the procedures established for those constraints. Such integer constraints will effectively force the model to examine an LP solution for each possible constraint limit condition, and then select the lowest cost solution from those options.

6.8.2 *Reserve sharing*

Integer constraints have been introduced to the model formulation to represent the non-convex operating region of the HVDC when sharing reserves in the reverse direction. Integer variables are used to identify the HVDC sending island and also identify the zone within which the HVDC is operating, which in turn affects the constraints that limit the quantity of reserves that can be shared across the HVDC in the reverse direction. The $InZone_{i,c,rrz}$ integer variable is also used to model the availability of reserve sharing in the forward direction when roundpower is disabled.

The lambda formulation is used to model the losses for the sent HVDC flow and the HVDC flow after accounting for shared reserves. These losses are used to adjust the shared reserves received in an island. Integer constraints are applied to the lambda formulation to ensure that at most two adjacent lambda variables are greater than zero in the model.

7 Post-Solve Checks

The following post-solve checks occur immediately after the solve and they can result in the model being re-solved. While it is necessary to perform some post-processing calculations in order to prepare the inputs required by these checks, the post-processing calculation of the final results only occurs after these checks and any associated re-solves have completed.

7.1 Circulating Flow Check

7.1.1 *Parameters*

Item	Description
$BadPriceFactorFromScarcity$	A multiplier applied to the maximum energy scarcity price to define positive and negative thresholds beyond which a price is assumed to have been set by a penalty price and not by a scarcity price. Used by the post-SOS1 price replacement to identify invalid prices.

7.1.2 *Cause of circulating flows*

The $ACLineFlow_k$ constraint defines AC line flow using two positive variables, each representing a directed flow as shown in Figure 8.

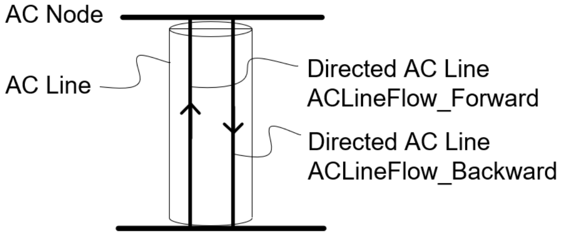


Figure 8: AC Line flow modelled as Directed AC Line flows

When the objective becomes non-convex (globally or locally) the formulation can produce circulating flows, as shown by the example in Figure 9.

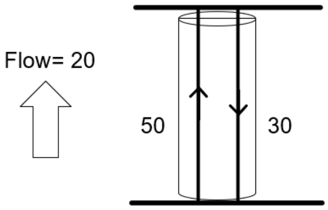


Figure 9: Example of circulating flows on an AC Line or HVDC Pole

Similarly, due to the modelling of HVDC Poles and Links as shown in Figure 10 and Figure 11, in non-convex situations the model can schedule circulating flow between the directed HVDC Lines within the HVDC Poles (e.g., flow on both BEN_HAY2.1 and HAY_BEN2.1), or between the directed HVDC Links, i.e., flow on both the NORTH link and the SOUTH link.

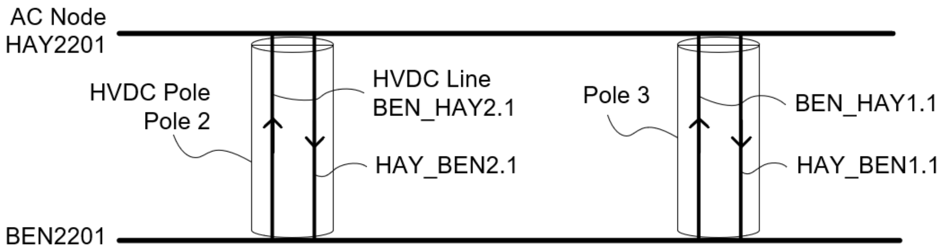


Figure 10: HVDC Flow modelled as directed HVDC Lines within Poles

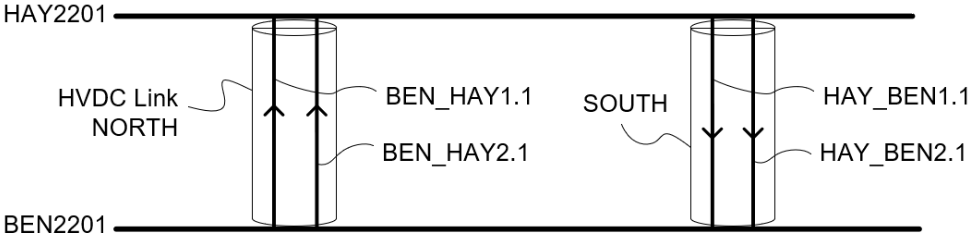


Figure 11: HVDC Flow modelled as directed Links

7.1.3 *Removal of circulating flows*

If post-processing detects circulating flows then the LP Model will be re-solved with the following SOS1 constraints active:

7.1.3.1. $$\sum_{\substack{po,h \\ HVDCLINEPOLE(po,h)}} HVDCLineFlow_h \leq \max_{\substack{po,h \\ HVDCLINEPOLE(p,h)}} HVDCLineCapacity_h$$

$$\forall po \in HVDCPOLES, \quad \forall h \in HVDCLINES$$

7.1.3.2. $$HVDCLinkFlow_{l=\,NORTH} + HVDCLinkFlow_{l=SOUTH} \leq \max_{l \in HVDCLINKS} HVDCLinkCapacity_l$$

$$\forall l \in HVDCLINKS$$

Because a SOS1 constraint only allows one of the variables in the constraint to be non-zero, when the SOS1 constraints are active they will enforce a single non-zero HVDCLineFlow within each HVDC Pole, and a single non-zero HVDCLinkFlow, i.e., either North or South.

For AC Lines the following constraints are applied to remove circulating flows:

7.1.3.3. $$ACLineFlow_{F(k)}^{Directed} \leq ACLineCapacity_{F(k)}^{Directed} \times ACLineFlowBinary_{F(k)}^{Directed}$$

$$\forall k \in ACLINES$$

7.1.3.4. $$ACLineFlow_{B(k)}^{Directed} \leq ACLineCapacity_{B(k)}^{Directed} \times ACLineFlowBinary_{B(k)}^{Directed}$$

$$\forall k \in ACLINES$$

7.1.3.5. $$ACLineFlowBinary_{F(k)}^{Directed} + ACLineFlowBinary_{B(k)}^{Directed} = 1$$

$$\forall k \in ACLINES$$

In addition to these constraints, if non-physical losses are detected on an HVDC branch then the following SOS2 constraint is activated which will prevent loss segment skipping in the lambda formulation.

7.1.3.6. $\sum_{bp=1}^{HVDCBreakpoints_l} Lambda_{l,bp} = 1$

If there is a requirement to schedule circulating power flow between HVDC poles then the integer constraints that prevent the circulating flows on HVDC poles will be disabled. This feature will only be used during commissioning.

7.1.4 *Replacement of invalid prices after SOS1*

Following a SOS1 solve there is the possibility of invalid prices at spur AC Nodes. This section described the post-processing that detects and replaces these invalid prices.

7.1.4.1. Cause of invalid prices

As described in the Circulating Flow Check section above, when circulating flows are detected a SOS1 solve is run that automatically restricts the flow to zero on one of the Directed AC Lines for each AC Line. This removes the possibility of circulating flows but also has the potential to produce invalid prices at AC Nodes which are connected to the system by AC Lines with zero scheduled flow.

The first contributing factor to invalid prices is that an AC Line with zero flow has no impact on the objective function value and therefore the direction that SOS1 restricts to zero is arbitrary.

The second contributing factor is that the energy price is based on the change in the objective function value due to a marginal change in the energy requirement at the AC Node. This marginal change can be an increase or a decrease, normally the price impact would be the same regardless but, when combined with the arbitrary restriction of branch flow by SOS1 for an AC Line with zero flow, can give rise to invalid price scenarios.

For an AC Node with zero flow in or out, the arbitrary direction of the SOS1 branch restriction could result in valid prices as shown in Figure 12.

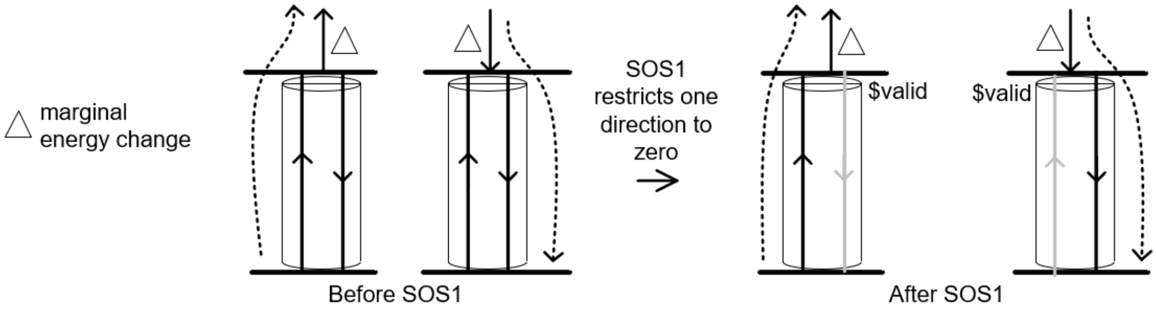
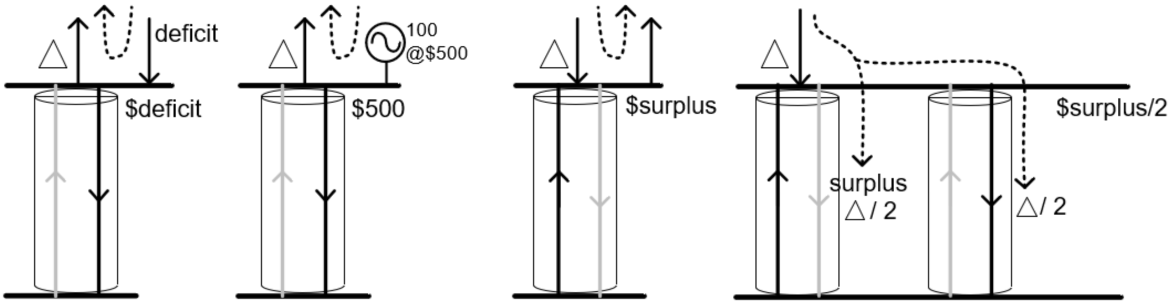


Figure 12: AC Node with zero flow, options for a valid price after SOS1

However, the SOS1 branch restriction could also restrict flow such that a marginal change can only be balanced by a deficit or surplus quantity, as shown in Figure 13, resulting in invalid prices.



a) i) a) ii) b) i) b) ii)

Figure 13: AC Node with zero flow, options for an invalid price after SOS1

7.1.4.2. Invalid price scenarios

The invalid price scenarios are as follows:

a) In the scenario where SOS1 restricts flow entering the AC Node and the price is based on energy leaving the AC Node then either:

  i) If there are no offers associated with the AC Node, then the energy balance constraint can only be balanced by deficit energy, resulting in a deficit energy price at the AC Node.

  ii) Or else if there are offers associated with the AC Node, then the AC Node price may be set by the offer price.

b) If SOS1 restricts outflow from the AC Node and the price is based on energy entering the AC Node, then either:

  i) The energy balance constraint is balanced by surplus energy, resulting in a surplus energy price at the AC Node.

  ii) Or else if there is a parallel Directed AC Line that is unrestricted for outflow, then a proportion of the energy can leave via flow on the unrestricted line with the necessary parallel flow on the restricted line achieved by surplus branch flow. The proportion of the outflow that is met by surplus flow will set the AC Node price at a proportion of the surplus price.

c) The other possibility is that the SOS1 constraints isolate the AC Node such that its shadow price is zero.

7.1.4.3. Detecting invalid prices

An AC Node is flagged as having an invalid price if it:

a) is not dead, and

b) has zero flow in and zero flow out, and

c) has zero net injection, and

d) has a zero deficit or surplus, and

e) has an energy price that is either

  i) zero, or

  ii) set by the lowest priced non-zero energy offer mapped to that AC Node, or

  iii) set by a penalty price.

For an RTD schedule, when determining if the price is set by the lowest priced non-zero energy offer the check needs to account for the impact of $PenaltyForMWChangeOf5Mindispatch$ which will either add to or subtract from the energy price depending on whether the marginal MW would move generation towards or away from $Generation_g^{Start}$.

When determining if the price is set by a penalty price the check must consider that such a price may be some proportion of the penalty price. For example as illustrated in Figure 13 b) ii), when SOS1 restricts directional flow on one AC Line but a parallel line is unrestricted, the penalty price will only be incurred for the proportion of flow that is met by surplus flow on the restricted line. To account for this, the check will look for a price that is less than the full penalty price but high enough that it cannot be a valid price set by $EnergyScarcityPrice_{es,j}$. Hence a price is determined to have been set by a penalty price if it is outside of the range:

$$\pm BadPriceFactorFromScarcity \times \max_{es,j} EnergyScarcityPrice_{e,j}$$

7.1.4.4. Replacing invalid prices

As shown in Figure 14 an AC Node flagged as having an invalid price will have its price replaced by the adjacent bus price. The adjacent bus price is checked and if it has an invalid price then the next adjacent bus price is checked. There is a limit on the number of steps.

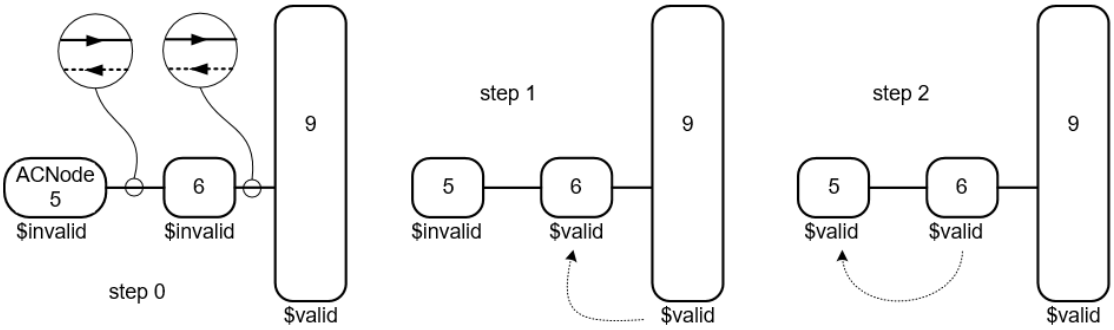


Figure 14: Example of replacing invalid prices after a SOS1 solve (two steps shown)

While the adjacent bus price may not meet any of the conditions for an invalid price, it may have had its price set by the bus with the invalid price if it is connected to that bus by a lossless AC Line, as illustrated in Figure 15. Hence there is an additional check to confirm that the adjacent bus does not have the same price as the bus with the invalid price.

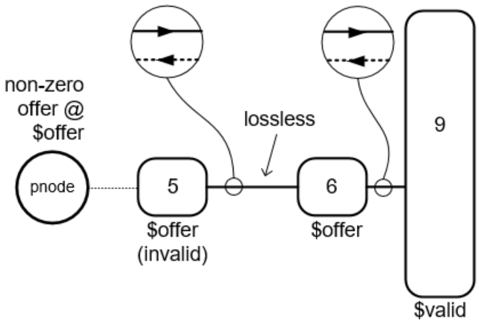


Figure 15 Potential for the transfer of invalid price

If the adjacent bus has the same price as the invalid price, then it is flagged as having an invalid price. Additionally, any buses that may have used this bus to obtain a replacement price are also flagged as having an invalid price.

7.2 Energy Shortfall Check

7.2.1 *Overview*

To ensure that energy shortfall results are as accurate as possible the Energy Shortfall Check is applied to Schedule Types that produce settlement prices, i.e., RTD, RTDP, PRS and any energy shortfall that is determined to be potentially inaccurate is removed.

7.2.2 *Parameters*

Item	Description
$EnergyShortfall_{pn}$	Pnode energy shortfall calculated as the sum over the scarcity energy blocks associated with the Pnode of the difference between the block limit and the block cleared
$DidShortfallTransfer_{pn}$	Binary value with an initial value of False that is set to True if $PnodeRequiredLoad_{pn}$ is adjusted by the Energy Shortfall Check
$MaxSolveLoops$	The maximum number of times that the Energy Shortfall Check will re-solve the model
$EnergyShortfallRemovalMargin$	When the Energy Shortfall Check removes a shortfall from a Pnode, this small margin is added to the removed amount in order to prevent any associated binding ACLine constraint from solving exactly on the endpoint and potentially setting a binding price. Note that if the Pnode is in the set of $DEADPNODES_{pn}$ then this margin is not applied

7.2.3 *Processing*

Figure 16 provides an overview of the Energy Shortfall Check. In the context of this diagram the components of the process are as follows.

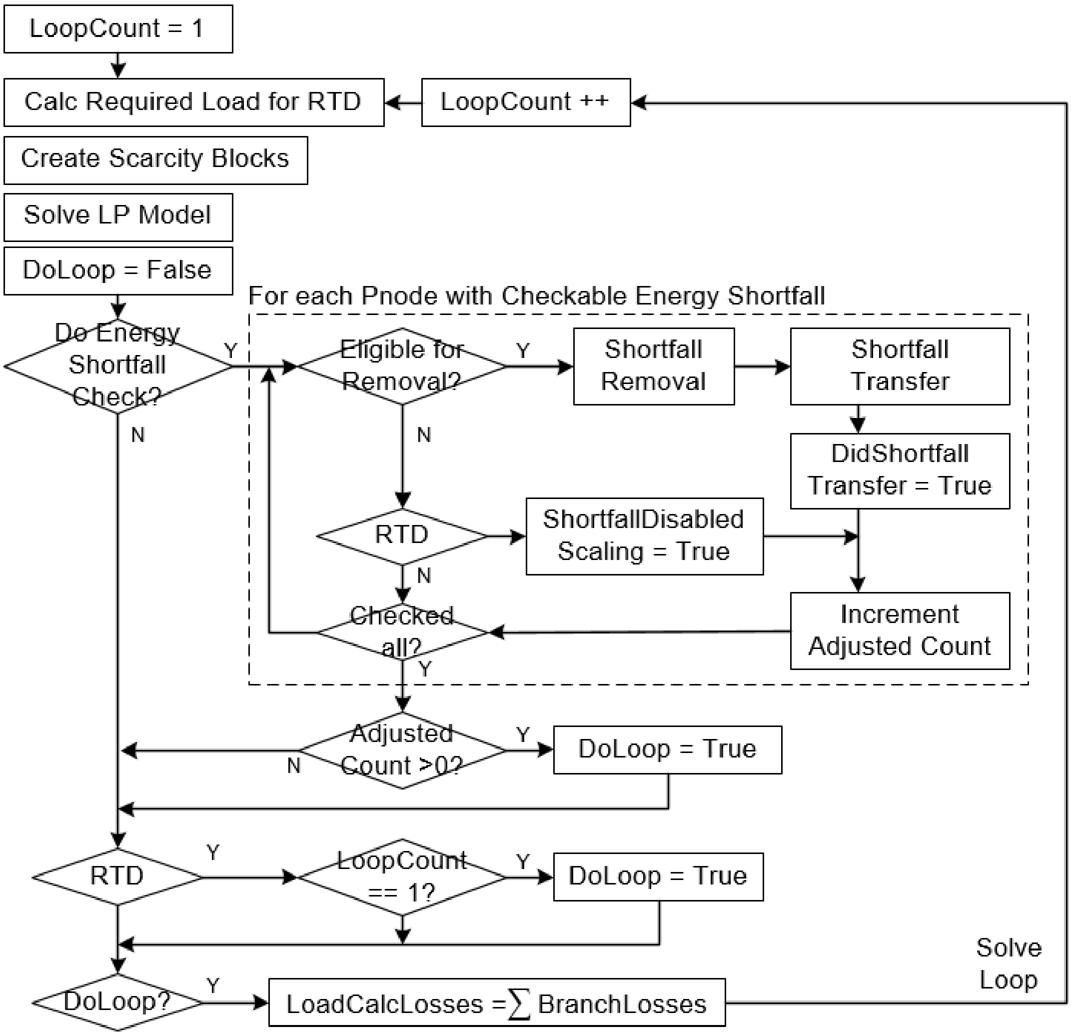


Figure 16: Overview of Energy Shortfall Check

a) **Checkable Energy Shortfall** If a Pnode has an *EnergyShortfall*$_{pn}$ greater than zero and the Pnode has *LoadIsOverride*$_{pn}$ set to False and the Pnode has *InstructedShedActive*$_{pn}$ set to False, then *EnergyShortfall*$_{pn}$ is checked.

b) **RTD** In the context of the Energy Shortfall Check, RTD refers to RTD and RTDP schedule types. References to the RTD Required Load calculation refer to the process described in the section Required Load for Real Time Dispatch.

c) **Eligible for Removal** An *EnergyShortfall*$_{pn}$ is eligible for removal if there is evidence that it is due to a modelling inconsistency (as described below), or if the RTD Required Load calculation used an estimated initial load rather than an actual initial load, or if the Pnode is in the set of *DEADPNODES*$_{pn}$.

d) **Shortfall Removal** If the shortfall at a Pnode is eligible for removal then a Shortfall Adjustment quantity is subtracted from the *PnodeRequiredLoad*$_{pn}$ in order to remove the shortfall. If the Pnode is in the set of *DEADPNODES*$_{pn}$ then the Shortfall Adjustment is equal to *EnergyShortfall*$_{pn}$ otherwise it is equal to *EnergyShortfall*$_{pn}$ plus *EnergyShortfallRemovalMargin*. If the adjustment would make *PnodeRequiredLoad*$_{pn}$ negative then *PnodeRequiredLoad*$_{pn}$ is assigned a value of zero. The adjusted Pnode has *DidShortfallTransfer*$_{pn}$ set to True so that the RTD Required Load calculation does not recalculate its *PnodeRequiredLoad*$_{pn}$.

e) **Shortfall Transfer** If the previous step adjusts *PnodeRequiredLoad*$_{pn}$ then the processing will search for a transfer target Pnode to receive the Shortfall

Adjustment quantity (the search process is described below). If a transfer target Pnode is found then the Shortfall Adjustment is added to the $PnodeRequiredLoad_{pn}$ of the transfer target Pnode and the $DidShortfallTransfer_{pn}$ of the transfer target Pnode flag is set to True.

f) **Scaling Disabled** For an RTD schedule type, when an $EnergyShortfall_{pn}$ is checked but the shortfall is not eligible for removal then $ShortfallDisabledScaling_{pn}$ is set to True which will prevent the RTD Required Load calculation from scaling $InitialLoad_{pn}$.

g) **Create Scarcity Blocks** If any $PnodeRequiredLoad_{pn}$ are adjusted then all Energy Scarcity blocks are recalculated.

h) **Solve Loop** If the number of solve loops has reached *MaxSolveLoops* then the shortfall check will still run but will take no action other than logging, and there will be no re-solve.

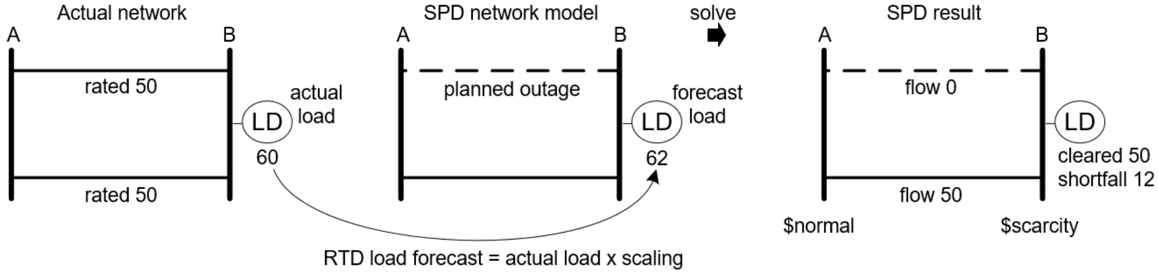


Figure 17: Illustration of modelling inconsistency

i) **Potential for Modelling Inconsistency** As illustrated by the example in Figure 17, the potential for a modelling inconsistency exists because while planned outages are incorporated in the Network Model they are not necessarily reflected in the forecast load.

j) **Detecting Modelling Inconsistency** The Shortfall Check detects a modelling inconsistency if the shortfall Pnode has an associated ACLine in $SHORTFALLACLINES_{pn}$ and this ACLine is removed from the model.

k) **Shortfall Transfer Target** In the Shortfall Transfer step, the search for a transfer target Pnode proceeds as follows.

The first choice candidate for price transfer source is the $PnodeTransferPnode_{pn}$ of the target Pnode. If the candidate is ineligible then the new candidate will be the $PnodeTransferPnode_{pn}$ of the candidate, if any, but only if this new candidate has not already been visited in this search. The process of locating and checking candidates will continue until an eligible transfer Pnode is located or until no more candidates are found.

A candidate Pnode is not eligible as a target if it has a non-zero $EnergyShortfall_{pn}$ in the solution being checked or had one in the solution of a previous solve loop, or if the candidate Pnode has $LoadIsOverride_{pn}$ set to True, or if the candidate Pnode has $InstructedShedActive_{pn}$ set to True, or if the Pnode with the shortfall is not in Electrical Island 0 and the $ElectricalIsland_{pn}$ of the candidate Pnode is not the same as the $ElectricalIsland_{pn}$ of the Pnode with the shortfall, or if the candidate Pnode is in the set of $DEADPNODES_{pn}$, or if the candidate Pnode is not in the same physical island.

8 Post-Processing

8.1 System Losses

Island power system losses are calculated as follows and assigned to the *LoadCalcLosses* parameter for use by the RTD Required Load calculation described in section 4.12.

8.1.1.1. $$SystemLosses_i = \sum_{l \in R_{HVDC}(n(i))} HVDCLinkLosses_l + \sum_{l \in HVDCLINKS_{n(i)}} \frac{1}{2} \times HVDCLinkFixedLosses_l + \sum_{q \in R_{AC}(n(i))} ACLineLosses_q^{Directed} + \sum_{k \in ACLINES_{n(i)}} \frac{1}{2} \times ACLineFixedLosses_k$$

$$\forall i \in ISLANDS$$

8.1.1.2. $$LoadCalcLosses_i = SystemLosses_i$$

$$\forall i \in ISLANDS$$

8.2 Dead Electrical Island

8.2.1 *Determination*

If an Electrical Island > 2 has no positive load or if the total energy offered within the Electrical Island is zero or if security constraints limit the total cleared energy in the island to zero then the Electrical Island is determined to be a dead Electrical Island and each ACNode in the Electrical Island is added to the set of DeadACNodes and each Pnode in the Electrical Island is added to the set of DeadPnodes.

8.2.2 *Zeroing of cleared quantities*

For each ACNode in the set of DeadACNodes, any cleared generation (which may occur due to fixed losses) is assigned a quantity of zero and any cleared reserve that is not of reserve type IL is assigned a quantity of zero. Reserve type IL is excluded because the ACNode associated with reserve type IL does not necessarily represent the physical location providing the IL reserve.

8.3 Energy Prices

8.3.1 *Input Parameters*

Item	Description
$PnodeTransferPnode_{pn}$	Pnode which has been deemed potentially suitable to provide a replacement price for *pn*
$SurplusPriceThreshold$	Negative threshold below which a negative energy price is considered to have been set by a penalty price

8.3.2 *Parameters*

Item	Description
$ACNodePrice_n$	The energy price for an ACNode *n* is the dual variable value (shadow price) of the AC Node energy balance constraint 6.4.1.1

8.3.3 *Dead Price Assignment*

For each ACNode in the set of DeadACNodes, if the ACNode has zero scheduled load but an associated energy scarcity block with a non-zero limit then the ACNode is assigned the price of the highest-priced energy scarcity block, and if the ACNode is in a Dead Electrical Island then all ACNodes in the electrical island are assigned this price.

8.3.4 *Pnode Price*

For each ACNode, the energy price $ACNodePrice_{en}$ is assigned to the $EnodePrice_{en}$ of the Enodes associated with the ACNode. Enode price is then assigned to Pnode price:

8.3.4.1.
$$PnodePrice_{pn} = \sum_{pn,en} PnodeEnodeWeight_{pn,en} \times EnodePrice_{en}$$
$$\forall en \in Enodes, \forall pn \in Pnodes$$

8.3.5 *Dead Price Replacement*

Dead Price Replacement is applied to the Schedule Types that produce settlement prices, i.e., RTD, RTDP, PRS. For each Pnode in the set of DeadPnodes the Dead Price Replacement processing will search for a suitable price transfer source Pnode to provide a replacement price.

The first choice candidate for price transfer source is the $PnodeTransferPnode_{pn}$ of the target Pnode. If the candidate is ineligible then the new candidate will be the $PnodeTransferPnode_{pn}$ of the candidate, if any, but only if this new candidate has not already been visited in this search. The process of locating and checking candidates will continue until an eligible transfer Pnode is located or until no more candidates are found.

A candidate Pnode is not eligible as a price source if it is in the set of $DEADPNODES_{pn}$, or if the candidate Pnode is not in the same physical island.

Note that a candidate Pnode with a shortfall is eligible.

If an eligible price transfer source is found then the energy price of the dead Pnode and its associated ACNode are assigned the energy price of the transfer Pnode. If no eligible price is found then the dead Pnode and its associated ACNode are assigned a price of zero.

8.3.6 *Surplus Price Replacement*

If a Pnode has an energy price less than the *SurplusPriceThreshold*, and also there either exists a non-zero Surplus Generation quantity in the same Electrical Island as the Pnode or else the Pnode is dead, then the Pnode and its associated ACNode are assigned an energy price of zero.

8.4 Disconnected Pnodes

Each Pnode that has a scheduled load of zero and is in the set of DeadPnodes is added to the set of DisconnectedPnodes.

8.5 Reserve Prices

Item	Description
$ReservePrice_{i,c}$	For each island *i* and reserve class *c*, the reserve price is given by the shadow price of constraint 6.5.5.1, which clears island reserve, **except where** the cleared island reserve quantity is zero. In that case, the reserve price is given by the sum of the shadow prices of constraints 6.5.5.2–6.5.5.5, which together impose the requirement that sufficient reserve be scheduled to cover the risk.

8.6 AC Node imbalance

Post-processing assigns any deficit or surplus imbalance quantity from an AC Node to the Pnode or Pnodes that map to that AC Node. If there is more than one Pnode then the imbalance is distributed equally.

If two AC Nodes are connected by a zero-impedance AC Line (a.k.a. zero-impedance branch ZBR) then the imbalance due to the load at one AC Node can be covered by an imbalance quantity at another AC Node. For example, the situation shown in Figure 18 can occur. In this example, binding lines have necessitated deficit quantities in order to meet the Required Load of Load A and Load D.

The deficit at ACNode1 will be correctly assigned to Pnode A using the Pnode-AC Node mapping that was calculated by pre-processing.

However, because a zero impedance AC Line has no losses, the Load D at ACNode7 can be (and has been) balanced by a deficit at ACNode8. The deficit at ACNode8 is assigned to Pnode D via an infeasibility mapping. The infeasibility mapping is created by post-processing as described in the following section.

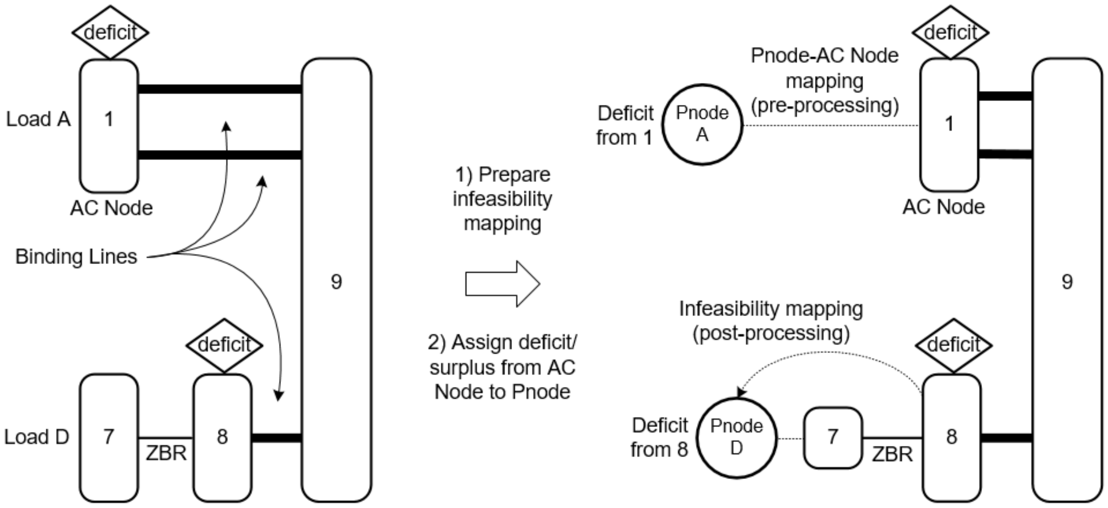


Figure 18 : Assigning AC Node imbalance to Pnodes

8.6.1 ***Infeasibility Mapping***

In order to ensure that infeasibility mappings are available if required to assign a deficit or surplus, if the result contains any imbalance quantities then a post-processing step is run to create infeasibility mappings between AC Nodes and Pnodes. The processing starts with "special" AC Nodes that have no associated Pnode, and "mapped" AC Nodes, that do have an associated Pnode, as shown in Figure 19.

Where a special AC Node is connected to a mapped AC Node via a zero-impedance AC Line (ZBR), an infeasibility mapping is created between the special AC Node and the Pnode of the mapped AC Node. The special AC Nodes that have been successfully mapped become the mapped AC Nodes for the next step and the process repeats for up to a pre-defined number of steps.

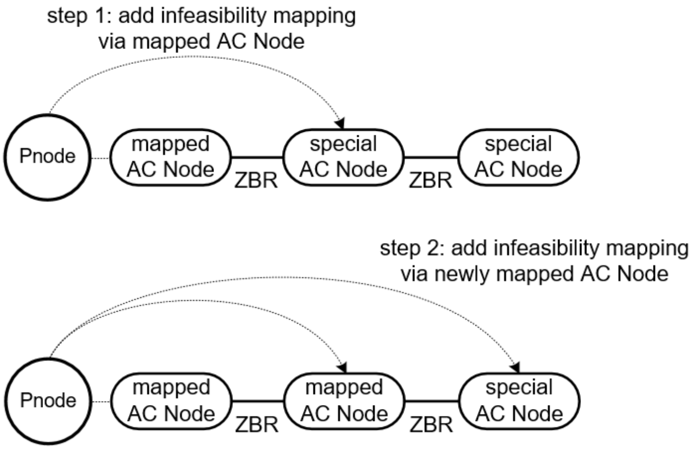


Figure 19 : Example of creating infeasibility mappings (two steps shown)

8.6.2 ***Imbalance at Dead Electrical Island***

If the Dead Electrical Island post-processing determines that an Electrical Island > 2 is a dead Electrical Island then all imbalances in that Electrical Island are set to zero, this is to account for any imbalances that may be present due to fixed losses.