Matlab files:

m files:

teran

teran_SR

teran_D

teran_LR_initial

teran_LR

Existing network (Short-Run)	Demand -Revenue model	Future network
SR1_inputs_market_industry	D1_input	LR1_inputs_market_industry
SR2_inputs_technology_spectrum	D2_demand_estimation	LR2_inputs_technology_spectrum
SR3_inputs_costs	D3_data_traffic_forecast	LR3_inputs_costs
SR4_inputs_finance	D4_demand_forecast	LR4_inputs_finance
SR5_supply_cell	D5_revenues	LR5_supply_cell
SR6_demand_cell	D6_marginal	LR6_demand_cell
SR7_site_infrastructure_positions		
SR8_site_infrastructure_layout		LR7_sites_to_invest_due_to_traffic
SR9_network_information		LR8_investments_scenario
SR10_assets_value		LR9_network_information
SR11_operating_costs		LR10_investments_cost
		LR11_assets_value
		LR12_operating_costs
		LR_D_marginal

Functions:

teranGUI

number_of_cells

cell_capacity_Mbps

cell_throughput_Mbps

basestations_initial_position

number_of_sites

define_site_infrastructure_struct

distribute_RAN_to_sites

io_write_site_infrastructure_to_file

io_read_site_infrastructure_from_file

extract_network_information

extract_network_information_all_RAN

number_of_TRx

fixed_assets_cost_cells

fixed_assets_cost_all_RAN

operating_cost_cells

operating_cost_all_RAN
calculate_inverse_demand
cell_throughput_progression_Mbps
convert_trafficGrowth_to_time
invest_to_sites
network_territory_coverage
cell_throughput_LR_Mbps
cell_throughput_correction_LR_Mbps
investment_cost_cell
investment_cost_all_RAN
disposal_value_cell
disposal_value_all_RAN

other functions:
kmeans
rangesearch

txt files:
cells_EUTRAN_CA
cells_EUTRAN_s
cells_UTRAN
cells_GERAN

Inputs

SR1_Market and Industry

Variable	*Data type*	*Example*	*Comment*
inputs_market_industry	Struct		
demographics	Nested struct		
land_Area_km2	double	9	
land_Area_country_km2	double	303900	
population	double	66449	
population_country	double	5478308	
mobile_market	Nested struct		
mobile_penetration	double	1.727	
market_share	double	0.39	
proportion_of_data_users	double	0.851	
service	Nested struct		
mobile_data_raffic_country_TB_mon	double	90474752/1024	2016
traffic_distribution_among_sites	double	15/50	space domain e.g., 50% of traffic is carried by 15% of sites(cells)
BHratio	double	0.12	time domain
ULtoDLratio	double	0.1	link domain, uplink to downlink ratio

SR2_Technology

Variable	*Data type*	*Example*	*Comment*
inputs_technology	Struct		
spectrum	Nested struct		
paired_700_MHz	double	10	
paired_800_MHz	double	10	
paired_900_MHz	double	11.4	
paired_1800_MHz	double	24.8	
paired_2100_MHz	double	19.8	
paired_2600_MHz	double	25	
unpaired_2100_MHz	double	5	
EUTRAN_s	Nested 1 x N struct		N = num of configurations
cellgroupID	double	1	
configuration	char	'EUTRAN@2600, LTE, R8, 20 MHz, 64 QAM, 2x2 MIMO, 172 Mbps'	
bands	double	2.6	
BW	double	20	
typical_BW_efficiency_bps_per_Hz	double	1.56	
cell_range_km	double	0.2	
territory_coverage	double	0.36	
development_year	double	2011	
max_power_basestation	double	1350	Maximum power (Pmax) for the 3sectored base station in Watt
share_of_max_power_in_idle_mode_basestation	double	0.58	Proportion of power consumption (po) during idle, non-operational mode
EUTRAN_CA	Nested 1 x N struct		N = num of configurations
cellgroupID	double	1	
configuration	char	'EUTRAN@2600+1800, LTE-A, R11, CA, 20+20 MHz, 64 QAM, 2x2 MIMO, 344 Mbps'	
CA_config	1 x N double	[1,0,1,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0]	N = num of EUTRAN_s configurations
territory_coverage	double	0.6	territory coverage of the cell with the lowest frequency in the CA configurations
development_year	double	2015	
max_power_basestation	double	1500	
share_of_max_power_in_idle_mode_basestation	double	0.58	
UTRAN	Nested 1 x N struct		N = num of configurations
cellgroupID	double	1	

	configuration	char	'UTRAN@2100, DC-HSPA+, R8, 64 QAM, 42 Mbps'	
	bands	double	2.1	
	BW	double	9.9	
	typical_BW_efficiency_bps_per_Hz	double	1.05	
	cell_range_km	double	0.23	
	territory_coverage	double	1	
	development_year	double	2010	
	max_power_basestation	double	1500	
	share_of_max_power_in_idle_mode_basestation	double	0.62	
GERAN		Nested 1 x N struct		N = num of configurations
	cellgroupID	double	1	
	configuration	char	'GERAN@900, GSM/EDGE'	
	bands	double	0.9	
	BW	double	0.95	
	typical_BW_efficiency_bps_per_Hz	double	0.4	
	cell_range_km	double	0.4	
	territory_coverage	double	0.5	
	reuse_factor	double	4	
	development_year	double	2004	
	max_power_basestation	double	2145	
	share_of_max_power_in_idle_mode_basestation	double	0.65	
other_RAN		Nested struct		
	BSC_capacity_TRx	double	2048	TRx per BSC
	RNC_capacity_nodeB	double	512	nodeBs per RNC
core		Nested struct		
	HSS_VLR_EIR_capacity_subs	double	4000000	capacity of the entities common to PS and CS domain (in number of subsciptions)
	EPC_capacity_mbps	double	40000 (MME, SGW, PGW)	
	PS_capacity_mbps	double	10000 (SGSN, GGSN)	
site_types		Nested 1 x N struct		N = number of site types
	newest_site_development_year	double	2010	
	oldest_site_development_year	double	1995	
	name	char	'owned_rooftop'	
	share	double	0.09	out of total number of sites
extra		Nested struct		
	GSM_channel_MHz	double	0.2	
	TRx_data_usage_share	double	0.25	2 out of 8 timeslots are used for data service. The rest are allocated to voice
	UMTS_data_usage_share	double	0.5	50% of UMTS sources (spectrum, network elements) are used for

					data services. The rest is for voice etc
		transmission_overhead	double	0.05	
		owned_RAN_transmission_share	double	0.5	share between leased and owned lines
		owned_core_transmission_share	double	0.8	share between leased and owned lines
		core_data_usage_share	double	0.3	
		operation_share_for_data_service	double	0.3	
		core_energy_consumption_parameter	double	0.1	10% of RAN consumption
		RAN_controllers_energy_consumption_parameter	double	0.15	15% additional energy to RAN from RNC and BSC
		default_sector_number	double	3	
		sites_to_basestations_ratio	double	0.4	
		caproom	double	0.1	It is assumed that the existing network has been built to cover the demanded traffic plus e.g.,10%
		antDir	1x3 double	[90,210,330]	antenna direction of sector in degrees (reference is north)
		RAN_labor	double	2	number of employees for RAN operation and data service
		CORE_labor	double	2	number of employees for CORE operation and data service

SR3_Costs

Variable	*Data type*	*Example*	*Comment*
inputs_costs	Struct		
operating_cost	Nested struct	OPEX	
cost_of_good_sold	Nested struct	Network: Monthly cost per unit	Operating expenditure (Network operations)
site_rental_tower	double	500	rent, euro per site per month
site_rental_shared_tower	double	500/3	rent, euro per site per month, assuming 3 mobile operators
site_rental_rooftop	double	600	rent, euro per site per month
site_rental_shared_rooftop	double	600/3	rent, euro per site per month, assuming 3 mobile operators
RAN_network_operation_maintenance	double	0.15	15% of all RAN cost_of_good_sold in the cell
RANtoCORE_network_operation_maintenance	double	0.15	15% of all RANtoCORE cost_of_good_sold in the cell
CORE_network_operation_maintenance	double	0.15	15% of all CORE cost_of_good_sold in the cell
RAN_leased_transmission_lines_Mbps	double	100	euro per Mbps
RANtoCORE_leased_transmission_lines_Mbps	double	100	euro per Mbps
CORE_leased_transmission_lines_Mbps	double	100	euro per Mbps
energy_MWh	double	40	euro per MWh
personnel	double	5000	euro per month
RAN_other	double	0.01	1% of RAN cost_of_good_sold in the cell
RANtoCORE_other	double	0.01	1% of RANtoCORE cost_of_good_sold in the cell
CORE_other	double	0.02	2% of CORE cost_of_good_sold in the cell
interconnection_and_other_fees	Nested struct	0.15	percentage of operating cost for data service
selling_general_and_administrative	Nested struct		Non_network operating expenditures (data service share of common costs for all services)
marketing_and_advertising	double		
customer_acquisition_retention	double		
devices_subsidies	double		
personnel	double		
property_material	double		
accounting_and_legal	double		
information_technology	double		
energy	double		energy consumption of non-network fixed assets (IT, retail)
other	double		
spectrum	Nested struct		
paired_700_cost	22,000,000		

		paired_700_acquisition_year	2017		
		paired_700_licence_valid_year	2033		
		paired_800_cost	33,340,000		
		paired_800_acquisition_year	2014		
		paired_800_licence_ valid_year	2033		
		paired_900_cost	692,818		
		paired_900_acquisition_year	2017		
		paired_900_licence_ valid_year	2019		
		paired_1800_cost	1,507,182		
		paired_1800_acquisition_year	2017		
		paired_1800_licence_ valid_year	2019		
		paired_2100_cost	17,707,317		
		paired_2100_acquisition_year	1999		
		paired_2100_licence_ valid_year	2019		
		paired_2600_cost	834,700		
		paired_2600_acquisition_year	2009		
		paired_2600_licence_ valid_year	2029		
		unpaired_2100_cost	4,292,683		
		unpaired_2100_acquisition_year	1999		
		unpaired _2100_licence_ valid_year	2019		
	capital_expenditure		Nested struct		CAPEX: Unit price (euro per network component)
		RAN	Nested struct		
		site_tower	double	50000	
		site_shared_tower	double	50000/3	
		site_rooftop	double	25000	
		site_shared_rooftop	double	25000/3	
		basestation_BTS	double	30000	
		basestation_NodeB	double	50000	
		basestation_eNodeB_s	double	60000	
		basestation_eNodeB_CA	double	60000	
		controller_BSC	double	350000	
		controller_RNC	double	1000000	
		owned_transmission_line	double	100	cost per Mbps
		other_tangible_fixed_asset	double	0.01	percentage of all the RAN cost (related to data service)
		RAN_to_core	Nested struct		
		owned_transmission_line	double	50	cost per Mbps
		other_tangible_fixed_asset	double	0.05	percentage of all the RAN_to_core_trans cost (related to data service)
		core	Nested struct		
		packet_switch_core	double	3000000	SGSN, GGSN
		evolved_packet_core	double	10000000	MME, SGW, PDW
		common_core_entities	double	8000000	HSS
		owned_transmission_line	double	200	

			other_tangible_fixed_asset	double	0.05	percentage of all the core cost (related to data service)
		non_network_tangible_fixed_assets		double	0.002	percentage of total network cost (related to data service)
		intangible_fixed_assets		Nested struct		IMPEX
			development_cost_site	double	35000	
			development_cost_other	double	500	
			other_intangible_fixed_asset	double	0.001	percentage of total network cost (related to data service)
	depreciation_and_amortization			Nested struct		
		network_depreciation		double		
		non_network_depreciation		double		
		spectrum_amortization		double		
		other_amortization		double		for other intangible fixed assets
	research_and_development			double		

SR4_finance

Variable	*Data type*	*Example*	*Comment*
inputs_finance	Struct		
current_year	double	2017	
wacc	double	0.09	
economic_asset_life			years
buildings_and_constructions	double	30	e.g., sites
machinery_and_equipment_in_buildings	double	18	e.g., sites equipment
telecom_networks	double	12	e.g., base stations, controllers, transmission, common network elements
exchanges_and_concentrators	double	8	e.g., core
equipment_for_network_and_exchanges	double	5	e.g., other tangible network fixed asset
telecom_terminals	double	3	
other_machinery_or_equipment	double	4	e.g., non-network tangible fixed assets
other	double	10	e.g., non-network intangible fixed assets
income_expense	Nested struct		
taxes		0.22	
interest			
liabilities			
other			

Building initial network infrastructure

A. From the scratch with input variables

SR5_Supply_cell: Adding fields to inputs.technology struct

Variable	*Data type and calculation*	*Comment*
inputs_technology	Struct	
EUTRAN_s	Nested 1 x N struct	N = num of configurations
cells	double, using: function number_of_cells(inputs_market_industry. demographics . land_Area_km2 inputs_technology. EUTRAN_s. territory_coverage inputs_technology. EUTRAN_s. cell_range_km)	
basestations	double, using: inputs_technology. EUTRAN_s. cells inputs_technology.extra.default_sector_number	
cell_capacity_Mbps	double, using: function cell_capacity_Mbps(inputs_technology.EUTRAN_s.BW inputs_technology.EUTRAN_s.typical_BW_efficiency_bps_per_Hz)	
EUTRAN_CA	Nested 1 x N struct	N = num of configurations
carriers	double, using: inputs_technology.EUTRAN_CA.CA_config inputs_technology.EUTRAN_s.cell_range_km	
bands	1 x N double, using: inputs_technology.EUTRAN_CA.CA_config inputs_technology.EUTRAN_s.bands	N = num of carriers
BW	1 x N double, using: inputs_technology.EUTRAN_CA.CA_config inputs_technology.EUTRAN_s.BW	N = num of carriers
cell_ranges_km	1 x N double, using: inputs_technology.EUTRAN_CA.CA_config inputs_technology.EUTRAN_s.cell_range_km	N = num of carriers
cell_range_km	double, using: inputs_technology.EUTRAN_CA. cell_ranges_km	
cells	double, using: function number_of_cells(inputs_market_industry. demographics . land_Area_km2 inputs_technology. EUTRAN_CA. territory_coverage inputs_technology. EUTRAN_CA. cell_max_range_km)	
basestations	double, using: inputs_technology. EUTRAN_CA. cells inputs_technology.extra.default_sector_number	
cells_non_aggregated	double, using: inputs_technology. EUTRAN_CA. cells	

			inputs_technology. EUTRAN_CA. carriers	
		cell_capacity_Mbps	double, using: inputs_technology.EUTRAN_CA.CA_config inputs_technology.EUTRAN_s. cell_capacity_Mbps	
		territory_coverage_all_carriers	double, using: inputs_market_industry. demographics . land_Area_km2 inputs_technology.EUTRAN_CA. cell_ranges_km inputs_technology.EUTRAN_CA.cells	
	UTRAN		Nested 1 x N struct	N = num of configurations
		cells	double, using: function number_of_cells(inputs_market_industry. demographics . land_Area_km2 inputs_technology. UTRAN. territory_coverage inputs_technology. UTRAN. cell_range_km)	
		basestations	double, using: inputs_technology. UTRAN. cells inputs_technology.extra.default_sector_number	
		cell_capacity_Mbps	double, using: function cell_capacity_Mbps(inputs_technology.UTRAN.BW inputs_technology.UTRAN.typical_BW_efficiency_bps_per_Hz)	
	GERAN		Nested 1 x N struct	N = num of configurations
		cells	double, using: function number_of_cells(inputs_market_industry. demographics . land_Area_km2 inputs_technology. GERAN. territory_coverage inputs_technology. GERAN. cell_range_km)	
		basestations	double, using: inputs_technology. GERAN. cells inputs_technology.extra.default_sector_number	
		cell_capacity_Mbps	double, using: function cell_capacity_Mbps(inputs_technology.GERAN.BW inputs_technology.GERAN.typical_BW_efficiency_bps_per_Hz)	

SR6_Demand_cell: Adding fields to inputs.technology struct

Variable	*Data type and calculation*	*Comment*
inputs_technology	Struct	
EUTRAN_s	Nested 1 x N struct	N = num of configurations
cell_throughput_Mbps	1 x N double, using: function cell_throughput_Mbps(inputs_technology.EUTRAN_s.cell_capacity_Mbps inputs_technology.extra.caproom inputs_market_industry.service.traffic_distribution_among_sites inputs_market_industry.service.BHratio inputs_technology.EUTRAN_s.cells)	N = num of cells
EUTRAN_CA	Nested 1 x N struct	N = num of configurations
cell_throughput_Mbps	1 x N double, using: function cell_throughput_Mbps(inputs_technology.EUTRAN_CA.cell_capacity_Mbps inputs_technology.extra.caproom inputs_market_industry.service.traffic_distribution_among_sites inputs_market_industry.service.BHratio inputs_technology.EUTRAN_CA.cells)	N = num of cells
UTRAN	Nested 1 x N struct	N = num of configurations
cell_throughput_Mbps	1 x N double, using: function cell_throughput_Mbps(inputs_technology.UTRAN.cell_capacity_Mbps inputs_technology.extra.caproom inputs_market_industry.service.traffic_distribution_among_sites inputs_market_industry.service.BHratio inputs_technology.UTRAN.cells)	N = num of cells
GERAN	Nested 1 x N struct	N = num of configurations
cell_throughput_Mbps	1 x N double, using: function cell_throughput_Mbps(inputs_technology.GERAN.cell_capacity_Mbps inputs_technology.extra.caproom inputs_market_industry.service.traffic_distribution_among_sites inputs_market_industry.service.BHratio inputs_technology.GERAN.cells)	N = num of cells

SR7_Site_infrastructure: find initial site position

Variable	*Data type and calculation*	*Comment*
temp_EUTRAN_s_bs_initial_position	N x 2 double, using: function basestations_initial_position (inputs_market_industry.demographics.land_Area_km2 inputs_technology.EUTRAN_s(:).cell_range_km inputs_technology.EUTRAN_s(:).basestations)	N = num of base stations Coordination x, y
temp_EUTRAN_CA_bs_initial_position	N x 2 double using: function basestations_initial_position (inputs_market_industry.demographics.land_Area_km2 inputs_technology.EUTRAN_CA(:).cell_range_km inputs_technology.EUTRAN_CA(:).basestations)	N = num of base stations Coordination x, y
temp_UTRAN_bs_initial_position	N x 2 double, using: function basestations_initial_position (inputs_market_industry.demographics.land_Area_km2 inputs_technology.UTRAN(:).cell_range_km inputs_technology.UTRAN(:).basestations)	N = num of base stations Coordination x, y
temp_GERAN_bs_initial_position	N x 2 double using: function basestations_initial_position (inputs_market_industry.demographics.land_Area_km2 inputs_technology.GERAN(:).cell_range_km inputs_technology.GERAN(:).basestations)	N = num of base stations Coordination x, y
temp_all_bs_initial_position		Coordination of all base stations
temp_actual_sites	double, using: function number_of_sites(temp_all_bs_initial_position temp_EUTRAN_s_bs_initial_position temp_EUTRAN_CA_bs_initial_position inputs_technology.(temp_RAN).basestations, inputs_technology.(temp_RAN).basestations inputs_technology.extra.sites_to_basestations_ratio)	Number of sites
temp_sites_position	N x 2 double using: function keans(temp_all_bs_initial_position, temp_actual_sites)	N = num of base sites Coordination of all sites

SR8_Site_infrastructure: create the structure and distribute the RAN to sites

Define and initialize site_infrastructure

Using: function site_infrastructure = define_site_infrastructure_struct (number of sites)

Variable	Data type	Example	Comment
Site_infrastructure	1 x N Struct		N = number of sites
site_ID	char	’’	
site_type	char	’’	
longitude	double	0	
latitude	double	0	
development_year	double	0	
EUTRAN_s	1 x 1 struct		
sector	Nested 1 x N struct		N = num of sectors
cell_ID	char	’’	
configuration	char	’’	
band	double	0	
BW	double	0	
cell_range_km	double	0	
cell_capacity_Mbps	double	0	
cell_throughput_Mbps	double	0	
development_year	double	0	
EUTRAN_CA	1 x 1 struct		
sector	Nested 1 x N struct		N = num of sectors
cell_ID	char	’’	
configuration	char	’’	
band	double	0	
BW	double	0	
cell_range_km	double	0	
cell_capacity_Mbps	double	0	
cell_throughput_Mbps	double	0	
development_year	double	0	
UTRAN	1 x 1 struct		
sector2100	Nested 1 x N struct		N = num of sectors
cell_ID	char	’’	
configuration	char	’’	
band	double	0	
BW	double	0	
cell_range_km	double	0	
cell_capacity_Mbps	double	0	
cell_throughput_Mbps	double	0	
development_year	double	0	
sector900	Nested 1 x N struct		N = num of sectors
cell_ID	char	’’	
configuration	char	’’	

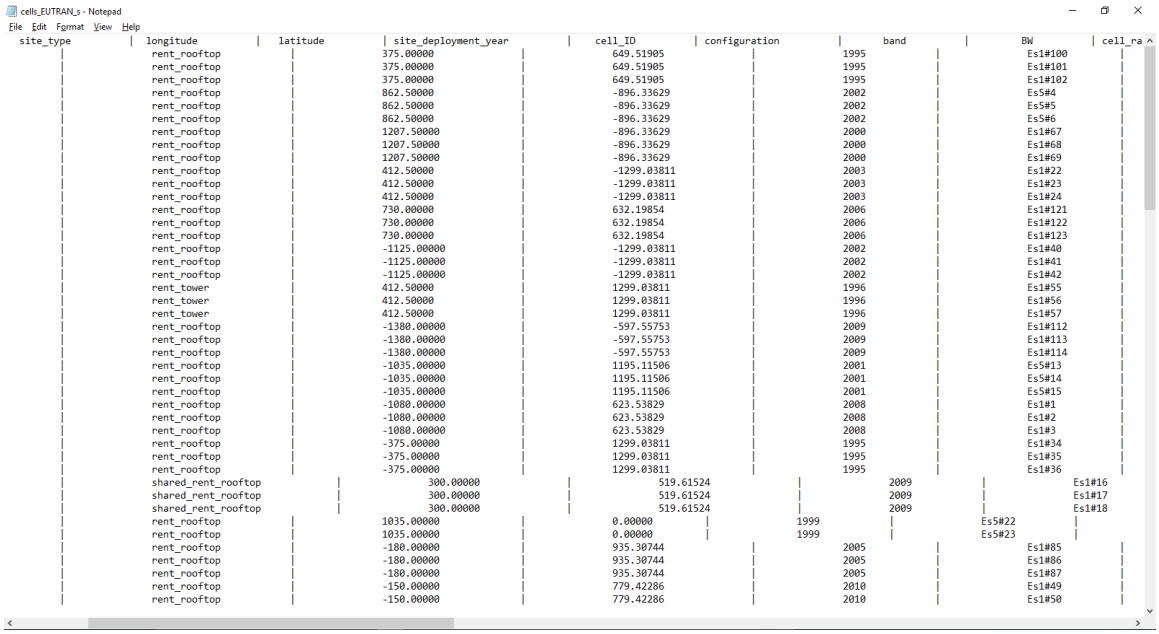
			band	double	0	
			BW	double	0	
			cell_range_km	double	0	
			cell_capacity_Mbps	double	0	
			cell_throughput_Mbps	double	0	
			development_year	double	0	
	GERAN			1 x 1 struct		
		sector900		Nested 1 x N struct		N = num of sectors
			cell_ID	char	’’	
			configuration	char	’’	
			band	double	0	
			BW	double	0	
			cell_range_km	double	0	
			cell_capacity_Mbps	double	0	
			cell_throughput_Mbps	double	0	
			development_year	double	0	
		sector1800		Nested 1 x N struct		N = num of sectors
			cell_ID	char	’’	
			configuration	char	’’	
			band	double	0	
			BW	double	0	
			cell_range_km	double	0	
			cell_capacity_Mbps	double	0	
			cell_throughput_Mbps	double	0	
			development_year	double	0	
	transmission			Nested 1 x 2 struct		
		owned_transmission_lines		double	0	
		leased_transmission_lines		double	0	
	configurations			char	’’	
	cell_directions			double	0	

Create the site_infrastructure

Algorithm which distribute the cells inputs_technology.(temp_RAN) to site_structure and clears the unused pre-allocated structs, fields etc
Using: function distribute_RAN_to_sites (inputs_technology, site_infrastructure, temp_RAN, temp_sector, temp_site_ID_prefix, temp_iter, temp_potential_sites)

Write the site infrastructure to file

Using function io_write_site_infrastructure_to_file (site_infrastructure)



cells_EUTRAN_s - Notepad

File Edit Format View Help

site_type	longitude	latitude	site_deployment_year	cell_ID	configuration	band	BW	cell_ra
rent_rooftop	375.00000	649.51905	1995	Es1#100	
rent_rooftop	375.00000	649.51905	1995	Es1#101	
rent_rooftop	375.00000	649.51905	1995	Es1#102	
rent_rooftop	862.50000	-896.33629	2002	Es5#4	
rent_rooftop	862.50000	-896.33629	2002	Es5#5	
rent_rooftop	862.50000	-896.33629	2002	Es5#6	
rent_rooftop	1207.50000	-896.33629	2000	Es1#67	
rent_rooftop	1207.50000	-896.33629	2000	Es1#68	
rent_rooftop	1207.50000	-896.33629	2000	Es1#69	
rent_rooftop	412.50000	-1299.03811	2003	Es1#22	
rent_rooftop	412.50000	-1299.03811	2003	Es1#23	
rent_rooftop	412.50000	-1299.03811	2003	Es1#24	
rent_rooftop	730.00000	632.19854	2006	Es1#121	
rent_rooftop	730.00000	632.19854	2006	Es1#122	
rent_rooftop	730.00000	632.19854	2006	Es1#123	
rent_rooftop	-1125.00000	-1299.03811	2002	Es1#40	
rent_rooftop	-1125.00000	-1299.03811	2002	Es1#41	
rent_rooftop	-1125.00000	-1299.03811	2002	Es1#42	
rent_tower	412.50000	1299.03811	1996	Es1#55	
rent_tower	412.50000	1299.03811	1996	Es1#56	
rent_tower	412.50000	1299.03811	1996	Es1#57	
rent_rooftop	-1380.00000	-597.55753	2009	Es1#112	
rent_rooftop	-1380.00000	-597.55753	2009	Es1#113	
rent_rooftop	-1380.00000	-597.55753	2009	Es1#114	
rent_rooftop	-1035.00000	1195.11506	2001	Es5#13	
rent_rooftop	-1035.00000	1195.11506	2001	Es5#14	
rent_rooftop	-1035.00000	1195.11506	2001	Es5#15	
rent_rooftop	-1080.00000	623.53829	2008	Es1#1	
rent_rooftop	-1080.00000	623.53829	2008	Es1#2	
rent_rooftop	-1080.00000	623.53829	2008	Es1#3	
rent_rooftop	-375.00000	1299.03811	1995	Es1#34	
rent_rooftop	-375.00000	1299.03811	1995	Es1#35	
rent_rooftop	-375.00000	1299.03811	1995	Es1#36	
shared_rent_rooftop	300.00000	519.61524	2009	Es1#16
shared_rent_rooftop	300.00000	519.61524	2009	Es1#17
shared_rent_rooftop	300.00000	519.61524	2009	Es1#18
rent_rooftop	1035.00000	0.00000	1999	Es5#22	
rent_rooftop	1035.00000	0.00000	1999	Es5#23	
rent_rooftop	-180.00000	935.30744	2005	Es1#85	
rent_rooftop	-180.00000	935.30744	2005	Es1#86	
rent_rooftop	-180.00000	935.30744	2005	Es1#87	
rent_rooftop	-150.00000	779.42286	2010	Es1#49	
rent_rooftop	-150.00000	779.42286	2010	Es1#50	

Per site:

number_of_site
site_ID
site_type
longitude
latitude
development_year

per cell:

cell_ID
configuration
band
BW
cell_range_km
cell_direction_az
cell_capacity_Mbps
cell_throughput_Mbps
development_year
transmission.owned_transmission_lines

B. By importing files with actual network information

Read the site infrastructure from file

Using function site_infarstructure = io_read_site_infrastructure_from_file (cells_EUTRAN_s, cells _EUTRAN_CA, cells _UTRAN, cells _GERAN)

Processing initial network infrastructure

SR9_network_information

Create network information struct and extract network information per technology using function:

network_information.(RAN) = extract_network_information()

RAN = EUTRAN_s, EUTRAN_CA, UTRAN_2100, UTRAN_900, GERAN_900 or GERAN _1800

network_information.(RAN).

Variable	*Data type*	*Comment*
configurations	cell(Nx1) of strings	N = num of RAN configurations
configuration_per_site	cell(1xN) of string	N = num of sites
bands_per_site	cell(1xN) of double	N = num of sites
BWs_per_site	cell(1xN) of double	N = num of sites
cells_per_site	double 1xN	N = num of sites
cells	double	sum of cells
basestations_per_site	double 1xN	N = num of sites
basestations	double	sum of cells
TRx_per_BTS	double	Only for GERAN and total
TRx	double	Only for GERAN and total
capacity_Mbps_per_cell	double MxN	M = default cell number N = num of sites
capacity_Mbps_per_site	double 1xN	N = num of sites
total_capacity_Mbps	double	sum of cells
avrg_cell_capacity_Mbps	double	
throughput_Mbps_per_cell	double MxN	M = default cell number N = num of sites
throughput_Mbps_per_site	double 1xN	N = num of sites
throughput_Mbps_in_BH_per_site	double 1xN	N = num of sites
max_safe_throughput_Mbps_per_cell	double MxN	M = default cell number N = num of sites
max_safe_throughput_Mbps_per_site	double 1xN	N = num of sites
max_safe_throughput_Mbps_in_BH_per_site	double 1xN	N = num of sites
max_techlimit_throughput_Mbps_per_cell	double MxN	M = default cell number N = num of sites
max_techlimit_throughput_Mbps_per_site	double 1xN	N = num of sites
max_techlimit_throughput_Mbps_in_BH_per_site	double 1xN	N = num of sites
total_throughput_Mbps	double	sum of cells
total_throughput_Mbps_in_BH	double	sum of cells
total_max_safe_throughput_Mbps	double	sum of cells
total_max_safe_throughput_Mbps_in_BH	double	sum of cells
total_max_techlimit_throughput_Mbps	double	sum of cells
total_max_techlimit_throughput_Mbps_in_BH	double	sum of cells
avrg_cell_throughput_Mbps	double	average

avrg_cell_throughput_Mbps_in_BH	double	average
avrg_cell_max_safe_throughput_Mbps	double	average
current_load_per_cell	double MxN	M = default cell number N = num of sites
current_load_per_site	double 1xN	N = num of sites
current_load_per_site_in_BH	double 1xN	N = num of sites
avrg_cell_load	double	average
avrg_cell_load_in_BH	double	average

Create network information struct for total network and extract network information using function:

network_information.all_RAN = extract_network_information_ all_RAN ()

network_information.all_RAN.

Variable	*Data type*	*Comment*
configurations	cell(Nx1) of strings	N = num of RAN configurations
configuration_per_site	cell(MxN) of string	M = num of technolopgy RANs N = num of sites
bands_per_site	cell(MxN) of double	M = num of technolopgy RANs N = num of sites
BWs_per_site	cell(MxN) of double	M = num of technolopgy RANs N = num of sites
cells_per_site	double 1xN	N = num of sites
cells	double	sum of cells
basestations_per_site	double 1xN	N = num of sites
basestations	double	sum of cells
TRx_per_BTS	double	N = num of sites
TRx	double	sum of cells
sites_owned_towers	double	new
sites_owned_rooftops	double	new
sites_rent_towers	double	new
sites_rent_rooftops	double	new
sites_shared_rent_towers	double	new
sites_shared_rent_rooftops	double	new
data_subs	double	new
avrg_site_data_subs	double	new
capacity_Mbps_per_site	double 1xN	N = num of sites
total_capacity_Mbps	double	sum of cells
avrg_cell_capacity_Mbps	double	
throughput_Mbps_per_site	double 1xN	N = num of sites
throughput_Mbps_in_BH_per_site	double 1xN	N = num of sites
max_safe_throughput_Mbps_per_site	double 1xN	N = num of sites
max_safe_throughput_Mbps_in_BH_per_site	double 1xN	N = num of sites
max_techlimit_throughput_Mbps_per_site	double 1xN	N = num of sites
max_techlimit_throughput_Mbps_in_BH_per_site	double 1xN	N = num of sites
total_throughput_Mbps	double	sum of cells
total_throughput_Mbps_in_BH	double	sum of cells
total_max_safe_throughput_Mbps	double	sum of cells

total_max_safe_throughput_Mbps_in_BH	double	sum of cells
total_max_techlimit_throughput_Mbps	double	sum of cells
total_max_techlimit_throughput_Mbps_in_BH	double	sum of cells
avrg_cell_throughput_Mbps	double	average
avrg_cell_throughput_Mbps_in_BH	double	average
avrg_cell_max_safe_throughput_Mbps	double	average
current_load_per_cell	double MxN	M = default cell number N = num of sites
current_load_per_site	double 1xN	N = num of sites
current_load_per_site_in_BH	double 1xN	N = num of sites
avrg_cell_load	double	average
avrg_cell_load_in_BH	double	average
mobile_data_traffic_from_zero_Mbps	double (1xX)	X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis

Costs calculation (Short-run)

SR10_assets_value

Create network cost struct per technology for fixed assets at the cell level (replacement cost, depreciation/amortization, book value) using function:

network_cost.(RAN).fixed_asset_cost_cell= fixed_assets_cost_cells()

RAN = EUTRAN_s, EUTRAN_CA, UTRAN_2100, UTRAN_900, GERAN_900 or GERAN _1800

network_cost(RAN).fixed_asset_cost_cell.

Variable	*Data type*	*Comment*
RAN	1 x N Struct	N = num of sites
site_development	1xN double	N = num of cells
site_equipment	1xN double	N = num of cells
basestations	1xN double	N = num of cells
controllers	1xN double	N = num of cells
owned_transmission_lines	1xN double	N = num of cells
other_tangible_fixed_asset	1xN double	N = num of cells
RANtoCORE	1 x N Struct	N = num of sites
owned_transmission_lines	1xN double	N = num of cells
other_tangible_fixed_asset	1xN double	N = num of cells
CORE	1 x N Struct	N = num of sites
PSnetwork_elements	1xN double	N = num of cells
common_network_elements	1xN double	N = num of cells
owned_transmission_lines	1xN double	N = num of cells
other_tangible_fixed_asset	1xN double	N = num of cells
non_network	1 x N Struct	N = num of sites
tangible_fixed_assets	1xN double	N = num of cells
intangible_fixed_assets	1xN double	N = num of cells
RAN_dep		
…		Same variables as RAN
RANtoCORE_dep		
…		Same variables as RANtoCORE
CORE_dep		
….		Same variables as CORE
non_network_dep		
…		Same variables as non_network
RAN_value		
…		Same variables as RAN
RANtoCORE_value		
…		Same variables as RANtoCORE

CORE_value		
…		Same variables as CORE
non_network_value		
…		Same variables as non_network
spectrum		
allocation		
spectrum_amort		
amortization		
spectrum_value		
value		

Create total network cost struct for fixed assets (replacement cost, depreciation/amortization, book value) using function:

network_cost.all_RAN.fixed_asset_cost= fixed_assets_cost_all_RAN ()

network_cost.all_RAN.fixed_asset_cost

Variable	*Data type*	*Comment*
fixed_asset_cost	Struct	
RAN_site_development	1xN double	N = num of sites
RAN_site_equipment	1xN double	N = num of sites
RAN_basestations	1xN double	N = num of sites
RAN_controllers	1xN double	N = num of sites
RAN_owned_transmission_lines	1xN double	N = num of sites
RAN_other_tangible_fixed_asset	1xN double	N = num of sites
RANtoCORE_owned_transmission_lines	1xN double	N = num of sites
RANtoCORE_other_tangible_fixed_asset	1xN double	N = num of sites
CORE_PSnetwork_elements	1xN double	N = num of sites
CORE_common_network_elements	1xN double	N = num of sites
CORE_owned_transmission_lines	1xN double	N = num of sites
CORE_other_tangible_fixed_asset	1xN double	N = num of sites
non_network_tangible_fixed_assets	1xN double	N = num of sites
non_network_intangible_fixed_assets	1xN double	N = num of sites
fixed_asset_depreciation	Struct	
	1xN double	Same variables as above
fixed_asset_value	Struct	
	1xN double	Same variables as above
spectrum_cost	Struct	
allocation	1xN double	N = num of sites
spectrum_amortization	Struct	
amortization	1xN double	N = num of sites
spectrum_value	Struct	
value	1xN double	N = num of sites

SR11_operating_costs

Create network cost struct per technology for operating costs at the cell level (cost of good sold and OPEX) using function:

[network_cost.(RAN).operating_cost_cell, network_energy. (RAN)] = operating_cost_cells()

RAN = EUTRAN_s, EUTRAN_CA, UTRAN_2100, UTRAN_900, GERAN_900 or GERAN _1800

Energy consumption is also calculated and extracted

network_cost(RAN). operating_cost_cell.

Variable	*Data type*	*Comment*
cost_of_good_sold	1xN struct	N = num of sites
RAN_site_rental	1xN double	N = num of cells
RAN_network_operation_maintenance	1xN double	N = num of cells
RAN_fixed_energy_consumption	1xN double	N = num of cells
RAN_current_variable_energy_consumption	1xN double	N = num of cells
RAN_variable_energy_consumption	1xN cell of 1xX doubles	N = num of cells X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis
RAN_leased_transmission_lines	1xN double	N = num of cells
RAN_personnel	1xN double	N = num of cells
RAN_other	1xN double	N = num of cells
RANtoCORE_leased_transmission_lines	1xN double	N = num of cells
RANtoCORE_network_operation_maintenance	1xN double	N = num of cells
RANtoCORE_other	1xN double	N = num of cells
CORE_fixed_energy_consumption	1xN double	N = num of cells
CORE_current_variable_energy_consumption	1xN double	N = num of cells
CORE_variable_energy_consumption	1xN cell of 1xX doubles	N = num of cells X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis
CORE_network_operation_maintenance	1xN double	N = num of cells
CORE_personnel	1xN double	N = num of cells
CORE_other	1xN double	N = num of cells
operating_expenses	1xN struct	N = num of sites
interconnection_and_other_fees	1xN double	N = num of cells
selling_general_and_administrative	1xN double	N = num of cells

network_energy (RAN).

Variable	*Data type*	*Comment*
energy_RAN	1xN double	N = num of sites
fixed_energy_consumption_MWh	1xN double	N = num of cells
variable_energy_consumption_MWh	1xN double	N = num of cells
current_variable_energy_consumption_MWh	1xN cell of 1xX doubles	N = num of cells X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis
energy_CORE	1xN double	N = num of sites
fixed_energy_consumption_MWh	1xN double	N = num of cells

	variable_energy_consumption_MWh	1xN double	N = num of cells
	current_variable_energy_consumption_MWh	1xN cell of 1xX doubles	N = num of cells X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis

Create total network cost struct for operating costs (cost of good sold and OPEX) using function:

[network_cost.all_RAN.operating_cost, network_energy. all_RAN] = operating_cost_all_RAN ()

Energy consumption is also calculated and extracted

network_cost.all_RAN.operating_cost.

Variable	*Data type*	*Comment*
cogs_RAN_site_rental	1xN double	N = num of sites
cogs_RAN_network_operation_maintenance	1xN double	N = num of sites
cogs_RAN_fixed_energy_consumption	1xN double	N = num of sites
cogs_RAN_current_variable_energy_consumption	1xN double	N = num of sites
cogs_RAN_variable_energy_consumption	1xN cell of 1xX doubles	N = num of sites X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis
cogs_RAN_leased_transmission_lines	1xN double	N = num of sites
cogs_RAN_personnel	1xN double	N = num of sites
cogs_RAN_other	1xN double	N = num of sites
cogs_RANtoCORE_leased_transmission_lines	1xN double	N = num of sites
cogs_RANtoCORE_network_operation_maintenance	1xN double	N = num of sites
cogs_RANtoCORE_other	1xN double	N = num of sites
cogs_CORE_fixed_energy_consumption	1xN double	N = num of sites
cogs_CORE_current_variable_energy_consumption	1xN double	N = num of sites
cogs_CORE_variable_energy_consumption	1xN cell of 1xX doubles	N = num of sites X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis
cogs_CORE_network_operation_maintenance	1xN double	N = num of sites
cogs_CORE_personnel	1xN double	N = num of sites
cogs_CORE_other	1xN double	N = num of sites
opex_interconnection_and_other_fees	1xN double	N = num of sites
opex_selling_general_and_administrative	1xN double	N = num of sites

network_energy.all_RAN.

Variable	*Data type*	*Comment*
energy_RAN	struct	
fixed_energy_consumption_MWh	1xN double	N = num of sites
variable_energy_consumption_MWh	1xN double	N = num of sites
current_variable_energy_consumption_MWh	1xN cell of 1xX doubles	N = num of sites X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis
energy_CORE	struct	
fixed_energy_consumption_MWh	1xN double	N = num of sites

	variable_energy_consumption_MWh	1xN double	N = num of sites
	current_variable_energy_consumption_MWh	1xN cell of 1xX doubles	N = num of sites X depends on the step (Mbps) from zero to the max traffic the network can satisfy *it is used for x-axis

Revenue modeling

D1_input

Variable	*Data type*	*Example*	*Comment*
inputs_demand_parameters	Struct		
market_share_mno	1xN double	0.38	N = num of historic years
future_market_share_mno	1xN double	0.38	N = num of future years
quantity	1xN double	90474752	N = num of historic years GB/month
determinants_of_demand	struct		
price	1xN double	0.76	N = num of historic years € GB/month
experienced_user_data_rate_factor	1xN double	1	N = num of historic years GB/month
future_experienced_user_data_rate_factor	1xN double	10	N = num of future years

D2_demand_estimation

A log-linear equation is found to represent the demand curve. All the coefficients are estimated

D3_data_traffic_forecast

Several traffic growth scenarios are studied (exponential, power, geometric)

D4_demand_forecast

The future price quantity equilibria are estimated

D5_revenues

The annual revenues are estimated

D6_marginal

The marginal revenue are estimated using function:
market_demand_revenues.inverse_demand.(year).(market).(scenario).actual =
calculate_inverse_demand()

year = year_0, year_1, year_2, year_3, or year_4,
market = total, or MNO_regional
scenario = low, medium, or high(traffic scenario)
market_demand_revenues.inverse_demand.(year).(market).(scenario).actual.

Variable	*Data type*	*Comment*
quantity_log_resolution	1xX double	X is the resolution of the quantity for better illustration. From zero to the snapshot traffic level (UL+DL)
price_log_resolution	1xX double	X is the resolution of the quantity for better illustration. From zero to the snapshot traffic level (UL+DL)
marginal_revenue_resolution	1xX double	X is the resolution of the quantity for better illustration. From zero to the snapshot traffic level (UL+DL)
quantity_log_resolution_long	1xX double	X is the resolution of the quantity for better illustration. From zero to traffic volume of the selected traffic scenario
price_log_resolution_long	1xXdouble	X is the resolution of the quantity for better illustration. From zero to traffic volume of the selected traffic scenario
marginal_revenue_resolution_long	1xX double	X is the resolution of the quantity for better illustration. From zero to traffic volume of the selected traffic scenario

Building the future network infrastructure

Initialization phase

LR1-LR4 inputs

LR1_inputs_market_industry
LR2_inputs_technology_spectrum
LR3_inputs_costs
LR4_inputs_finance

The future network evolution scenario is described by the variables in the following m files. The files are similar to the SR1-SR4, but their variables are future-oriented.

LR5_supply_cell

Next the target network supply (cells, capacity per technology) is estimated. The process is the same as in SR5 m file. The target network is the network infrastructure at the end of the investments.

LR6_demand_cell

Next, the demand evolution is calculated for each cell. An algorithm generates the cell throughput progression from the current traffic level to the maximum traffic that the future network can carry. The current algorithm simply calculates the progression based on the weighing factor of current traffic load in a cell compared to the other cells.

function network_throughput_progression.(RAN) = cell_throughput_progression_Mbps()

network_throughput_progression.

Variable	*Data type*	*Comment*
(RAN)	NxMxY	N = num of cells M = num of sites Y depends on the step (Mbps) from zero traffic growth to the max traffic growth the future network can satisfy
all_RAN.per_site	1xMxY	M = num of sites Y depends on the step (Mbps) from zero traffic growth to the max traffic growth the future network can satisfy
all_RAN. traffic_increase	1xY	Y depends on the step (Mbps) from zero traffic growth to the max traffic growth the future network can satisfy
all_RAN.growth	1xY	Y depends on the step (Mbps) from zero traffic growth to the max traffic growth the future network can satisfy

Build the future network infrastructure (snapshot for selected traffic growth)

LR7_sites_to_invest_due_to_traffic

1. Get traffic growth rate from the graphical user interface

2. Traffic growth rate and corresponding year based on traffic growth scenarios

Using function convert_trafficGrowth_to_time()

3. Identifying which cells need investments due to the traffic growth

LR_network_information_growth_snapshot.

Variable	*Data type*	*Comment*
traffic_growth_snapshot_index	double	The index of the total network traffic progression
traffic_growth_snapshot_time	1xN double	In years, N = Num of traffic growth scenarios
(RAN)	1xY	Y depends on the step (Mbps) from zero traffic growth to the max traffic growth the future network can satisfy
inv_trigger_per_cell	NxM logical	N = num of cells M = num of sites 1s show the cell which has capacity problem and requires investments
sites_to_invest	1xN double	Shows the site number (unique) N = num of sites which require investments
site_cell_to_invest	Nx2	N = num of investment cells Column 1 shows the site Column 2 shows the cell which require investment in the site
all_RAN	1xY	Y depends on the step (Mbps) from zero traffic growth to the max traffic growth the future network can satisfy
growth_per_site	1xN logical	M = num of sites
sites_to_invest	1xN double	Shows the site (unique) N = num of sites which require investments

LR8_investments_scenario

Creating the LR_site_infrastructure based on the current infrastructure, the future network scenario and the traffic growth

1. Find the most critical sites:
    a. All the potential sites to invest according to the scenario
    b. Compare with the sites which need investment for the specified traffic growth. Group sites:
        i. investment_sites = need investment and belong to potential sites to invest
        ii. no_investment_sites_affected_by_investments = they are inside the range of investment_sites and the investmne site can carry their traffic
        iii. no_investment_sites = scenario cannot solve their traffic problem

2. Copy existing infrastructure to the future one (site_infrastructure = LR_site_infrastructure)
3. Network changes based on the scenario and the traffic growth
    a. Install equipment, using function: LR_site_infrastructure = invest_to_sites()
    b. Uninstall equipment
4. Changes on throughputs after investments, considering the followings:
    a. traffic growth using function LR_site_infrastructure = cell_throughput_LR_Mbps()
    b. the investment_sites carry the total traffic of uninstalled technologies, using function LR_site_infrastructure = cell_throughput_correction_LR_Mbps()
    c. the investment_sites carry the exceeded traffic of collocated technologies, , using function LR_site_infrastructure = cell_throughput_correction_LR_Mbps()
    d. the investment_sites carry the exceeded traffic of no_investment_sites_affected_by_investments, using function LR_site_infrastructure = cell_throughput_correction_LR_Mbps()
5. The new territory coverage per technology is calculated using function LR_inputs_technology.LR.(RAN)_territory_coverage_new = network_territory_coverage()

Processing future network infrastructure snapshot

LR9_network_information

Similar to SR9_network_information

Cost calculation (snapshot for selected traffic growth rate)

LR10_investments_cost

Create LR network cost struct per technology for the invest cost and disposal value at the cell level using functions:

LR_network_cost.(RAN).investment_cost_cell = investment_cost_cell()
LR_network_cost.(RAN).disposal_value_cell = disposal_value_cell()

RAN = EUTRAN_s, EUTRAN_CA, UTRAN_2100, UTRAN_900, GERAN_900 or GERAN _1800

LR_network_cost.(RAN).investment_cost_cell.
LR_network_cost.(RAN). disposal_value_cell.

Variable	*Data type*	*Comment*
RAN	1 x N Struct	N = num of sites
site_development	1xN double	N = num of cells
site_equipment	1xN double	N = num of cells
basestations	1xN double	N = num of cells
controllers	1xN double	N = num of cells
owned_transmission_lines	1xN double	N = num of cells
other_tangible_fixed_asset	1xN double	N = num of cells
RANtoCORE	1 x N Struct	N = num of sites
owned_transmission_lines	1xN double	N = num of cells
other_tangible_fixed_asset	1xN double	N = num of cells
CORE	1 x N Struct	N = num of sites
PSnetwork_elements	1xN double	N = num of cells
common_network_elements	1xN double	N = num of cells
owned_transmission_lines	1xN double	N = num of cells
other_tangible_fixed_asset	1xN double	N = num of cells
non_network	1 x N Struct	N = num of sites
tangible_fixed_assets	1xN double	N = num of cells
intangible_fixed_assets	1xN double	N = num of cells

Create LR total network cost struct for the invest cost and disposal value per site using functions:

LR_network_cost.all_RAN.investment_cost = investment_cost_all_RAN()
LR_network_cost.all_RAN.disposal_value = disposal_value_all_RAN()

LR_network_cost.all_RAN.investment_cost .

LR_network_cost.all_RAN.disposal_value.

Variable	*Data type*	*Comment*
RAN_site_development	1xN double	N = num of sites
RAN_site_equipment	1xN double	N = num of sites
RAN_basestations	1xN double	N = num of sites
RAN_controllers	1xN double	N = num of sites
RAN_owned_transmission_lines	1xN double	N = num of sites
RAN_other_tangible_fixed_asset	1xN double	N = num of sites
RANtoCORE_owned_transmission_lines	1xN double	N = num of sites
RANtoCORE_other_tangible_fixed_asset	1xN double	N = num of sites
CORE_PSnetwork_elements	1xN double	N = num of sites
CORE_common_network_elements	1xN double	N = num of sites
CORE_owned_transmission_lines	1xN double	N = num of sites
CORE_other_tangible_fixed_asset	1xN double	N = num of sites
non_network_tangible_fixed_assets	1xN double	N = num of sites
non_network_intangible_fixed_assets	1xN double	N = num of sites

LR11_assets_value

Similar to SR_11

LR12_operating_costs

Similar to SR_12

LR_D_marginal

Estimate the price of GB per month and the marginal revenue for the specific year and traffic volume as specified from the selected traffic growth rate.

Using function:

market_demand_revenues.inverse_demand.(year).(market).(scenario).actual = calculate_inverse_demand()