Mathematical Formulation and Explanation of the MGET GAMS Code

Version 0.2.0

Contents

1 Nomenclature

1.1 Sets

Symbol	Description
Y	Set of years in the planning horizon
H	Set of operational hours
E	Set of energy carriers (e.g., Gas, Hydrogen, Electricity)
N	Set of nodes (geographical locations)
A	Set of arcs (pipeline connections between nodes)
$A_e(n)$	Set of arcs entering node n, representing gas inflows.
$A_s(n)$	Set of arcs leaving node n, representing gas outflows.
F	Set of fuels (e.g., Natural Gas, Hydrogen)
Z	Set of demand shortage variables
$ypred(y2, y)$	Set of past years affecting bidirectionality

1.2 Indices

Symbol	Description
y	Index for years ($y \in Y$)
h	Index for hours ($h \in H$)
e	Index for energy carriers ($e \in E$)
n	Index for nodes ($n \in N$)
a	Index for arcs ($a \in A$)
f	Index for fuels ($f \in F$)
e	Index for the original energy carrier before repurposing
f	Index for the new energy carrier after repurposing
z	Index for demand shortages ($z \in Z$)

1.3 Parameters

Symbol	Description
$c_p(n, e, y)$	Cost per unit of gas production at node n for energy carrier e in year y
$c_a(a, e, y)$	Cost of transporting one unit of energy carrier e through arc a in year y.
$c_{ab}(a, e, y)$	Cost per unit of bidirectional arc capacity for energy carrier e on arc a in year y
$c_{ax}(a, e, y)$	Cost per unit of expanding capacity for arc a, energy carrier e in year y

$c_z(z, e)$	Cost per unit of unmet demand for energy carrier e in shortage zone z
$c_z(\text{ZD2}, e)$	Cost per unit of unmet demand for energy carrier e at the NUTS2 regional level
$c_{we}(n, e)$	Cost per unit of gas extracted from storage at node n for energy carrier e
$c_{lr}(n, e)$	Cost per unit of regasification for node n, energy carrier e
$c_{bl}(e, f)$	Cost per unit of blending energy carrier e into fuel f.
$cap_{wi}(n, e, y)$	Maximum gas injection capacity at node n for energy carrier e in year y.
$cap_{we}(n, e, y)$	Maximum gas extraction capacity at node n for energy carrier e in year y.
$cap_p(n, e, y, h)$	Maximum production capacity at node n for energy carrier e in year y, hour h
$dmd(n, e, y, h)$	NUTS3 level gas demand for energy carrier e at node n in year y and hour h.
$dmd2(nuts2, e, y, h)$	NUTS2 level gas demand for energy carrier e in region $nuts2$ in year y and hour h.
$e_a(a, e)$	Efficiency factor for gas transportation along arc a for energy carrier e.
$e_w(n, e)$	Efficiency factor regulating storage operations at node n for energy carrier e
$EOH(y)$	Equivalent operational hours in year y
$f_{ar}(a, e, f, y)$	Fixed cost for repurposing arc a from fuels e to f in year y
$f_{ab}(a, y)$	Fixed cost of converting arc a to bidirectional operation in year y
$lb_p(n, e, y, h)$	Minimum production requirement at node n for energy carrier e in year y, hour h
$ub_{bl}(f, e)$	Upper bound fraction for blending energy carrier e into fuel f
$r(y)$	Discount rate for year y
$scaleUp(h)$	Hourly scaling factor that adjusts gas production to reflect variations in demand, operational intensity, or peak production periods.
$vola2(e)$	Volumetric adjustment factor for energy carrier e
$bigM$	Large constant for enforcing binary constraints
$is\_bid(a)$	Indicator if arc a is already bidirectional
$yscai(y2, y)$	Scaling factor linking previous and current years*

1.4 Decision Variables

1.4.1 Binary Decision Variables

Symbol	**Description**
$B_{BD}(a, y)$	Indicator for decision to make arc a bidirectional in year y
$B_{AR}(a, e, f, y)$	Indicator for repurposing arc a from energy carrier e to f in year y
$B_{WR}(n, e, f, y)$	Indicator for repurposing storage at node n from carrier e to f in year y

1.4.2 Continuous Decision Variables

Symbol	**Description**
TC	Total cost (objective function)

$BD(a, y)$	Indicator whether arc a is bidirectional in year y
$F_A(a, e, y, h)$	Quantity of gas transported through arc a for energy carrier e in year y and hour h.
$K_A(a, e, y)$	Capacity of arc a for energy carrier e in year y
$K_W(n, e, y)$	Working gas capacity for energy carrier e at node n in year y
$K_{BD}(a, e, y)$	Capacity of bidirectional flow on arc a for energy carrier e in year y
$K_{OPP}(a, e, y)$	Reversed arc capacity for a in year y
$K_{RA}(a, e, f, y)$	Repurposed arc capacity from carrier e to f in year y
$K_{RW}(n, e, f, y)$	Repurposed storage capacity at node n from carrier e to f in year y
$Q_P(n, e, y, h)$	Amount of gas production at node n for energy carrier e
$Q_E(n, e, y, h)$	Amount of gas extracted from storage at node n for energy carrier e in year y and hour h
$Q_I(n, e, y, h)$	Amount of gas injected to storage at node n for energy carrier e in year y and hour h
$Q_R(n, e, y, h)$	Amount of LNG regasified at node n in year y and hour h
$Q_B(n, e, f, y, h)$	Amount of energy carrier e blended into fuel f at node n in year y and hour h.
$Q_S(n, e, y, h)$	Demand covered (sales) of energy carrier e at node n in year y and hour h.
$X_A(a, e, y)$	Capacity expansion for arc a and energy carrier e in year y
$ZDS(z, n, e, y, h)$	Unmet gas demand at node n in shortage zone z for energy carrier e in year y and hour h
$ZN2(nuts2, e, y, h)$	Unmet gas demand in region $nuts2$ for energy carrier e in year y and hour h.

2 Scenario Definition and Data Initialization

The first part of the code defines global parameters and loads the scenario configuration from a separate setup file. This modular structure supports flexible scenario management without modifying the main model script.

2.1 Configuration File Loading

GAMS Code (`MGET_v0_2_0.gms`)

```
$if not setglobal CASE_CONFIG $setglobal CASE_CONFIG "cases/Spain_case/case_config.gms"
$include %CASE_CONFIG%
$include load_input_from_Excel.gms
```

Explanation: The main script `MGET_v0_2_0.gms` begins by checking whether a `CASE_CONFIG` path has already been set. If not, it defaults to `cases/Spain_case/case_config.gms`, which contains all case-specific parameters. Including this external configuration ensures a clean separation between scenario settings and model logic.

2.2 Scenario and Global Parameters (Defined in `case_config.gms`)

GAMS Code (`case_config.gms`)

```
$setglobal KeepGdx    ''
$setglobal Verify     '*'
$setglobal data       Spain
$setglobal path       cases/%data%_case
$setglobal hor        2040
$setglobal oper       4
$setglobal scen       0
$setglobal string     %data%_%hor%_%oper%_%scen%
$setglobal excel_file "%path%/input_data/%string%.xlsx"
$include load_input_from_Excel_v0_2_0.gms
```

Scenario Settings: This script sets parameters for the selected case:

- `data`: Case name (e.g., Spain)
- `hor`: Planning horizon (final year)
- `oper`: Operational representative hours
- `scen`: Scenario index (e.g., 0 = moderate, 2 = ambitious)
- `string`: Scenario identifier used in output filenames
- `excel_file`: Full path to the input Excel file

Modular Setup: By moving scenario definitions into a dedicated file, users can easily modify scenarios without changing the main model. This approach improves maintainability, clarity, and adaptability to new use cases.

Case-Specific Settings: These global variables define the dataset, planning horizon, and scenario assumptions used in the model. The scenario ID is dynamically created using the dataset name, last year, operational time step, and decarbonization scenario.

Decarbonization and Hydrogen Scenarios: The variable scen is set to 2, indicating an ambitious decarbonization and hydrogen uptake scenario, while alternative values allow for moderate policy scenarios.

2.3 Data Input from External Files:

GAMS Code

```
$INCLUDE load_input_from_Excel_v0_2_0.gms
```

The external script $load\_input\_from\_Excel\_v0\_2\_0.gms$ is included to import required data for the model. This script extracts parameters such as production capacities, demand profiles, and infrastructure constraints.

2.4 Definition of Decision Variables

GAMS Code

```
Binary Variable
    B_BD(a,y)         Indicator for decision to make arc bidirectional
    B_AR(a,e,f,y)     Indicator for arc repurposing decision
    B_WR(n,e,f,y)     Indicator for storage repurposing decision
;
Free Variable
    TC                 Model objective total costs
;
Positive Variable
    BD(a,y)            Indicator that Arc is bidirectional
    F_A(a,e,y,h)       Flow - hourly
    K_A(a,e,y)         Arc capacity
    K_W(n,e,y)         Working gas capacity
    K_BD(a,e,y)        Volume for volume-dependent cost component for making an arc bidirectional
    K_OPP(a,e,y)       Reversed Arc capacity
    K_RA(a,e,f,y)      Repurposed Arc capacity
    K_RW(n,e,f,y)      Repurposed Working gas capacity
    Q_B(n,e,f,y,h)     Energy carrier e blended into f
    Q_E(n,e,y,h)       Storage extraction - hourly
    Q_I(n,e,y,h)       Storage injection - hourly
    Q_P(n,e,y,h)       Production - hourly
    Q_R(n,e,y,h)       Regasification - hourly
    Q_S(n,e,y,h)       Demand covered ('sales') - hourly
```

```
X_A(a,e,y)         Arc expansion
ZDS(z,n,e,y,h)     Nodal deficits and surpluses
ZN2                NUTS2 level demand shortage
```

For clarity, the nomenclature of all sets, indices, parameters, and variables used in the model is provided in Section 1.

3 Input Verification

During the data loading process, an additional script called `verify_nodes_arcs.gms` is included to perform consistency checks on the input network structure.

GAMS Code Snippet (from `load_input_from_Excel_v0_2_0.gms`)

```
$include Verify_nodes_arcs.gms
```

Purpose of Verification: This verification script automatically checks whether:

- All nodes are connected to at least one arc (i.e., no isolated nodes)
- Each arc has valid start and end nodes
- Nodes or arcs with missing or inconsistent data are flagged

Error Handling: If errors are detected (e.g., a node has no connection or an arc lacks capacity), the script assigns them to diagnostic sets such as `abort_n` (problematic nodes) or `abort_a` (problematic arcs), and the model will halt with a clear error message. This helps the user identify data issues early.

Benefit: Including this step ensures robustness and data integrity before model execution, reducing the likelihood of infeasible or misleading optimization results.

4 Mathematical Formulation

4.1 Objective Function

The objective function minimizes the total cost with the following GAMS script:

GAMS Code:

```
obj.. TC =E=  sum((a,e,y),        r(y)*EOH(y)*c_ax(a,e,y)*    X_A(a,e,y))+
              sum((a,e,f,y),      r(y)*EOH(y)*f_ar(a,e,f,y)* B_AR(a,e,f,y))+
              sum((a,e,f,y),      r(y)*EOH(y)*c_ar(a,e,f,y)* K_RA(a,e,f,y))+
              sum((a,y),          r(y)*EOH(y)*f_ab(a,y)*      B_BD(a,y))+
              sum((a,e,y),        r(y)*EOH(y)*c_ab(a,e,y)*    K_BD(a,e,y))+
              sum((n,e,y,h),      r(y)*EOH(y)*c_p(n,e,y)*     Q_P(n,e,y,h)      *scaleUp(h))+
              sum((a,e,y,h),      r(y)*EOH(y)*c_a(a,e,y)*     F_A(a,e,y,h)      *scaleUp(h))+
              sum((z,n,e,y,h),    r(y)*EOH(y)*c_z(z,e)*       ZDS(z,n,e,y,h)    *scaleUp(h))+
              sum((nuts2,e,y,h),  r(y)*EOH(y)*c_z('ZD2',e)*   ZN2(nuts2,e,y,h)  *scaleUp(h))+
              sum((n,e,y,h),      r(y)*EOH(y)*c_we(n,e)*      Q_E(n,e,y,h)      *scaleUp(h))+
              sum((n,e,y,h),      r(y)*EOH(y)*c_lr(n,e)*      Q_R(n,e,y,h)      *scaleUp(h))+
              sum((n,e,f,y,h),    r(y)*EOH(y)*c_bl(e,f)*      Q_B(n,e,f,y,h)    *scaleUp(h))
;
```

Mathematical Representation

$$
\begin{aligned}
TC = &\sum_{(a,e,y)} r(y) \cdot EOH(y) \cdot c_{ax}(a,e,y) \cdot X_A(a,e,y) && \text{(1a)}\\
&+ \sum_{(a,e,f,y)} r(y) \cdot EOH(y) \cdot f_{ar}(a,e,f,y) \cdot B_{AR}(a,e,f,y) && \text{(1b)}\\
&+ \sum_{(a,e,f,y)} r(y) \cdot EOH(y) \cdot c_{ar}(a,e,f,y) \cdot K_{RA}(a,e,f,y) && \text{(1c)}\\
&+ \sum_{(a,y)} r(y) \cdot EOH(y) \cdot f_{ab}(a,y) \cdot B_{BD}(a,y) && \text{(1d)}\\
&+ \sum_{(a,e,y)} r(y) \cdot EOH(y) \cdot c_{ab}(a,e,y) \cdot K_{BD}(a,e,y) && \text{(1e)}\\
&+ \sum_{(n,e,y,h)} r(y) \cdot EOH(y) \cdot c_p(n,e,y) \cdot Q_P(n,e,y,h) \cdot scaleUp(h) && \text{(1f)}\\
&+ \sum_{(a,e,y,h)} r(y) \cdot EOH(y) \cdot c_a(a,e,y) \cdot F_A(a,e,y,h) \cdot scaleUp(h) && \text{(1g)}\\
&+ \sum_{(z,n,e,y,h)} r(y) \cdot EOH(y) \cdot c_z(z,e) \cdot ZDS(z,n,e,y,h) \cdot scaleUp(h) && \text{(1h)}\\
&+ \sum_{(\text{nuts2},e,y,h)} r(y) \cdot EOH(y) \cdot c_z('ZD2',e) \cdot ZN2(\text{nuts2},e,y,h) \cdot scaleUp(h) && \text{(1i)}\\
&+ \sum_{(n,e,y,h)} r(y) \cdot EOH(y) \cdot c_{we}(n,e) \cdot Q_E(n,e,y,h) \cdot scaleUp(h) && \text{(1j)}\\
&+ \sum_{(n,e,y,h)} r(y) \cdot EOH(y) \cdot c_{lr}(n,e) \cdot Q_R(n,e,y,h) \cdot scaleUp(h) && \text{(1k)}\\
&+ \sum_{(n,e,f,y,h)} r(y) \cdot EOH(y) \cdot c_{bl}(e,f) \cdot Q_B(n,e,f,y,h) \cdot scaleUp(h) && \text{(1l)}
\end{aligned}
$$

Equation set (1) defines the total cost function, where individual terms Equations (1a) to (1l) represent different cost components.

4.1.1 Arc Expansion Cost

$$\sum_{(a,e,y)} r(y) \cdot EOH(y) \cdot c_{ax}(a,e,y) \cdot X_A(a,e,y) \tag{1a}$$

Equation (1a) captures the total expansion cost by summing over all arcs a, energy carriers (types of gas) e, and years y. The term $r(y)$ represents the discount factor applied to each year to account for the time value of money, ensuring that future costs are appropriately discounted to their present value. The equivalent operating hours $EOH(y)$ in year y scale the cost according to the annual utilization of the infrastructure.

The unit expansion cost, denoted as $c_{ax}(a, e, y)$, represents the cost per unit of capacity expansion for arc a and energy carrier e in year y. This parameter reflects investment costs related to pipeline construction, material, labor, and associated infrastructure adjustments required to accommodate increased capacity. Finally, $X_A(a, e, y)$ is the decision variable representing the amount of additional capacity installed for arc a and energy carrier e in year y. The optimization model determines the optimal values of $X_A(a, e, y)$ to minimize total system costs while ensuring that demand and operational constraints are met.

By incorporating these parameters, the expansion cost function ensures an economically efficient allocation of resources, considering both present and future infrastructure development costs.

4.1.2 Arc Repurposing Fixed Cost

$$\sum_{(a,e,f,y)} r(y) \cdot EOH(y) \cdot f_{ar}(a, e, f, y) \cdot B_{AR}(a, e, f, y) \tag{1b}$$

The arc repurposing fixed cost represents the expense incurred when modifying an existing pipeline to transport a different energy carrier, such as hydrogen or another gas. This cost is included in the model to ensure that infrastructure changes required for energy transition are considered.

Equation (1b) calculates the total repurposing cost by summing over all arcs a, energy carriers e, new energy carriers f, and years y. The term $r(y)$ is the discount rate applied to future costs, ensuring that expenses in later years are appropriately adjusted. The factor $EOH(y)$ represents the equivalent operational hours in year y, helping to scale costs based on system usage.

Term $f_{ar}(a, e, f, y)$ defines the **fixed cost** of repurposing arc a from energy carrier e to f in year y. This cost includes all necessary modifications to make the pipeline suitable for transporting the new energy carrier.

The binary variable $B_{AR}(a, e, f, y)$ determines whether repurposing takes place. If the arc is repurposed, $B_{AR}(a, e, f, y) = 1$, and the corresponding cost is included in the total. If no repurposing occurs, $B_{AR}(a, e, f, y) = 0$, and the cost is excluded.

4.1.3 Arc Repurposing Variable Cost

The variable cost associated with repurposing an arc accounts for the expenses incurred based on the capacity that is repurposed. Unlike the fixed repurposing cost, which is independent of the arc's capacity, the variable cost scales with the volume of infrastructure being modified. This cost component is mathematically represented as:

$$\sum_{(a,e,f,y)} r(y) \times EOH(y) \times c_{ar}(a,e,f,y) \times K_{RA}(a,e,f,y) \tag{1c}$$

In this equation, $c_{ar}(a,e,f,y)$ represents per unit cost of repurposing per unit capacity for arc a from energy carrier e to f in year y. The term $K_{RA}(a,e,f,y)$ denotes the repurposed arc capacity, capturing the extent of the infrastructure conversion.

This cost function ensures that the economic feasibility of repurposing is evaluated in proportion to the infrastructure modifications required. Higher repurposing capacities increase costs, influencing the optimization of pipeline infrastructure for alternative gas energy.

4.1.4 Bidirectional Conversion Fixed Cost

The bidirectional conversion fixed cost accounts for the investment required to enable bidirectional flow in a pipeline. The mathematical representation of this cost is given by:

$$\sum_{(a,y)} r(y) \times EOH(y) \times f_{ab}(a,y) \times B_{BD}(a,y) \tag{1d}$$

where $f_{ab}(a,y)$ denotes the fixed cost of modifying arc a in year y to support bidirectional operation. This includes technical upgrades such as valve installations, pressure regulation modifications, and additional operational mechanisms. $B_{BD}(a,y)$ is a binary decision variable that takes the value 1 if arc a is converted to bidirectional flow in year y, and 0 otherwise.

4.1.5 Bidirectional Conversion Variable Cost

The variable cost of bidirectional conversion accounts for the expenses incurred based on the capacity assigned to bidirectional flow. This cost component is mathematically formulated as:

$$\sum_{(a,e,y)} r(y) \times EOH(y) \times c_{ab}(a,e,y) \times K_{BD}(a,e,y) \tag{1e}$$

where $c_{ab}(a,e,y)$ denotes the cost per unit of bidirectional arc capacity for energy carrier (gas types) e on arc a in year y. This parameter accounts for additional operational and maintenance costs associated with bidirectional flow. $K_{BD}(a, e, y)$ is the assigned capacity for bidirectional flow on arc a for energy carrier e in year y.

4.1.6 Gas Production Cost

The gas production cost component accounts for the expenses associated with producing gas at different nodes over time. The mathematical formulation is given as:

$$\sum_{(n,e,y,h)} r(y) \times EOH(y) \times c_p(n, e, y) \times Q_P(n, e, y, h) \times scaleUp(h) \tag{1f}$$

In this equation, $c_p(n, e, y)$ represents the cost per unit of gas production at node n for energy carrier e in year y. This parameter captures variations in production costs due to infrastructure, energy source, and economic conditions.

The variable $Q_P(n, e, y, h)$ denotes the gas production level at node n for energy carrier e in year y and hour h. This reflects the volumetric flow of gas being produced and contributes to the overall production expenditure. The scaling factor $scaleUp(h)$ adjusts for different hourly production rates, ensuring the equation properly accounts for variations in operational intensity.

4.1.7 Pipeline Flow Cost

The cost associated with transporting gas through the pipeline network is represented by the pipeline flow cost component. This cost depends on the volume of gas transported, the specific pipeline arc used, and the operational conditions over time. Mathematically, it is formulated as follows:

$$\sum_{(a,e,y,h)} r(y) \times EOH(y) \times c_a(a, e, y) \times F_A(a, e, y, h) \times scaleUp(h) \tag{1g}$$

The cost of transporting one unit of energy carrier e through arc a in year y is represented by $c_a(a, e, y)$. The quantity of gas transported through arc a for energy carrier e in year y and hour h is denoted by $F_A(a, e, y, h)$.

4.1.8 Demand Deficit Cost

The demand deficit cost accounts for the economic penalty associated with unmet gas demand. In cases where the available supply of an energy carrier is insufficient to meet demand, a cost is incurred to reflect the consequences of the shortfall. The mathematical representation of this cost component is given by:

$$\sum_{(z,n,e,y,h)} r(y) \times EOH(y) \times c_z(z,e) \times ZDS(z,n,e,y,h) \times scaleUp(h) \tag{1h}$$

In this equation, $c_z(z,e)$ represents the cost per unit of unmet demand for energy carrier e in shortage zone z. The variable $ZDS(z,n,e,y,h)$ quantifies the shortfall in gas supply at node n for energy carrier e in year y and hour h.

4.1.9 NUTS2 Demand Shortage Penalty

The NUTS2 demand shortage penalty represents the economic cost associated with regional gas supply shortages at the NUTS2 level. This component is included in the model to further incentivize sufficient gas supply across broader geographical regions, ensuring that regional imbalances are accounted for in the optimization framework.

$$\sum_{(nuts2,e,y,h)} r(y) \times EOH(y) \times c_z(\mathrm{ZD2},e) \times ZN2(nuts2,e,y,h) \times scaleUp(h) \tag{1i}$$

In this equation, $c_z(\mathrm{ZD2},e)$ represents the cost per unit of unmet demand for energy carrier e at the NUTS2 level, denoted by ZD2. This cost reflects the financial consequences of regional supply shortfalls. The variable $ZN2(nuts2,e,y,h)$ quantifies the extent of unmet demand at the NUTS2 regional level for energy carrier e in year y and hour h. This demand shortfall is aggregated at a higher geographical resolution compared to individual nodes, allowing for a broader assessment of energy security. Note that the resolution for hydrogen demand and production is NUTS2 level.

4.1.10 Storage Extraction Cost

The storage extraction cost accounts for the operational expenses associated with withdrawing gas from storage facilities at different nodes. This cost is influenced by the extraction volume and the predefined unit cost for each node and energy carrier. It is mathematically represented as:

$$\sum_{(n,e,y,h)} r(y) \times EOH(y) \times c_{we}(n,e) \times Q_E(n,e,y,h) \times scaleUp(h) \tag{1j}$$

where $c_{we}(n,e)$ is the cost per unit of gas extracted from storage at node n for energy carrier e.

$Q_E(n, e, y, h)$ represents the volume of gas extracted from storage at node n for energy carrier e in year y and hour h.

4.1.11 Regasification Cost

Regasification refers to the process of converting liquefied natural gas (LNG) back into its gaseous state before it can be injected into the pipeline network. The total regasification cost is given by:

$$\sum_{(n,e,y,h)} r(y) \times EOH(y) \times c_{lr}(n, e) \times Q_R(n, e, y, h) \times scaleUp(h) \tag{1k}$$

where $c_{lr}(n, e)$ denotes the cost per unit of regasification at node n for energy carrier e. $Q_R(n, e, y, h)$ is the volume of energy carrier e regasified at node n in year y and hour h.

4.1.12 Blending Cost

Blending refers to the process of mixing different energy carriers, such as hydrogen with natural gas. The total blending cost is expressed as:

$$\sum_{(n,e,f,y,h)} r(y) \times EOH(y) \times c_{bl}(e, f) \times Q_B(n, e, f, y, h) \times scaleUp(h) \tag{1l}$$

where $c_{bl}(e, f)$ is the cost per unit of blending energy carrier e into f, incorporating infrastructure adjustments and operational expenses. $Q_B(n, e, f, y, h)$ represents the quantity of energy carrier e blended into f at node n in year y and hour h.

4.2 Gas Production Capacity Constraint

GAMS code:

```
Q_P.fx(n,e,y,h)$(cap_p(n,e,y,h)<=0)=0;
p_cap(n,e,y,h) $(cap_p(n,e,y,h)>0)  ..
   Q_P(n,e,y,h)  + sum(f,Q_B(n,e,f,y,h))  =L= cap_p(n,e,y,h);
p_min(n,e,y,h)$(lb_p(n,e,y,h)>0)   ..
   Q_P(n,e,y,h) + sum(f,Q_B(n,e,f,y,h)) =G= lb_p(n,e,y,h);
```

Mathematical Representation

$$Q_P(n,e,y,h) = 0, \qquad \text{if } cap_p(n,e,y,h) \le 0 \tag{3}$$

$$Q_P(n,e,y,h) + \sum_f Q_B(n,e,f,y,h) \le cap_p(n,e,y,h), \qquad \text{if } cap_p(n,e,y,h) > 0 \tag{4}$$

$$Q_P(n,e,y,h) + \sum_f Q_B(n,e,f,y,h) \ge lb_p(n,e,y,h), \qquad \text{if } lb_p(n,e,y,h) > 0 \tag{5}$$

The gas production constraints regulate the amount of gas that can be produced and blended at each node.

Zero production pondition in Equation (3) ensures that if the available production capacity $cap_p(n,e,y,h)$ is zero or negative, the corresponding gas production $Q_P(n,e,y,h)$ must also be zero. This prevents infeasible production when capacity is unavailable.

Production upper bound in Equation (4) enforces a constraint ensuring that the total gas production $Q_P(n,e,y,h)$ and blended gas $Q_B(n,e,f,y,h)$ at node n do not exceed the maximum available production capacity $cap_p(n,e,y,h)$. This condition maintains operational feasibility by ensuring that production does not surpass the infrastructure's designed capacity.

Production lower bound in Equation (5) guarantees that gas production meets a minimum threshold $lb_p(n,e,y,h)$, preventing production from dropping below a required level where applicable. This ensures a minimum level of supply to meet operational and contractual obligations.

4.3 Demand Satisfaction Constraints

GAMS Code:

```
Q_S.fx(n,e,y,h)$(not_h(e) AND dmd(n,e,y,h) <= 0) = 0;
ZDS.fx('ZD2',n,e,y,h)$(not_h(e) AND dmd(n,e,y,h) <= 0) = 0;
dmd_n3(n,e,y,h)$(not_h(e) AND dmd(n,e,y,h) > 0)..
    Q_S(n,e,y,h) =E= dmd(n,e,y,h) - ZDS('ZD2',n,e,y,h);
dmd_n2(nuts2,e,y,h)$(is_h(e) AND dmd2(nuts2,e,y,h))..
    sum(n$n_in_2(n,nuts2), Q_S(n,e,y,h)) =E= dmd2(nuts2,e,y,h) - ZN2(nuts2,e,y,h);
```

Mathematical Formulation:

$$Q_S(n,e,y,h) = 0, \qquad \text{if } dmd(n,e,y,h) \leq 0 \tag{6}$$

$$ZDS('ZD2',n,e,y,h) = 0, \qquad \text{if } dmd(n,e,y,h) \leq 0 \tag{7}$$

$$Q_S(n,e,y,h) = dmd(n,e,y,h) - ZDS('ZD2',n,e,y,h), \qquad \text{if } dmd(n,e,y,h) > 0 \tag{8}$$

$$\sum_{n\in N} Q_S(n,e,y,h) = dmd2(nuts2,e,y,h) - ZN2(nuts2,e,y,h) \tag{9}$$

The demand satisfaction constraints regulate the supply of gas to meet demand at both the NUTS3 (nodal) and NUTS2 (regional) levels.

Equation (6) ensures that if the demand at node n is zero or negative for energy carrier e in year y and hour h, then the satisfied demand $Q_S(n,e,y,h)$ is also set to zero. Equation (7) follows a similar logic, ensuring that the demand shortage variable $ZDS('ZD2',n,e,y,h)$ is zero when there is no positive demand. Equation (8) enforces that the amount of gas supplied at node n equals the demand minus any shortages represented by $ZDS('ZD2',n,e,y,h)$. Equation (9) extends the constraint to the regional (NUTS2) level, ensuring that the total satisfied demand across all nodes in region $nuts2$ matches the total demand $dmd2(nuts2,e,y,h)$, minus any regional shortages $ZN2(nuts2,e,y,h)$.

4.4 Mass Balance Constraints

The mass balance equation ensures that at each node n, the total incoming energy equals the total outgoing energy, maintaining network stability in gas transportation.

GAMS Code:

```
mb(n,e,y,h)..
    Q_P(n,e,y,h) + sum(a$a_e(a,n),F_A(a,e,y,h)*e_a(a,e)) + Q_E(n,e,y,h)
    + Q_R(n,e,y,h) + sum(f,Q_B(n,f,e,y,h)) =E=
    Q_S(n,e,y,h) + sum(a$a_s(a,n),F_A(a,e,y,h)) + Q_I(n,e,y,h);
```

Mathematical Formulation:

$$\begin{aligned} Q_P(n,e,y,h) &+ \sum_{a\in A_e(n)} F_A(a,e,y,h)\cdot e_a(a,e) + Q_E(n,e,y,h) + Q_R(n,e,y,h) \\ &+ \sum_{f} Q_B(n,f,e,y,h) = Q_S(n,e,y,h) + \sum_{a\in A_s(n)} F_A(a,e,y,h) + Q_I(n,e,y,h) \end{aligned} \tag{10}$$

This mass balance constraint (10) ensures that all incoming energy at node n is equal to the outgoing energy, preserving the conservation of mass within the gas network: The left-hand side represents the total energy supply at node n, which consists of: gas production (Q_P), pipeline inflows (F_A), adjusted by an efficiency factor $e_a(a,e)$, gas extraction from storage (Q_E), regasification of LNG (Q_R), blended gas contributions (Q_B).

The right-hand side represents the total energy demand and outflows, including final demand fulfillment (Q_S), pipeline outflows (F_A), gas injection into storage (Q_I). This constraint ensures that the gas network remains balanced.

4.5 Blending Constraints

GAMS Code:

```
Q_B.fx(n,e,f,y,h)$(not_h(e) OR not_g(f)) = 0;
Q_B.fx(n,e,f,y,h)$(cap_p(n,e,y,h) <= 0) = 0;
Q_B.fx(n,e,e,y,h) = 0;

max_bl(n,f,e,y,h)$(is_h(f) AND is_g(e) AND cap_p(n,f,y,h) > 0) ..
    Q_B(n,f,e,y,h)  =L= ub_bl(f,e) * (Q_S(n,e,y,h) + sum(a$a_s(a,n), F_A(a,e,y,h)) + Q_I(n,e,y,h));
```

Mathematical Formulation:

$$Q_B(n,e,f,y,h) = \begin{cases} 0, & \text{if } e \notin H \text{ or } f \notin G \\ 0, & \text{if } cap_p(n,e,y,h) \leq 0 \\ 0, & \text{if } e = f \end{cases} \tag{11}$$

$$Q_B(n,f,e,y,h) \leq ub_{bl}(f,e) \times \left(Q_S(n,e,y,h) + \sum_{a \in A_s(n)} F_A(a,e,y,h) + Q_I(n,e,y,h) \right) \tag{12}$$

The blending constraints regulate the integration of hydrogen into the natural gas network, ensuring that the blending process adheres to physical and operational limits. Equation (11) restricts blending to cases where the input energy carrier e is hydrogen (H) and the output fuel f is gas (G). Additionally, it ensures that blending does not occur if production capacity is zero or negative and prevents self-blending where an energy carrier is blended with itself. Equation (12) imposes an upper bound on blending, ensuring that the blended quantity does not exceed a fraction $ub_{bl}(f,e)$ of the total available gas, which includes: gas sales (demand) $Q_S(n,e,y,h)$, pipeline inflows $\sum_{a \in A_s(n)} F_A(a,e,y,h)$, storage injection $Q_I(n,e,y,h)$.

These constraints guarantee that hydrogen blending is technically feasible and does not disrupt the integrity of the natural gas network while ensuring compliance with operational and infrastructure limits.

4.6 Bidirectionality Constraints

These constraints regulate the capacity and operational feasibility of bidirectional flows within the pipeline network. They define the upper limits of pipeline flow, control bidirectional operation, and ensure capacity consistency over time.

GAMS Code:

```
a_lim(a,e,y,h)..
    F_A(a,e,y,h) * vola2(e) =L= K_A(a,e,y) + K_OPP(a,e,y);
a_opp1(a,e,y)..
    K_OPP(a,e,y) =L= BD(a,y) * bigM;
a_opp2(a,e,y)..
    K_OPP(a,e,y) =L= sum(ao$opp(ao,a),K_A(ao,e,y));
bd_cost(a,e,y,y2)$(yscai(y2,y) AND NOT is_bid(a))..
   K_BD(a,e,y)  =G= K_OPP(a,e,y2) - (1-B_BD(a,y)) * bigM;
BD.up(a,y) = 1;
BD.fx(a,y)$is_bid(a) = 1;
bidir(a,y)$(NOT is_bid(a))..
    BD(a,y) =L= B_BD(a,y) + sum(y2$ypred(y2,y), BD(a,y2));
```

Mathematical Formulation:

$$F_A(a,e,y,h) \times vola2(e) \leq K_A(a,e,y) + K_{OPP}(a,e,y) \tag{13}$$

$$K_{OPP}(a,e,y) \leq BD(a,y) \times M \tag{14}$$

$$K_{OPP}(a,e,y) \leq \sum_{\substack{ao \in A \\ opp(ao,a)}} K_A(ao,e,y) \tag{15}$$

$$K_{BD}(a,e,y) \geq K_{OPP}(a,e,y2) - (1 - B_{BD}(a,y)) \times M \tag{16}$$

$$BD(a,y) = \begin{cases} 1, & \text{if } is\_bid(a) = 1 \\ B_{BD}(a,y) + \sum_{y2 \in ypred(y2,y)} BD(a,y2), & \text{otherwise} \end{cases} \tag{17}$$

The bidirectionality constraints define the rules for assigning bidirectional flow capacity to pipelines. Equation (13) ensures that total gas flow along arc a does not exceed the sum of forward and reverse capacities $K_A(a,e,y)$ and $K_{OPP}(a,e,y)$. Equation (14) enforces that reverse flow capacity $K_{OPP}(a,e,y)$ is only assigned if the arc is bidirectional ($BD(a,y) = 1$). Equation (15) ensures that reverse flow does not exceed the available capacity in the opposite direction. Equation (16) ensures that bidirectional capacity expansion depends on past decisions, using a large-M formulation to control feasibility. Equation (17) maintains time consistency, ensuring that once an arc becomes bidirectional, it stays bidirectional. Equation (17) also fixes bidirectional arcs that are predefined.

4.7 Repurposing Constraints

These constraints regulate the repurposing of infrastructure, ensuring that: The total repurposed capacity is equal to the previous period's capacity. Only explicitly allowed repurposing decisions are assigned capacity.

GAMS Code:

```
* First period capacity is given by input data
K_A.fx(a,e,y)$(ORD(y)=1) = cap_a(a,e,y);
K_W.fx(n,e,y)$(ORD(y)=1) = cap_ww(n,e,y);

* Capacity is all capacity (re)purposed to the current carrier and
* whatever is invested specific for the current carrier
ar_cap(a,e,y)$(ord(y)>1)..
    K_A(a,e,y) =E= sum(f, K_RA(a,f,e,y)) + sum(y2$ypred(y2,y), X_A(a,e,y2));
wr_cap(n,e,y)$(ord(y)>1)..
    K_W(n,e,y) =E= sum(f, K_RW(n,f,e,y));

* Only one (re)purposing destination
sos_a(a,e,y)$(ord(y)>1)..
    sum(f,B_AR(a,e,f,y)) =E= 1;
sos_w(n,e,y)$(ord(y)>1)..
    sum(f,B_WR(n,e,f,y)) =E= 1;
```

```
* No repurposing in the first period
B_AR.fx(a,e,e,y)$(ORD(y)=1) = 0;
K_RA.fx(a,e,f,y)$(ORD(y)=1) = 0;
B_WR.fx(n,e,e,y)$(ORD(y)=1) = 0;
K_RW.fx(n,e,f,y)$(ORD(y)=1) = 0;

* Sum of repurposing capacities away from the specific
*carriers equals the previous period carrier capacity
bil_a1(a,e,y)$(ord(y)>1)..
    sum(f, K_RA(a,e,f,y)) =E= sum(y2$ypred(y2,y), K_A(a,e,y2));
bil_w1(n,e,y)$(ORD(y)>1)..
    sum(f, K_RW(n,e,f,y)) =E= sum(y2$ypred(y2,y), K_W(n,e,y2));

* Only the specific repurposing can get the capacity
bil_a2(a,e,f,y)$(ord(y)>1)..
    K_RA(a,e,f,y) =L= B_AR(a,e,f,y) * bigM;
bil_w2(n,e,f,y)$(ord(y)>1)..
    K_RW(n,e,f,y) =L= B_WR(n,e,f,y) * bigM;
```

Mathematical Formulation

$$K_A(a,e,y) = \sum_f K_{RA}(a,f,e,y) + \sum_{y2 \in ypred(y2,y)} X_A(a,e,y2) \tag{18}$$

$$K_W(n,e,y) = \sum_f K_{RW}(n,f,e,y) \tag{19}$$

$$\sum_f B_{AR}(a,e,f,y) = 1 \tag{20}$$

$$\sum_f B_{WR}(n,e,f,y) = 1 \tag{21}$$

$$K_{RA}(a,e,f,y) = 0, \quad \text{if } y = 1 \tag{22}$$

$$K_{RW}(n,e,f,y) = 0, \quad \text{if } y = 1 \tag{23}$$

$$\sum_f K_{RA}(a,e,f,y) = \sum_{y2 \in ypred(y2,y)} K_A(a,e,y2) \tag{24}$$

$$\sum_f K_{RW}(n,e,f,y) = \sum_{y2 \in ypred(y2,y)} K_W(n,e,y2) \tag{25}$$

$$K_{RA}(a,e,f,y) \leq B_{AR}(a,e,f,y) \times bigM \tag{26}$$

$$K_{RW}(n,e,f,y) \leq B_{WR}(n,e,f,y) \times bigM \tag{27}$$

The repurposing constraints ensure that storage and pipeline infrastructure can be repurposed while maintaining logical consistency across time. Equation (18) ensures that the total arc capacity in year y is equal to the sum of repurposed arc capacity and the previous investments made in the network. Equation (19) applies the same logic to storage capacity, ensuring that repurposed storage capacity is correctly accounted for in each period. Equation (20) enforces that only one repurposing decision is made per arc, meaning that a pipeline can only be repurposed into one other energy carrier. Equation (21) enforces the same restriction at the storage level, ensuring that each storage facility is repurposed for only one alternative energy carrier. Equations (22) and (23) ensure no repurposing occurs in the first period. This means that all initial pipeline and storage capacities remain unchanged at the start of the planning horizon. Equation (24) ensures that the total repurposed arc capacity in the current period equals the arc capacity from the previous period, preventing over-repurposing and maintaining capacity consistency. Equation (25) applies the same constraint to storage capacity, ensuring that repurposed storage remains within previously available capacity limits. Equation (26) ensures that repurposed arc capacity is only assigned if an explicit repurposing decision has been made, enforced using a large-M constraint to prevent infeasible repurposing. Equation (27) applies the same logic to storage facilities, ensuring that repurposed storage capacity is only assigned when explicitly decided.

4.8 Capacity Constraints for Gas Operations

These constraints impose upper limits on regasification, gas injection, and extraction capacities in the model. They ensure that gas operations do not exceed predefined infrastructure limits.

GAMS Code

```
* Regasifier capacity limit
Q_R.up(n,e,y,h) = dat_r(n,e,'2025','ub');
* Injection and extraction capacity limit; currently not adjustable.
Q_I.up(n,e,y,h) = cap_wi(n,e,y);
Q_E.up(n,e,y,h) = cap_we(n,e,y);
```

Mathematical Formulation:

$$Q_R^{up}(n,e,y,h) = dat_r(n,e,'2025','ub') \tag{28}$$

$$Q_I^{up}(n,e,y,h) = cap_{wi}(n,e,y) \tag{29}$$

$$Q_E^{up}(n,e,y,h) = cap_{we}(n,e,y) \tag{30}$$

Regasification Capacity Limit: Equation (28) enforces an upper bound on the regasification capacity at node n for energy carrier e in year y and hour h. The parameter $dat_r(n,e,'2025','ub')$ provides a predefined regasification capacity limit extracted from the dataset for the year 2025. This constraint ensures that regasification at any node does not exceed its designated upper limit.

Injection Capacity Limit: Equation (29) imposes an upper bound on gas injection capacity at node n for energy carrier e in year y and hour h. The parameter $cap_{wi}(n,e,y)$ specifies the predefined gas injection capacity limit at node n. This constraint ensures that gas injection into storage does not exceed its predefined capacity.

Extraction Capacity Limit: Equation (30) establishes an upper bound on gas extraction capacity at node n for energy carrier e in year y and hour h. The parameter $cap_{we}(n,e,y)$ defines the predefined gas extraction capacity limit at node n. This constraint ensures that gas withdrawal from storage does not exceed predefined capacity limits.

4.9 Storage Constraints

These constraints regulate the extraction and injection of gas into storage, ensuring that storage operations remain within capacity limits and maintain operational balance.

GAMS Code

```
* Storage Capacity Limitation
w_lim(n,e,y)..
    sum(h,scaleUp(h)*Q_E(n,e,y,h))*vols2(e) =L= K_W(n,e,y);

* Storage Cycle Balance Constraint
w_cyc(n,e,y)..
    sum(h,scaleUp(h)*Q_E(n,e,y,h))
    =E= e_w(n,e)*sum(h,scaleUp(h)*Q_I(n,e,y,h));
```

Mathematical Formulation

$$\sum_{h} \text{scaleUp}(h) \cdot Q_E(n,e,y,h) \cdot vols2(e) \leq K_W(n,e,y) \tag{31}$$

$$\sum_{h} \text{scaleUp}(h) \cdot Q_E(n,e,y,h) = e_w(n,e) \times \sum_{h} \text{scaleUp}(h) \cdot Q_I(n,e,y,h) \tag{32}$$

Storage Capacity Limitation: Equation (31) ensures that total gas extraction from storage does not exceed the available storage capacity (working capacity) $K_W(n,e,y)$ at node n for energy type e in year y. The term $vols2(e)$ adjusts for volume constraints, and scaleUp(h) accounts for operational scaling.

Storage Cycle Balance: Equation (32) maintains a balance between gas extraction and injection into storage. The efficiency factor $e_w(n,e)$ ensures that the relationship between injected and extracted gas respects operational efficiency constraints. This equation prevents excessive withdrawal without corresponding replenishment.

5 Model Execution and Result Reporting

This section describes the execution process, solver settings, and result reporting for the MGET model. It outlines the computational setup, optimization execution, and result storage.

5.1 Computation Time and Solver Selection

GAMS Code:

```
option reslim=7200;
option mip=cplex, limrow=1E3, limcol=1E3;
```

Solver Time Limit: This constraint sets the time limit for solving the optimization problem to 7200 seconds (2 hours), ensuring computational feasibility within reasonable execution time.

Solver Option: CPLEX is selected as the solver for the mixed-integer programming (MIP) problem, ensuring efficient computation of the optimization model.

Solution Display Limits: The settings limit the number of rows and columns displayed in the solution report to 1000 each, preventing excessive output that could affect readability.

5.2 Model Definition and Optimization Execution

GAMS Code:

```
model MGET /all/;
solve MGET min TC using MIP;
```

Model Definition: The MGET optimization model includes all constraints and decision variables, ensuring a comprehensive system representation.

Objective Function Minimization: The model solves for the minimum total cost (TC) using MIP approach, identifying the most cost-efficient energy transition strategy.

5.3 Reporting Scripts

GAMS Code:

```
$INCLUDE report.gms
```

External Reporting Script: The *report.gms* script is included to generate and format solution outputs, ensuring structured post-processing of results.

5.4 Exporting Results to GDX File

GAMS Code:

```
execute_unload 'gdx/%string%_var',
    BD, F_A, K_A, K_OPP, K_BD, K_RA, Q_B, Q_P, Q_S, X_A,
    ZDS, B_AR, B_BD, TC;
```

Exporting Decision Variables: The solution variables are stored in a GDX file, allowing further analysis and visualization.

5.5 Feasibility Check and Deficit Reporting

GAMS Code:

```
parameter flag;
flag('Infeasible?') = sum((z,n,e,y,h), ZDS.l(z,n,e,y,h));
display 'If flag is positive, check reports for surpluses and deficits', flag;
```

Feasibility Flag: This check detects whether the model has a low demand by adding the deficit variable ZDS between all nodes and time periods.

Interpreting the Flag:

- If flag $= 0$, the model solution is feasible.
- If flag > 0, the model has unmet demand, requiring a review of constraints and supply parameters.

Automated Deficit Reporting: If a deficit is detected, the user is asked to check the reports to investigate possible energy imbalances in the system.