

# Formation toolbox pour le design et controle des micro-reseaux

Gestion des scénarios

Corentin Boënnec

Groupe Genesys, Laplace, Université de Toulouse

17 mars 2025

# Qu'appelle-t-on scénario ?

## Définition

Un scénario est une structure de données qui recueille l'ensemble de variables d'environnement. Les éléments numériques qui ne sont ni des paramètres de notre modélisation du micro-réseau, ni des décisions de dimensionnement ou de contrôle.

C'est grâce aux scénarios que l'on peut modifier le contexte spatio-temporel dans lequel évolue un micro-réseau. Par conséquent, le design et l'opération d'un même micro-réseau varient d'un scénario à l'autre.

## Les variables d'environnement (sous forme de séries temporelles)

- ▶ Au pas d'opération  $\Delta h$  :
  - ▶ Des prix d'achat/vente sur le réseau ;
  - ▶ Des données de consommation ;
  - ▶ Des données météorologiques ;
  - ▶ ...
- ▶ Au pas d'investissement  $\Delta y$  :
  - ▶ Des prix pour les composants ;
  - ▶ ...

# Le rôle des scénarios

## Rôles

Les scénarios occupent deux rôles complémentaires :

1. Inscrire le réseau dans un contexte spatio-temporel. (inhérent à chaque scénario) ;
2. Augmenter la robustesse des décisions prises. (amené par la multiplication des scénarios)

Considérons les variables d'environnement comme des variables aléatoires qui seront révélés à l'avenir.

- ▶ Chaque scénario est alors une réalisation combinée de toutes les variables d'environnement. La distribution de ces variables est conditionnée au contexte spatio-temporel ;
- ▶ La multiplication des réalisations tend à mieux rendre compte des distributions associées aux variables et ainsi augmente les chances d'avoir couvert le déroulement des événements futurs.

# Situer le réseaux dans l'espace et le temps

## Bonne pratiques

De façon générale il est recommandé d'utiliser des données dont le contexte spatio-temporelle est proche de la situation étudiée. Néanmoins des compromis sont parfois nécessaires.

## Données météorologiques

La localisation doit être la même ou la plus proche possible.

## Données de consommation

Le profil des consommateurs est également important (nombre de personnes dans les logements, surface des logement, quels appareils sont alimentés par la source considérée...)

## Coût

Pour ce qui est des données relatives au prix ils dépendent également du lieu considéré.

# Amener de la robustesse dans les décisions

## Multi-scénario

La multiplication des scénarios tend à rendre compte des différentes réalisations possibles des variables aléatoires.

Bien entendu cela ne garanti pas avec certitudes la couverture de toutes les éventualités.  
D'autant plus pour des phénomène exceptionnels absents des données passées recueillis.

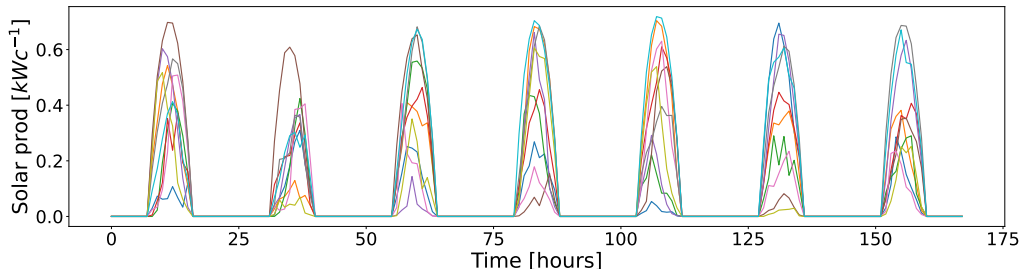


Figure – Variabilité sur 10 années de production solaire

# Prise en compte de l'incertitude

## Problème d'optimisation associé

La prise en compte de la multiplication des scénarios a pour conséquence de créer un problème plus grand. Il s'agit de trouver le dimensionnement qui répond le mieux à l'ensemble des scénarios — avec un dimensionnement unique et l'opération du système pour chaque scénario.

Le nombre de variables de décision relatives au dimensionnement ne change pas, mais le nombre de variables de décision relatives à l'opération est multiplié par le nombre de scénarios.

## Impact sur la résolution

Dans le meilleur des cas (complexité linéaire), cela revient à multiplier le temps de calcul par le nombre de scénarios (éventuellement beaucoup plus pour certaines classes de problème en programmation mathématique). Afin d'être en mesure de trouver ce meilleur dimensionnement, il convient de définir des métriques pour évaluer une multitude de résultats.

## Parallélisation

La simulation d'un même réseau sur plusieurs scénarios est particulièrement propice à la parallélisation. En effet les scénarios étant des univers différents, aucune interaction entre eux n'est prévu, seul leur évaluation est mise en commun. Leur déroulement peut alors aisément se faire dans des fils d'exécutions distincts et ainsi réduire le temps de calcul.

# L'approche par espérance

## Pondération par la probabilité d'occurrence (formelle)

Pour un ensemble  $\omega = \{\omega_1, \dots, \omega_{ns}\} \in \Omega^N$  de ns scénario dont la probabilité de réalisation est donné par le vecteur  $\{P(\omega_1), \dots, P(\omega_{ns})\}$ . Si l'on considère que le coût associé à un scénario  $\omega_i$  est donné par la fonction  $Cost: \Omega \rightarrow \mathbb{R}$ , alors on peut écrire le coût associé à l'ensemble des scénarios comme :

$$Cost(\omega) = \sum_{i=1}^{ns} Cost(\omega_i) \times P(\omega_i) \quad (1)$$

## Pondération par la probabilité d'occurrence (informelle)

L'approche pondérée où la probabilité de réalisation d'un scénario indique son poids dans le calcul d'une somme pondérée. Dans le cas où l'on ne connaîtrait pas ces probabilités, une répartition uniforme s'impose.

## Note :

Cette approche convient généralement bien aux objectifs.

# L'approche par mesure de risque

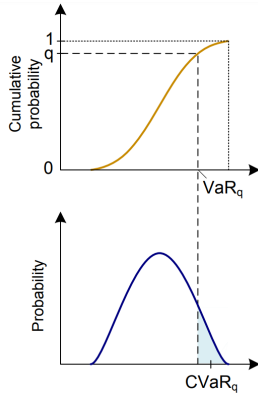


Figure – Illustration de la relation VaR, CVaR pour une variable aléatoire [1]

## Conditional Value at Risk (CVaR)

Pour une valeur  $q \in [0, 1]$ , On définit d'abord Value-at-Risk  $VaR_q$  comme la limite du  $q$ -quantile de la fonction de probabilité. la  $CVaR_q$  est alors calculée comme l'espérance des valeurs supérieures à  $VaR_q$ . Prenons  $q = 50\%$  et considérons une contrainte de Renewable Energy Share à 80%.

$$CVaR_q = \mathbb{E}[RES \mid RES \geq VaR_q] \quad (2)$$

et on exprime finalement la contrainte comme :  $CVaR_q \geq 80\%$   
 $CVaR_0$  sera équivalent à l'espérance.  $CVaR_1$  à contrario correspond au pire cas. Un avantage certain est que la CVaR est linéarisable [2, 3] elle permet donc d'adresser des contraintes de façon probabiliste en ajustant la rigidité de la contrainte. (cela peut permettre d'éviter le surdimensionnement du au pire cas).

## Note :

Cette approche convient généralement bien aux contraintes.



# Collecte de donnée

## Quelles données associé à chaque composant ?

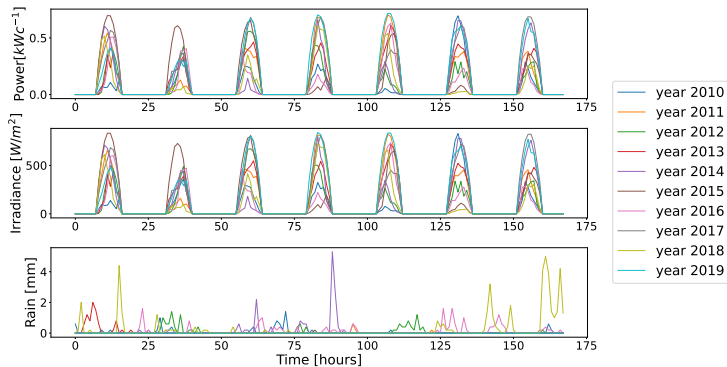
Pour construire les scénarios de données, il faut collecter des séries temporelles relatives à l'ensemble des éléments du réseau. Lorsque c'est possible, on souhaite récolter le plus possible d'occurrence pour les données dont le caractère est aléatoire. (par exemple plusieurs profils de consommation de plusieurs foyers sur plusieurs années).

- ▶ Demands : Pour chaque type de consommation, il faut des séries temporelles ( $\Delta h$ ) pour la consommation associant TimeStamp et puissances.
- ▶ generations : Pour chaque élément de production, il faut des séries temporelles permettant le calcul de la puissance générée associé à instant ( $\Delta h$ ) TimeStamp ainsi que le coût d'investissement associé au composant ( $\Delta y$ ).
- ▶ storages : Pour chaque élément de stockage, il convient de recueillir des informations permettant le calcul du coût d'investissement ( $\Delta y$ ).
- ▶ converters : Pour chaque élément de conversion énergétique, il convient de recueillir des informations permettant le calcul du coût d'investissement ( $\Delta y$ ).
- ▶ grids : Pour chaque réseau, il faut des séries temporelles ( $\Delta h$ ) associant TimeStamp et prix d'achat par unité de même que prix de vente. Enfin, selon les cas, on peut prévoir des informations relatives aux coûts d'abonnement et de dépassement ( $\Delta y$ ).

## Quelques exemples

- ▶ Demands : Charge en électricité, des données horaires de consommation en kW.
- ▶ generations : Panneau Photovoltaïque, le prix en €/kWc par année, ainsi que des données de production par unité installé au pas horaire.
- ▶ storages : Batterie Lithium-ion, le prix par année des batteries par kWh installé.
- ▶ converters : Electrolyzer, le prix par année des cellules par surface installée.
- ▶ grids : Connexion au réseau EDF, Tarif par heure en achat/vente et coût d'abonnement et de dépassement.

# Cohérence et corrélation des données



**Figure** – Illustration de la corrélation entre irradiation, production PV et pluviométrie. Données sur Toulouse de 2010 à 2019, première semaine de l'année

## corrélation

Certaines données présentent des corrélations fortes et d'autres des plus faibles. Globalement, il convient de les conserver au maximum.

Par exemple, l'irradiation et la production de PV sont très fortement corrélées. La pluie et l'ensoleillement aussi (avec une corrélation négative).

Un contexte spatio-temporel commun permet d'accomplir cette cohérence.

# Multiplication des données

## Le but

La problématique est donc que l'on cherche des séries de données, cohérentes entre elles et on souhaite en avoir le plus grand nombre. Ceci étant généralement difficile, des méthodes existent pour chercher à démultiplier ces données. De façon générale, plus les données initiales réelles sont nombreuses plus les méthodes seront efficaces.

## Génération de profils synthétiques par chaînes de Markov

La méthode utilisée dans le package consiste en :

1. Un processus de clusterisation afin de créer des états représentatifs des jour de semaine/weekend, pour chaque heure de chaque période (mois) ;
2. La détermination de probabilités de transition entre les états en suivant les liens des séries originelles entre les clusters ;
3. Une marche aléatoire à travers ces états afin de synthétiser des nouveaux scénarios évoluant dans l'intervalle de la distribution décrite par les scénarios réels fournis comme entrée.

## Mise en oeuvre

Plus d'information sont disponibles dans la thèse de Hugo Radet [4] ainsi qu'un article dédié à cette question [5] mais aussi dans le fichier `src/scenario/generation.jl` et le fichier `Examples/build_scenarios/main_stochastic.jl`

# Introduction aux questions relatives à l'incertitude

## Les différents types d'incertitudes

Il existe plusieurs types d'incertitude que l'on peut ranger dans 2 catégories principales :

1. Les incertitudes aléatoires : qui émane de la variabilité ou stochasticité (par exemple la météo, la consommation des ménages ou encore le prix des éléments du système).
2. Les incertitudes épistémiques qui proviennent d'un manque de connaissances (par exemple, l'incapacité intrinsèque des modèles à représenter la réalité.).

## Le rôle des scénarios

- ▶ Les scénarios peuvent être un outil pour aborder la problématique de l'incertitude aléatoire. En effet, bien que l'on puisse, en utilisant les données passées correspondant à la situation que l'on cherche à étudier, générer des séries temporelles crédibles pour représenter la situation. Ne connaissant pas le futur, ces séries temporelles ne contiendront qu'un fragment des situations que le futur réserve.
- ▶ En revanche, ils ne permettent pas ou peu d'adresser l'incertitude épistémique.

# La banque de scénario

## Pré-requis et contenu de la banque

Pour construire un scénario on a besoin de données que l'on stock dans des fichiers au format **.jld2**, un format qui permet le stockage de données en Julia.

Ces fichiers contiennent un **Dict** dont les clés (actuelles) et le contenu sont mentionné ci-dessous. Le lien entre une clé (en gras) et sa valeur est pointé par l'opérateur **=>** et le format des données exprimées entre crochets (nh = 8760 et ns le nombre de scénarios de 1 année contenu dans la base). La collecte de ces données doit être réalisée comme énoncé précédemment.

- ▶ ***p<sub>v</sub>*** => Dict(***t*** => [nh,1,ns], ***power*** => [nh,1,ns], ***cost*** => [1,ns])
- ▶ ***grid\_Elec*** => Dict(***cost\_in*** => [nh,1,ns], ***cost\_out*** => [nh,1,ns], ***cost\_exceed*** => [1,ns]),
- ▶ ***ld\_H*** => Dict(***t*** => [nh,1,ns], ***power*** => [nh,1,ns])
- ▶ ***ld\_E*** => Dict(***t*** => [nh,1,ns], ***power*** => [nh,1,ns])

- ▶ ***elyz*** => Dict(***cost*** => [1,ns])
- ▶ ***liion*** => Dict(***cost*** => [1,ns])
- ▶ ***fc*** => Dict(***cost*** => [1,ns])
- ▶ ***tes*** => Dict(***cost*** => [1,ns])
- ▶ ***heater*** => Dict(***cost*** => [1,ns])
- ▶ ***h2tank*** => Dict(***cost*** => [1,ns])

## Évolution

Cette liste n'est pas figée et peut être prolongée pour y inclure des données relatives à un autre composant. Ou encore d'autre données météorologique (pluie, vent...). L'intégration de nouvelles données ne perturbera pas le fonctionnement actuel. En revanche, des fonctionnalités devront être développées pour faire usage de ces nouvelles entrées. Par conséquent, il est tout à fait possible d'ajouter des éléments à une banque existante sans en perturber le fonctionnement.

# La structure Scenarios

## Structure

Concrètement la structure `Scenarios` est déclarée comme suit :

```
mutable struct Scenarios <: AbstractScenarios
  demands::Vector{NamedTuple}
  generations::Vector{NamedTuple}
  storages::Vector{NamedTuple}
  converters::Vector{NamedTuple}
  grids::Vector{NamedTuple}
end
```

## Contenu

Elle fonctionne plus ou moins de la même façon que la banque de données présentée précédemment mais va être construite sur la base d'un microgrid en particulier, en prenant en compte à la fois son architecture mais aussi ses horizons temporels et la multiplicité des séries temporelles (nh, ny, ns).

Une différence notable est que les champs seront triés selon l'élément du réseau concerné et qu'au lieu de dictionnaire, on retrouvera des `NamedTuple` dont la structure est figée.

# Le constructeur de Scenarios

## Le Constructeur

Le structure va être construite pour un microgrid `mg` en prenant `mg.parameters.ny` × `mg.parameters.ns` années aléatoires parmi la banque de données `data`.

```
w = Scenarios(mg, data)
```

## Les options

On peut également spécifier deux éléments grâce aux paramètres optionnels de la fonction du constructeur.

```
w = Scenarios(mg, data; same_year=true, seed=[x for x in 1:ns])
```

- ▶ Le paramètre `same_year` permet de spécifier si l'on souhaite que les mêmes données soient utilisées sur l'ensemble de l'horizon temporel.
- ▶ Le paramètre `seed` permet de choisir les scénarios à utiliser si l'on souhaite générer de la reproductibilité ou si les scénarios ont un ordre particulier. (Par exemple un accroissement des prix au fil des années).

# Récapitulatif sur les scénarios au sein du package

## La Banque de donnée

- ▶ La banque de données contient de nombreuses entrées concernant une multitude d'éléments utilisable pour le réseau ;
- ▶ Il n'est pas nécessaire de tous les utiliser ;
- ▶ Les données sont soit sous la forme de matrices  $[nh, 1, ns]$  si elles concernent un pas d'opération ( $\Delta h$ ) soit  $[1, ns]$  si elles concernent un pas d'opération ( $\Delta y$ ).

## Le constructeur

- ▶ Le constructeur vient puiser dans la banque de données des scénarios et distribue dans des vecteurs correspondant à la forme de la structure microgrid les informations relatives aux différents composants du microgrid ;
- ▶ Les séries temporelles longues de une année sont alors agrégées pour prendre la forme qui sied à l'horizon temporel du microgrid considéré.

## Attention !

Le nombre de scénarios contenu dans la banque de données doit être supérieur ou égale à celui spécifié dans les paramètres du microgrid étudié.



# TP4 - Notions à acquérir

## Création d'une banque de données

- ▶ Savoir identifier les données adéquates et nécessaires à la création de scénarios (Durée, fréquence, nombre ...);
- ▶ Être en mesure de mettre en forme les données collectées pour que leurs dimensions soient cohérentes;
- ▶ Créer et sauvegarder un Dictionnaire de données en .jld2.

## Création de Scenarios

- ▶ Pouvoir adapter la création de scénarios pour accueillir de nouvelles séries de données;
- ▶ Savoir utiliser la fonction de création de scénario.

## Utilisation des Scenarios

- ▶ Récupérer les données au sein des scénarios pour les utiliser dans d'autres fonctions;
- ▶ Comprendre les problématiques relatives à la robustesse et l'incertitude.

# TP4 - Modèles pour le barrage

## Pluie

La pluie en [mm] multipliée par la surface peuvent être ajoutée au volume pour augmenter le soc.

## Évaporation

L'évaporation est un volume calculé chaque heure selon la température de l'air à 2m  $T_{2m}$  [°C], l'irradiation  $G(i)$  [W/m<sup>2</sup>] et le vent [km/h]. la formule que l'on utilise est la suivante (elle n'est ni juste ni homogène) :

$$(6T_{2m} + 6G(i) + 2V_{vent}) * 10^{-4} \quad (3)$$

# Étude de la construction d'une banque de données



Démonstration par un exemple concret

Voir le début du script TP4.1

# TP4 - Consignes

## Banque de données : Détails des consignes

1. D'après les modèles de barrage fournis, de quelles données avons nous besoin ?
2. Mettre en forme les données de vent et de température pour le pas horaire. Ci-après une liste de fonctions utiles : `CSV.File`, `DataFrames.DataFrame`, `DataFrames.filter!`, `repeat`, `x[2:2:end]`, `.+=`, `Hour`
3. Modifier la création de la banque de données pour y insérer les données de vent.

## Utilisation des scénarios - Détails des consignes

4. Modifier les fonctions de mise à jour du barrage et y inclure une diminution du volume en utilisant l'équation d'évaporation fournie. Pour cela vous devez :
  - 4.1 récupérer les données concernées et les stocker dans la structure barrage via `update_operation_informations` ;
  - 4.2 modifier : `mutable struct` `Barrage` <: `AbstractStorage`, `preallocate!`, `get_pluie`, `get_evaporation`, `update_operation_informations`, `compute_operation_soc`.