Parallel Statistical Model Checking for Safety Verification in Smart Grids

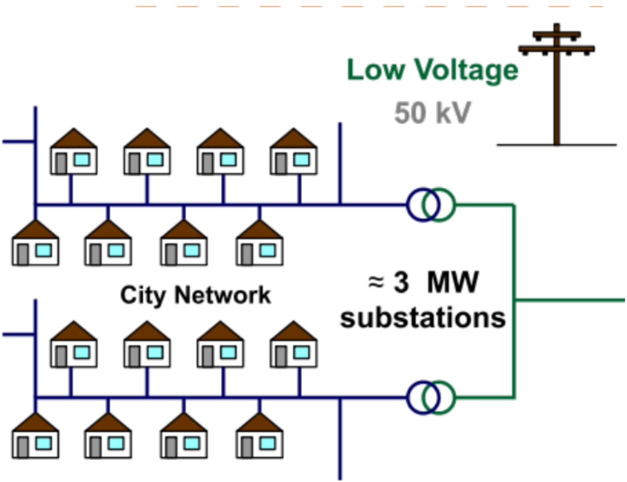
Enrico Tronci

joint work with Toni Mancini, Federico Mari, Igor Melatti, Ivano Salvo, Jorn Klaas Gruber, Barry Hayes, Milan Prodanovic, Lars Elmegaard

IWES 2018 – University of Siena

Sapienza Università di Roma

Electric Distribution Network: Substations and Houses



Autonomous Demand Response

- Distribution System Operators (DSOs) compute price tariffs for residential users
- Expected Power Profiles (EPPs): how residential users will respond to price tariffs
- DSOs compute price tariffs so that EPPs do not threat substations safety
  - in each t, Aggregated Power Demand (APD) must be below the substation safety power threshold (e.g., 400 kW)

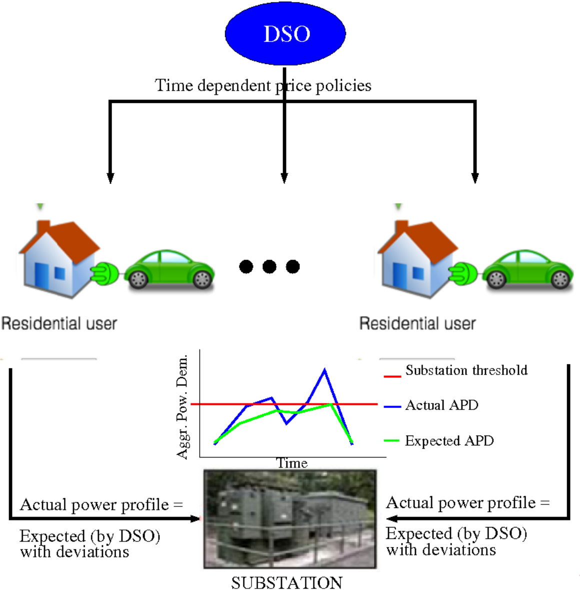
Autonomous Demand Response

- Residential users may or may not follow their corresponding EPPs
  - there may be automatic tools to enforce EPPs
  - implemented on small devices on users premises
  - still, there is no guarantee, due to unexpected needs, bad forecasts, limited computational resources, etc.

Problem

Given that users may deviate from EPPs with a given probability distribution, what is the resulting probability distribution for the APD?

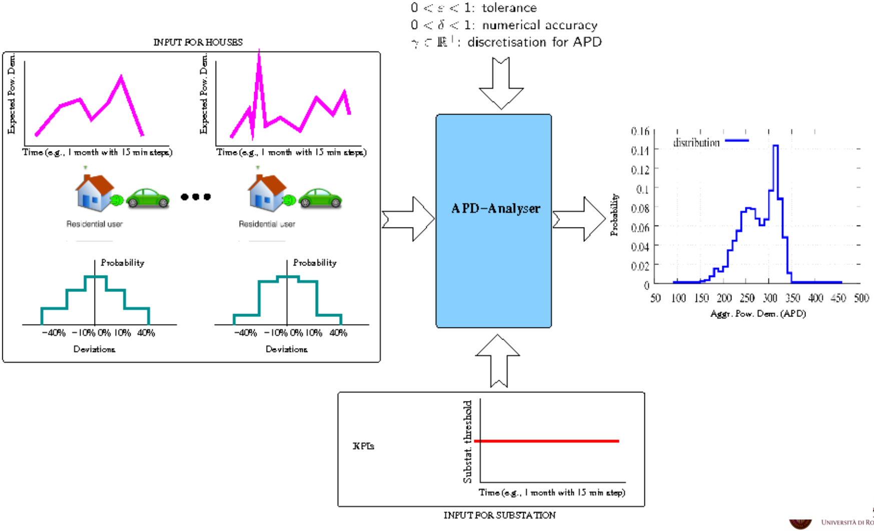
Problem at a Glance



APD-Analyser

- We present the APD-Analyser tool
- Main goal: compute the probability distribution for the APD
- So as to compute KPIs on it
  - probability distribution that a given substation threshold is exceeded
  - rank APD probability distributions according to their similarity to desired shapes

APD-Analyser: Input and Output



APD-Analyser: Input

- Set of residential users U connected to the same substation
- Period of time T (e.g., one month), divided in time-slots (e.g., 15 minutes)
- Expected Power Profiles (EPP)
  - one for each user $u \in U$: for each time-slot $t \in T$, the expected power demand of u in t
  - $p_u : T \to \mathbb{R}$
  - if $p_u(t) < 0$, production from PV panels exceeds consumption in time-slot t
- A probabilistic model for users deviations from EPPs
  - a real function $dev_u : D_u \to [0,1]$, for each user $u \in U$
  - $\int_{D_u} dev_u(x)dx = 1$
  - $\int_a^b dev_u(x)dx =$ probability that actual power demand of u in any time-slot $t \in T$ is in $[(1+a)p_u(t), (1+b)p_u(t)]$

APD-Analyser: Input

- Substation safety requirements
  - $p_s : T \to \mathbb{R}$
  - for each $t \in T$, DSO wants the APD to be below $p_s(t)$
  - that is, $\forall t \in T \to \sum_{u \in U} p_u(t) \leq p_s(t)$
- Key Performance Indicators (KPIs)
  - e.g., probability distribution that $p_s(t)$ is exceeded in any $t \in T$
- Parameters
  - $0 < \delta, \varepsilon < 1$: as for output probability distributions, the values must be correct up to tolerance ε with statistical confidence δ
    - $\Pr[(1-\varepsilon)\mu \leq \tilde{\mu} \leq (1+\varepsilon)\mu] \geq 1-\delta$
    - μ: (unknown) correct value, $\tilde{\mu}$: computed value
  - $\gamma \in \mathbb{R}^+$: discretisation step for output probability distribution

APD-Analyser: Output

- Probability distribution for APD resulting from EPPs disturbed with given probabilistic disturbance model
  - easy to evaluate KPIs once such distribution is computed
  - formally: $\Psi(v, W)$ is the probability that APD takes a value in interval W in any time-slot t s.t. $p_s(t) = v$
- Exactly computing Ψ is infeasible, thus we compute $\tilde{\Psi}$ as a (ε, δ) approximation of a γ-discretisation of the APD
- For each γ-discretised value $w = \text{APD}_{min} + k\gamma$, and for $v \in p_s(T)$, we compute $\tilde{\Psi}(v, w)$ s.t., with confidence at least $1 - \delta$:
  - if $\tilde{\Psi}(v, w) = \perp \notin [0, 1]$ then $\Psi(v, [w, w + \gamma)) < \varepsilon$
  - otherwise, $\Psi(v, [w, w + \gamma))$ is within $(1 \pm \varepsilon)\tilde{\Psi}(v, w)$

APD-Analyser: Algorithm

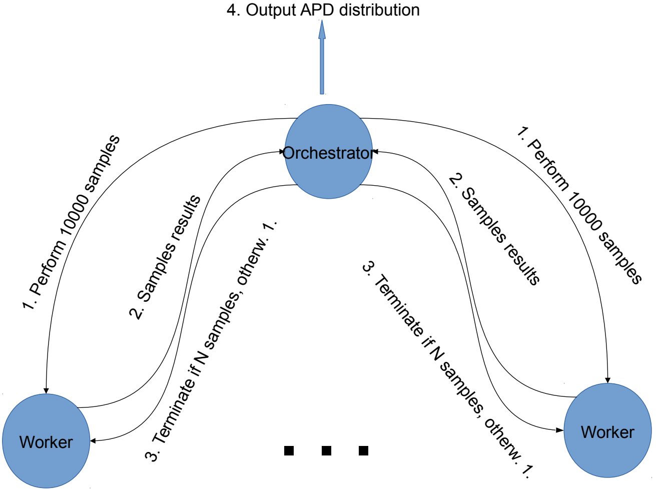
- *Monte-Carlo model checking*
  - goal: estimate the mean of a 0/1 random variable $Z_{v,w}$
  - taken at random a $t \in p_s^{-1}(v)$, is the value of the APD inside w, when perturbed using deviations model dev_u?
- Method: perform N independent experiments (samples) for $Z_{v,w}$, and then the mean is $\frac{\sum_{i=1}^{N} Z_i}{N}$
  - Optimal Approximation Algorithm (OAA) by Dagum & al. (2000) + Quantitative Model Checking (QMC) by Grosu & Smolka (2005)
  - the value of N is automatically adjusted, at run-time, while performing the samples
  - so that the desired tolerance ε is achieved with desired accuracy δ

APD-Analyser: HPC Algorithm

- N can be prohibitively high
  - easily order of 10^9 in our experiments
  - if performed with a sequential algorithm, order of 1 month for the computation time
- We re-engineer the OAA to be run on a HPC infrastructure, i.e., a cluster
  - main obstacle: value of N depends on samples outcomes! To be computed at run-time
- One *orchestrator* node instructs *worker* nodes to perform given number of samples
  - worker nodes perform samples in parallel and send results to the orchestrator
  - the orchestrator is responsible for termination checking
  - that is: is current number of samples ok for desired ε, δ?
- As a result, less than 2 hours of computation

APD-Analyser: HPC Implementation Sketch

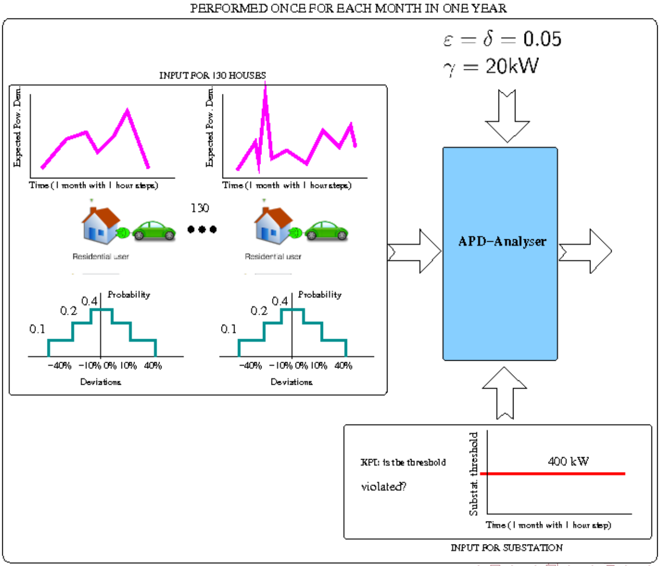
Steps 3-4: Monte-Carlo OAA (Dagum2000) and QMC (Grosu2005)



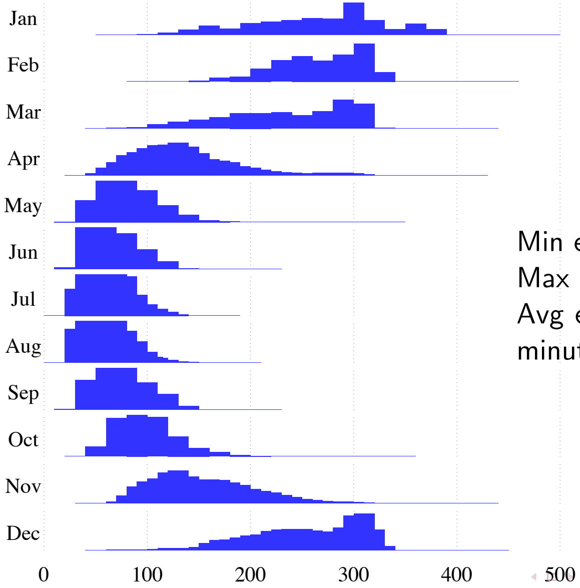
Experimental Evaluation: Case Study

- 130 houses in Denmark, all connected to the same substation
- EPPs computed by using methodologies from the literature
    - namely, computed as collaborative users which respond to individualised price policies
- Very liberal deviation model: up to $\pm 40\%$ variations with 10% probability, up to $\pm 20\%$ variations with 20% probability
- Challinging scenario: we want to compute the APD for each month of the year
    - by using time-slots of 1 day, we have $5^{30 \times 130}$ overall number of deviations

Experimental Evaluation: Case Study



Experimental Results



Min exec time: 4782 secs
Max exec time: 6448 secs
Avg exec time: 1 hour, 28 minutes and 7 seconds

Experimental Results: HPC Scalability

# workers	samples/sec	speedup	efficiency
1	*5924.89*	1×	100%
20	79275.028	13.38×	66.90%
40	162578.98	27.44×	68.60%
60	257791.96	43.51×	72.52%
80	335823.24	56.68×	70.85%

Conclusions

- We presented the HPC-based tool APD-Analyser
- Main purpose: support DSOs in analysing effects of price policies on aggregated power demand (APD) at substation level
  - especially for highly-fluctuating and individualised price policies
- From expected power profiles disturbed by probabilistic deviations (input) to probability distribution for APD (output)
- As a result, APD-Analyser enables safety assessment of price policies in highly dynamic ADR schemas

Thanks!