ECAI 2024
U. Endriss et al. (Eds.)
© 2024 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA241025

Gas Grid Copilot: Can a MORL Agent Assist a Dispatcher in Managing a Gas Grid?

Alexander Schiendorfer[a,*,1], Gheorghe Lisca[a,1], Karima Outafraout[a], Lilia Michailov[b], Pascal Kätzel[b] and Rudolf Felix[b]

[a]Technische Hochschule Ingolstadt, Institute AImotion Bavaria
[b]PSI FLS Fuzzy Logik & Neuro Systeme GmbH
ORCID (Alexander Schiendorfer): https://orcid.org/0000-0002-5283-5304, ORCID (Gheorghe Lisca): https://orcid.org/0000-0002-1518-4514, ORCID (Karima Outafraout): https://orcid.org/0000-0003-1781-6121

Abstract. The distribution of fuel gases is undergoing major changes due to decarbonization efforts: Non-fossil gases such as biomethane or renewable hydrogen can lead to the reuse of existing gas infrastructure for gas storage, transport, and distribution to reduce greenhouse gas emissions while maintaining a high energy security. For safe and efficient operation, we propose *Gas Grid Copilot* (GGC) as a demonstrator of a multi-objective reinforcement learning agent that trains in a simulated gas grid environment to control a grid by modifying its inflow into a mass storage. Multiple, possibly conflicting reward signals are included. Their conflicts and synergies of rewards are analyzed using techniques from multi-criteria decision making, more specifically a conflict interaction matrix based on extended fuzzy logic. That way, dispatchers of a gas grid can explore the effects of reward prioritizations and their consequences safely.

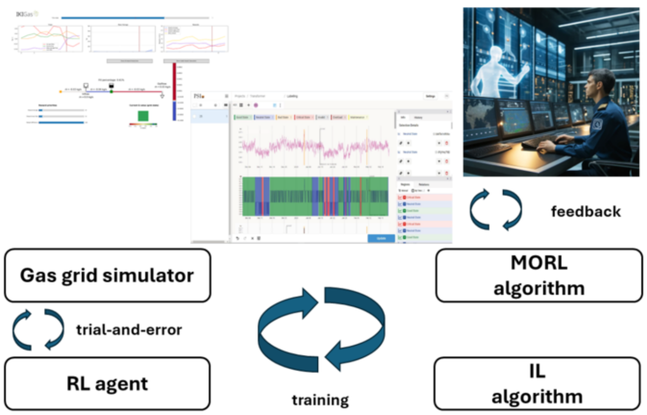

Figure 1. *Gas Grid Copilot* system overview.

1 Introduction

National and international natural gas distribution systems undergo massive changes due to the ongoing energy transition [2]. Until the infrastructure allows for a fully renewable heating system, ensuring an uninterrupted energy supply for industry and households remains a critical challenge – the total length of Germany's gas grid alone is more than 500,000 km with natural gas being the second most important primary energy source in Germany's energy mix [13]. This fact is also reflected in the decision to include natural gas in the EU taxonomy for sustainable transformation [12]. Additionally, as the supply and demand for non-fossil gases, such as biogas or green hydrogen obtained from power-to-gas, continues to grow [11], AI methods developed for currently available gas grids can be readily applied to retrofitted/repurposed existing gas infrastructure.

A gas network dispatcher must ensure that gas (natural gas including synthetic one, biogas, hydrogen) is delivered safely, reliably, and efficiently from suppliers to industrial, commercial, and residential consumers, thereby respecting physical constraints such as pressure limits and transmission velocities. The dispatchers' decisions must account for a large diversity of parameters: *linepack* (i.e., the total volume of gas contained within the system), gas pressure, and *demand and weather forecasts* – esp. when it comes to heating demands. Simultaneously, the decisions must, in general, fulfill multiple objectives: all consumer demands are fulfilled, the linepack and gas pressure are within defined limits, the oscillations in gas entry flow are minimal.

> We showcase ***Gas Grid Copilot*** **(GGC)**, our demonstrator capable of learning to control the inflow of gas within a gas distribution grid modeled in pandapipes [5], while accounting for multiple objectives that can be prioritized at runtime.

GGC's purpose is to illustrate, on a simplified use case (see Figure 2), how a dispatcher can be supported in the decision-making processes required for managing a gas network [1]. Generally, actions include the management of inflow into hydraulic groups at a certain point in time or opening/closing valves. Effects, and therefore rewards, are typically delayed by several hours due to inertia. Aiming for effective assistance with a human in the loop, GGC simulates the situation in a control room, where measurements of the state of the gas grid are continuously monitored and control actions are recommended to the dispatcher – depending on the *current* prioritization of the objectives. This is how an AI-based recommendation would be deployed in production: it will observe the grid's state and dispatcher's actions, but have no access to the grids's control system, due to regulation of critical infrastructure. In order to also address expectation conformity, Imitation Learning (IL) will be employed.

GGC's main contributions are:

* Corresponding Author. Email: alexander.schiendorfer@thi.de
[1] Equal contribution.

1. it combines a simulator of a gas grid with MORL
2. it showcases how a Multi-Objective RL (MORL) agent can assist a dispatcher with multiple policies corresponding to different reward prioritization in fictional grid states, and
3. it shows how the expected returns of a given policy interact (i.e., tradeoffs, synergies, etc.) using extended fuzzy logic and multi-criteria decision making.

2 System overview

The *Gas grid simulator*, *RL agent*, and *MORL algorithm* (see Figure 1) represent GGC's main components, of which the *MORL algorithm* is the one interacting directly with the dispatcher (user of the demo).[2] The interaction between the *Gas grid simulator* and *RL agent* forms the *reinforcement learning* loop, and allows the *RL agent* to find out the effects of its control actions, in terms of multiple rewards and, therefore, returns. Once trained, the user can step back and forth in a timeline and investigate the actions the agent suggests as well as the consequences over time, along with the rewards. The *Gas grid simulator* uses the pandapipes library [5] to describe the topology of the gas grid and simulate the flow of the gas within it, depending on the control actions applied to its active components (valves, compressors, flow rates). Despite its simplicity, pandapipes is a versatile tool for hydraulic and thermodynamic steady-state and quasi-stationary simulation, thereby enabling quick prototyping and evaluation of complex RL algorithms. The *RL agent* takes as observational input the complete state of the gas grid, and outputs the control actions which will be applied to the network's active components. *RL agent*'s policy is implemented as a neural network coded in PyTorch, based on StableBaselines3 [8].

The interactions between the *Gas grid simulator*, *RL agent*, *MORL algorithm*, and *IL algorithm* (which provides one or multiple rewards that represent conformity to observed policies) form the *training* loops. The *MORL algorithm* will continuously strive to improve the *RL agent*'s capabilities of achieving the different objectives while making trade-offs which are increasingly acceptable for dispatchers.

The interaction between the *MORL algorithm* and the dispatcher forms the *feedback* loop, and allows the *RL agent* to: 1) signal to the dispatcher the imminent risks of having some parameters exceeding the predefined limits and propose control actions which should effectively mitigate these risks, and 2) collect and learn from the feedback of the dispatcher on how valid the signaled risks and how effective the proposed mitigating actions were. The *feedback* loop is of particular importance because, by allowing human intervention during training, this loop enables the development of safe RL algorithms [9].

3 Features and functionalities

GGC focuses on a relatively simple scenario (see Figure 2): a small gas grid is defined in pandapipes, consisting of an external grid source, a local gas source (e.g., a biogas plant), a local sink (e.g., an industrial plant), and a mass storage – the main asset to be controlled. These components are available in the pandapipes core library and connected using pipe segments based on realistic dimensions (i.e. pressure limits, distances, material constants, and diameters). In this scenario, the agent can control the mass storage inflow/outflow such that ideally, locally sourced gas is conserved for when it is later needed – again locally. The external grid can supply or consume – depending on what is currently needed. Pandapipes is capable of performing steady-state hydraulic and thermodynamic simulation to solve for velocities and pressures occurring in the grid – and detecting feasibility of the hydraulic flows, in the first place. It does so by iteratively solving a non-linear system of equations using a CPU-based Newton-Raphson solver. All of that together makes up the *Gas grid simulator* part.

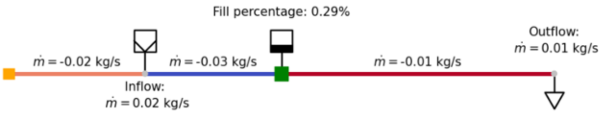

Figure 2. The minimalistic sample gas grid of this demonstrator, consisting of an external grid (in orange), a local source such as, e.g., a biogas plant, a mass storage, and a local sink, e.g., an industrial plant.

To train RL agents on top of pandapipes, a wrapper for the classical Gymnasium environment as well as the MO-Gymnasium [4] (for multiobjective environments) by the Farama foundation is included in the demo. As the gymnasium interfaces requires a `step` method that the agent calls repeatedly during training and inference episodes, we had to implement a wrapper that allows for one-timestep progression of the pandapipes time series simulation. Usually, it is intended for analysis of multiple time steps at once. The gas flows of the local source and sink are predefined–loaded from a time series, the external grid's flow are calculated in simulation, as are the velocities and pressure values. Three reward signals are derived from the states calculated in simulation:

1. *Reward Storage* If the fill percentage of the mass storage is between 25% and 75%, positive reward is issued with a maximum at 50%. This is implemented as a piecewise linear, triangular function.
2. *Reward External Grid Mass Flow* The negative absolute value of external in or outflow in kg/s which optimally should be 0. This represents the goal of self-sufficiency.
3. *Reward Difference* Ideally, dispatchers do not have to change the inflow settings frequently. Therefore, the negative absolute difference between the previous inflow A_{t-1} and A_t is returned as a reward.

They define the three-dimensional reward $\mathbf{r} = [r_1, r_2, r_3]$ of the environment which is not immediately scalarized.

When it comes to training agents, we currently support two implementations: A "naive" MO-RL agent which uses a user-defined weighting $\mathbf{w} = [w_1, w_2, w_3]$ such that a total reward $r_{\text{total}} = \mathbf{w}^T\mathbf{r}$ is obtained which can be used with any scalar RL algorithm. Our demo offers any weighting $\mathbf{w} \in \{1, 10, 100\}^3$ which amounts to 27 agents that have each been trained individually for 10,000 timesteps. This pretraining is carried out such that users may explore different trajectories of rewards as well as mass storage fill percentages and actions interactively. As a "free" add-on, the critic network's q-values are visualized to give dispatchers a visual feedback of the quality of the current state. By contrast, a "true" MORL implementation is also provided using Multi-Objective Natural Evolution Strategies (MONES) [6] which searches for an approximation of the Pareto frontier of policies, i.e., all policies whose vector of expected returns (in different rewards) are not Pareto-dominated. Instead of a single optimal policy w.r.t. one scalar reward, we get to choose from different tradeoffs. Figure 4 demonstrates a three-dimensional Pareto landscape where every point refers to one policy and its associated expected returns in every reward dimension. Dominated policies are grayed out and shown for demonstration only.

[2] Source code: https://github.com/AImotion-Bavaria/gas_grid_copilot

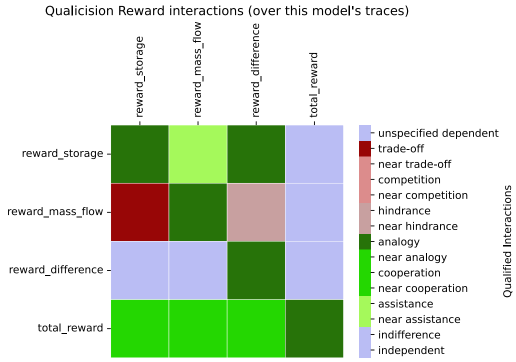

Figure 3. An exemplary interaction matrix using extended fuzzy logic.

Finally, a special feature of GGC is the integration of *conflict and interaction matrices* into the interactive decision phase. For this purposes, sequences $S_0, A_0, \mathbf{R}_0, S_1 \ldots$ are collected from the training episodes. Then, a multi-criteria conflict and interaction analysis [3] is performed to check whether the rewards are positively correlated ("harmonizing goals") or conflicting. Cooperations, tradeoff situations, and other possible interaction patterns are visualized in an interaction matrix as shown in Figure 3. Here, we can see that keeping the external mass flow is a tradeoff with keeping the storage in desirable boundaries because we need a certain non-zero external flow to keep the filling percentage there. On the other hand, reward storage and reward difference can go hand in hand if the inflow level is sufficiently high that no change still keeps the filling percentage in the good range. These analyses are performed based on the commercial software *Qualicision AI* which can then also be employed to rank a set of decision alternatives (here, policies) according to a user's criteria preferences. The theory is derived from extended fuzzy logic (i.e., satisfaction membership functions mapping to $[-1, 1]$) which is why the reward signals have to be adjusted to fit that range.

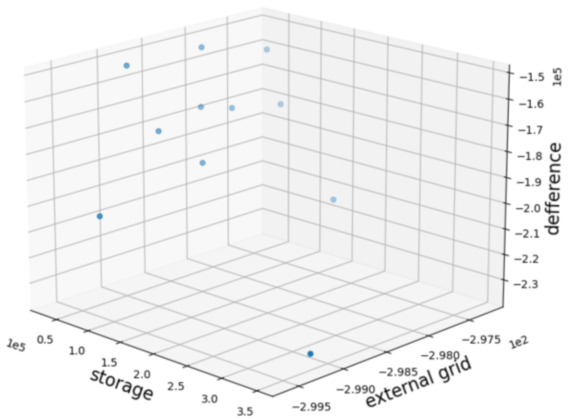

Figure 4. The 3D Pareto frontier determined by MONES. Each vector corresponds to one policy in the population, shaded ones are dominated.

4 Conclusion and Future Work

The main purpose of this demonstration, GGC, is to introduce concepts such as multiple rewards and reinforcement learning to the intended expert audience, dispatchers of a gas grid. Therefore, it was important to showcase how different policies would act in the same

Figure 5. *Qualitative Labeling with Qualicision AI* [7].

state of the grid. However, this initial grid is clearly too simplistic for everyday use. Therefore, a more realistically sized grid is currently modeled and analyzed.

This next experimental grid is currently prepared in the IKIGas project [1] by energy provider Avacon along with a more professional gas grid simulation, PSIGanesi [10]. For this grid, historicized data of inflow, outflow (consumption), and linepack is available. The *IL algorithm* based on Generative Adversarial Imitation Learning (GAIL) will enable the *RL agent* to learn from past control actions of the dispatchers. Discrepancies between proposed and actually taken actions could then be visualized "in the moment".

One limitation that we already found in the approach, given historical data, is the current lack of weather forecasts and gas demand in the observation space. We plan on extending GGC with publicly available weather information as well as recorded past forecasts: At one extreme, if the dispatcher's forecasted demands turn out to be lower than the real consumption, then the consumers will deplete the gas supply available in the network and generate shortage. At the other extreme, the gas network remains overloaded if the predicted demands turn out higher than the real consumption. Both extremes generate substantial financial expenses.

For now, GGC uses three reward signals that can be calculated based on a given state in the simulation. In practice, however, sometimes the dispatchers can only *qualitatively* deem a state as good or bad – based on intuition and past reference values. To capture this kind of "fuzzy" information, we plan to use two additional sources of reward: i) using *Qualitative Labeling with Qualicision* (see Figure 5) [7], time series can be conveniently labeled into different categories, e.g., using a Likert scale from 1 to 5. The dispatcher is presented with grid parameter plots, and asked to segment out intervals and label them with the quality of their values. ii) the anomaly/normalcy scored reported by trained generative model parameterized by θ, $p(s \mid \theta)$ could indicate the likelihood of the observed state s (e.g., its linepack value). A high value of $p(s \mid \theta)$ indicates a "normal" observation that the agents should strive for, a low value would show an anomaly.

By addressing critical challenges such as forecasting demand, labeling grid parameters, and optimizing control strategies, RL agents can indeed assist a dispatcher to manage also a realistic gas grid.GGC aims at a first impression for domain experts to familiarize themselves with the capabilities of MORL and add it to their workflow.

Acknowledgements

This research has been funded by the Federal Ministry for Education and Research (BMBF), Germany under the "Research for Civil Security" program, project IKIGas (13N16369).

References

[1] A. Baschin, M. Baschin, R. Böselt, R. Felix, C. Fernandez, S. Gehring, A. Görtz, L. Harpeng, Y. Hei, N. Hildebrandt, et al. Industrielle künstliche intelligenz für sichere gasnetze. In *Tagungsband 18. Symposium Energieinnovation 2024*. Technische Universität Graz, 2024.

[2] A. Energiewende. Ein neuer Ordnungsrahmen für Erdgasverteilnetze. Analysen und Handlungsoptionen für eine bezahlbare und klimazielkompatible Transformation, 2023.

[3] R. Felix. On consistency of decision goals and separability of preferences of decision alternatives. In *Proc. IFSA-EUSFLAT-15*, pages 844–850. Atlantis Press, 2015.

[4] F. Felten, L. N. Alegre, A. Nowé, A. L. C. Bazzan, E. G. Talbi, G. Danoy, and B. C. da. Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

[5] D. Lohmeier, D. Cronbach, S. R. Drauz, M. Braun, and T. M. Kneiske. Pandapipes: An open-source piping grid calculation package for multi-energy grid simulations. *Sustainability*, 12(23), 2020. ISSN 2071-1050. doi: 10.3390/su12239899. URL https://www.mdpi.com/2071-1050/12/23/9899.

[6] S. Parisi, M. Pirotta, and J. Peters. Manifold-based multi-objective policy search with sample reuse. *Neurocomputing*, 263:3–14, 2017.

[7] PSI FLS Fuzzy Logik & Neuro Systeme GmbH. Qualitative Labeling with Qualicision AI. https://www.fuzzy.de/fileadmin/Resources/Public/pdf_english/pm/PM_2024-01_Qualitative_Labeling_with_Qualicision_AI_EN.pdf, May 2024.

[8] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[9] W. Saunders, G. Sastry, A. Stuhlmueller, and O. Evans. Trial without error: Towards safe reinforcement learning via human intervention. *arXiv preprint arXiv:1707.05173*, 2017.

[10] D. Scheibe and A. Weimann. Dynamic gas network simulation with ganesi. *GAS UND WASSERFACH GAS ERDGAS*, 1999:610–617, 1999.

[11] W. Terlouw, D. Peters, J. van Tilburg, M. Schimmel, T. Berg, J. Cihlar, G. U. R. Mir, M. Spöttle, M. Staats, A. V. Lejaretta, et al. Gas for climate. the optimal role for gas in a net-zero emissions energy system. *Navigant Netherlands BV, März*, 2019.

[12] E. Union. EU taxonomy for sustainable activities. https://finance.ec.europa.eu/sustainable-finance/tools-and-standards/eu-taxonomy-sustainable-activities_en, 12.07.2023.

[13] Wirtschaft und Klimaschutz, BMWK - Bundesministerium für. Erdgasversorgung in deutschland. https://www.bmwk.de/Redaktion/DE/Artikel/Energie/gas-erdgasversorgung-in-deutschland.html, 21.06.2023.