Energy Resources Scheduling in Energy Communities: A comparison between Mixed Integer Linear Programming and Hybrid-adaptive Differential Evolution with decay function

Eduardo Gomes, Lucas Pereira, Hugo Morais *Senior Member, IEEE*

***Abstract*—With the ever growing number of technologies that interact with the grid, optimization and control strategies become critical to ensure the quality of service without disruption. This work analyses the application of two operational planning optimization methods, Mixed Integer Linear Programming (MILP) and Hybrid-adaptive Differential Evolution with decay function (HyDE-DF) metaheuristic, applied to an Energy Community including generation, consumption flexibility, energy storage systems, electric vehicles with vehicle-to-grid capability and the electricity exchange with the main network, with the aim of minimizing operation costs. Throughout this work, differences between approaches are highlighted, as well as considerations for each method to help convergence and improve solution quality. Both methods achieve comparable operation costs, 159.3330 and 153.4883 for MILP and HyDE-DF, respectively.**

***Index Terms*—Electric Vehicles, Energy Communities, Hybrid-adaptive Differential Evolution, Metaheuristics, Mixed Integer Linear Programming**

I. Introduction

Energy transition is an ever growing multidisciplinary topic of research, bringing about innovation to multiple sectors that affect daily life [1]. European Union has proposed set a set of guidelines defining the targets to achieve net-zero greenhouse gas emissions by 2050, presenting strategies such as increasing green energy resources and investment on identified key sectors. Among highlighted sectors are electricity, transportation and industry [2], [3].

Optimization is a research field that aims to provide the user with information on what is the best solution for a given problem, finding applications in a multitude of fields. Stork et al. [4], present a taxonomy for optimization algorithms, as well as providing a review of existing algorithms and applications that range from hyperparameter optimization in neural networks to deciding the best path to take, i.e. travelling salesman problem.

In the energy field, optimization is ever present in several problems. Active research areas include forecasting [5], storage [6], load disaggregation [7], scheduling [8], [9], and energy trading [10]. Other research problems include achieving maximization of self-consumption [11] and cost minimization [12]. How these problems are modelled determines the choice of algorithm for solving, directly influencing the degree of precision of the solution. Linear Programming (LP) methods are widely known, having a deterministic and exact solution. On the other side of the spectrum lie function approximation methods that aim to approximate to a value as best as possible, with examples of algorithms such as metaheuristics.

Considering the challenges that ECs face, optimization and planning becomes a key area of intervention with works exploring how to best use the various RES of a community such as EVs, storage and energy trading. In [13], the authors study the use of a community storage system and apply a multi-objective optimization (MOO) method taking into account grid costs and emissions, with results showing a benefit of using said system. In [12], a LP approach is adopted in a study on the impact an EV-centered market approach for ECs, where the authors show benefits to the network whenever a prosumer-to-vehicle (P2V) strategy is considered. Various other studies have tackled the issue of managing a Community, with different approaches such as Reinforcement Learning being also considered [14].

This work provides with two main contributions - the first lies on a *Python* implementation of the HyDE-DF algorithm [15], with the second being the comparison between the implementation of the same problem using a Mixed Integer Linear Programming approach and a HyDE-DF optimization approach on the optimization of an ECs, emphasizing on the issues and required steps to achieve comparable results.

II. Materials and Methods

The entirety of this work was implemented in *Python*, with the code being publicly available in [16]. The implementation counts with two major components - a mixed integer linear programming approach, implemented using the *Pyomo* package (6.4.0) [17], and IBM ILOG CPLEX Optimization Studio 22.1.0. The second component is a fresh implementation of the HyDE-DF algorithm that was originally developed in MATLAB.

Pyomo is a package for *Python* that focuses on optimization, providing a unified framework capable of handling different optimization techniques. Various known solvers including CPLEX can be used and changed without requiring specialized development, depending on the desired method.

Both components were developed on a laptop with an Intel(R) Core(TM) i7-8750H CPU and 16 GB of RAM.

Eduardo Gomes is with ITI/LARSyS, INESC-ID and Instituto Superior Técnico-IST, Universidade de Lisboa.

The implementation does not count with GPU-accelerated packages, relying entirely on the CPU.

A. Hybrid-adaptive differential evolution with a decay function (HyDE-DF)

HyDE-DF is a modified version of Hybrid-adaptive differential evolution (HyDE). Population generation is inspired by the original differential evolution algorithm [18], in which each population element is updated by a factor of the difference between two other elements. The operator, as described by the authors, is presented in Equation 1:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + \delta_G \cdot [F_i^1(\epsilon \cdot \vec{x}_{best} - \vec{x}_{i,G})] + F_i^2(\vec{x}_{r1,G} - \vec{x}_{r2,G}) \tag{1}$$

where $\vec{x}_{i,G}$ is the current vector, $\vec{x}_{best}$ represents the best found solution thus far, and $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ represent two individuals from the population. Vectors $\vec{x}_{i,G}$, $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ are different among themselves. The hybrid-adaptive component of the algorithm is responsible for keeping track of scale factors for each population element throughout iterations. These are represented in Equation 1 by F_i^1, F_i^2 and ϵ, where $\epsilon = \mathcal{N}(F_i^3, 1)$, inspired by [19]. The term $[F_i^1(\epsilon \cdot \vec{x}_{best} - \vec{x}_{i,G})]$ promotes convergence in the direction of current best candidate solutions [15], having its weight reduced over time by a decay factor δ_G , similar to Simulated Annealing (SA) [20].

The chosen encoding strategy for HyDE-DF solutions for this work is given by a vector of length N, where N represents the total number of variables of the problem. This is achieved through the process of turning the variable matrices in vectors, followed by concatenation. Each solution can be decoded as the size of each variable is known beforehand, as well as the placement within the encoded vector.

The HyDE-DF implementation for this work follows the implementation provided by the original authors in [21]. However, the original implementation only allows the definition of a single variable range, shared across all variables. This constitutes a challenge in problems where, for example, different variables operate in different ranges. Motivated by this issue, the current implementation of HyDE-DF allows for the specification of individual variable ranges, furthering the algorithm's capabilities.

Due to the chosen encoding, the interactions and dependencies between variables are not taken into account and thus feasibility of generated solution is not guaranteed. Following the generation of the new population at each iteration, every population member is analyzed and fixed based on the MILP constraints. As with the original implementation, *elitism* is promoted, with the best solution of each iteration injected into the next iteration. Several implementation optimizations were carried out to ensure this process is as fast as possible. The use of matrices instead of loops for variable fixing was prioritized to make code execution more efficient. The use of *Cython* [22], further contributed to trim the execution time by compiling the developed $Python$ code in C, taking advantage of the latter's speed. These improvements resulted in a final value of 35 iterations per second, from a starting value of 1.5 iterations per second when considering a population size of 10 individuals, using the same hardware.

Finally, besides the mentioned solution fixing, an initial solution was also provided to HyDE-DF to accelerate the convergence of the algorithm. The initial solution provided to the algorithm had maximum generation values for the generators, maximum import quantities, maximum charging station consumption and storage flags set to battery charging. The remainder of the variables are set to zero. This solution was given after empirically testing multiple variable combinations.

B. Problem Formulation

The work presented is based on a scenario described by [23], where an Energy Community composed by seven generators, six loads, three batteries and five electric vehicles is considered. Furthermore, it is possible to import and export energy to the main grid, as well as using electric vehicles to inject power into the grid. The scenario has a total duration of 24 hours, with a time resolution of one hour.

Each of the following subsections presents the constraints for the respective aspects of the scenario. The implementation approach for Electric Vehicles using HyDE-DF is also described, describing the heuristic utilized.

C. Power Imports and Exports

The considered scenario allows the use of energy supplied by the main grid(import) whenever the generation and power discharge from storage systems and EVs cannot meet the requested power demand of the loads and EVs. The possibility to supply energy to the main network is also contemplated to account for periods where there is a surplus of power and cannot be stored or there is a chance to earn revenue. Maximum values are defined for both import and export values, $pMaxImp$ and $pMaxExp$, respectively. Equation 2 represents the constraint for maximum import and Equation 3 represents the constraint for maximum export quantities.

$$pImp_t \leq pMaxImp, \forall t \in T \tag{2}$$

$$pExp_t \leq pMaxExp, \forall t \in T \tag{3}$$

D. Generation

The generation resources of this scenario are distinguished between renewable and non-renewable generators. Equation 4 represents the constraint on generated power by renewable sources $gAct$, based on forecasted values $gForecast$. The minimum and maximum power generated by non-renewable sources are contemplated by constraints 5 and 6. The binary variable $genXo$ represents the state of the non-renewable generator, complemented by the minimum generation value $gMinAct$ and maximum $gMaxAct$.

$$gAct_{g,t} + gExcAct_{g,t} = gForecast_{g,t}, \forall g \in G, \forall t \in T \tag{4}$$

$$gAct_{g,t} \geq gMinAct_{g,t} \times genXo_{g,t}, \forall g \in G, \forall t \in T \tag{5}$$

$$gAct_{g,t} \leq gMaxAct_{g,t} \times genXo_{g,t}, \forall g \in G, \forall t \in T \quad (6)$$

E. Loads

The considered loads are a parameter, drawn from forecast consumption values. These are grid needs that should be met, with either power from generators or stored. Load reduction, curtailment, as well as energy not supplied are considered as extra variables to relax the problem and avoid an instance in which the problem does not have a solution. Load reduction, $lRed$, considered in Equation 7 contemplates the possibility to reduce energy provided to a consumer, helping the EC manager to prevent service disruption. This can also be achieved by load curtailment $lCut$, in Equation 8, in which the power provided is completely cut off using a binary variable $loadXo$. Finally, Equation 9 describes the relationship between all load variables and considers the energy not supplied, $lENS$.

$$lRed_{l,t} \leq loadMaxRed_{l,t}, \forall l \in L, \forall t \in T \quad (7)$$

$$lCut_{l,t} = load_{l,t} \times loadXo_{l,t}, \forall l \in L, \forall t \in T \quad (8)$$

$$lENS_{l,t} + lRed_{l,t} + lCut_{l,t} \leq load_{l,t}, \forall l \in L, \forall t \in T \quad (9)$$

F. Storage

Batteries can be discharged to provide energy to the grid, as well as charged whenever possible. Constraints are imposed on the maximum storage and minimum storage by $sMin$ and $sMax$, respectively. Charging and discharging actions, $sDchXo$ and $sChXo$ are mutually exclusive as shown in Equation 10. Each action has associated maximum values, represented by $sDch$ and sCh, presented in Equations 11 and 12. Equation 15 describes the update of battery state, $sEner$, establishing the required temporal dependencies, where the efficiency of charging (η_{ch}), and discharging (η_{ch}) is also considered. The initial state of charge (SOC) of batteries is considered a parameter, in the form of a percentage of maximum capacity. For the HyDE-DF approach, the variable $sMinRelax$ was not considered.

$$sDchXo_{s,t} + sChXo_{s,t} \leq 1, \forall s \in S, \forall t \in T \quad (10)$$

$$sDch_{s,t} \leq sMaxDch_{s,t} \times sDchXo_{s,t}, \forall s \in S, \forall t \in T \quad (11)$$

$$sCh_{s,t} \leq sMaxCh_{s,t} \times sChXo_{s,t}, \forall s \in S, \forall t \in T \quad (12)$$

$$sEner_{s,t} \leq sMax, \forall s \in S, \forall t \in T \quad (13)$$

$$sEner_{s,t} \geq sMin_{s,t} - sMinRelax_{s,t}, \forall s \in S, \forall t \in T \quad (14)$$

$$sEner_{s,t} = sEner_{s,t-1} + sCh_{s,t}\eta_{ch} - \frac{sDch_{s,t}}{\eta_{dch}}, \forall s \in S, \forall t \in T \quad (15)$$

G. Electric Vehicles

Similarly to the batteries, it is possible to charge and discharge the vehicles with the added restriction of requiring an active connection to a charging station. To aid the convergence of HyDE-DF, a heuristic was applied for electric vehicles. Whenever a vehicle is connected to a charging station, the next SOC requirement will be taken into account and the amount of timesteps required to meet this value are calculated. This is possible as the schedule for the vehicles is a considered parameter, resulting in a charging plan for the vehicle, effectively rewriting the generated solution altogether.

$$vDch_{v,t} + vCh_{v,t} \leq 1, \forall v \in V, \forall t \in T \quad (16)$$

$$vDch_{v,t} \leq vMaxDch_{v,t} \times vDchXo_{v,t}, \forall v \in V, \forall t \in T \quad (17)$$

$$vCh_{v,t} \leq vMaxCh_{v,t} \times vChXo_{v,t}, \forall v \in V, \forall t \in T \quad (18)$$

$$vEner_{v,t} \leq vMax, \forall v \in V, \forall t \in T \quad (19)$$

$$vEner_{v,t} \geq vReq_{v,t} - vMinRelax_{v,t} \forall v \in V, \forall t \in T \quad (20)$$

$$vEner_{v,t} \geq vMin - vMinRelax_{v,t} \forall v \in V, \forall t \in T \quad (21)$$

$$vEner_{v,t} = vEner_{v,t-1} + vCh_{v,t}\eta_{ch} - \frac{vDch_{s,t}}{\eta_{dch}}, \quad \forall v \in V, \forall t \in T \quad (22)$$

H. Charging Stations

Each station supports a single electric vehicles at any given time. A defined schedule and knowledge of when a vehicle will be connected to any given station is provided as parameter, with the binary variable $csXo$ indicating the connection state. Charging stations consider maximum and minimum charging limit, expressed by the $csMax$ and $csMin$ parameters in Equation 23 and 24. The HyDE-DF approach considers the handling of the charging stations as part of electric vehicle schedule building as they are closely related.

$$csPower_{c,t} \leq csMax_{c,t}, \forall c \in C, \forall t \in T \quad (23)$$

$$csPower_{c,t} \geq csMin_{c,t}, \forall c \in C, \forall t \in T \quad (24)$$

$$csPower_{c,t} = (vCh_{v,t} - vDch_{v,t}) \times csXo_{c,v,t}, \quad \forall c \in C, \forall v \in V, \forall t \in T \quad (25)$$

$$csPowerNet_{c,t} = (vCh_{v,t}\eta_{ch} - \frac{vDch_{v,t}}{\eta_{dch}}) \times csXo_{c,v,t}, \quad \forall c \in C, \forall v \in V, \forall t \in T \quad (26)$$

I. Balance and Objective Function

At each timestep, the grid balance of all components must amount to zero. This can be expressed by the difference of the sum of generation and imports, and load consumption, batteries, vehicles and charging stations. This equality is represented by Equation 27.

$$\begin{aligned} 0 = &\sum_{g}^{G} gAct_{g,t} - gExcAct_{g,t} + pImp_t - pExp_t \\ &- \sum_{l}^{L} load_{l,t} - lRed_{l,t} - lCut_{l,t} - lENS_{l,t} \\ &- \sum_{s}^{S} sCh_{s,t} - sDch_{s,t} \\ &- \sum_{v}^{V} vCh_{v,t} - vDch_{v,t}, \forall t \in T \end{aligned} \tag{27}$$

Equation 28 describes the objective function for the mixed integer linear programming part of this work, with its several components. Each variable has a cost associated designated by ϕ.

$$gens = \sum_{g}^{G} \sum_{t}^{T} gAct_{g,t}\phi_{g,t}^{gen} + gExcAct_{g,t}\phi_{g,t}^{NDE}$$

$$loads = \sum_{l}^{L} \sum_{t}^{T} lRed_{l,t}\phi_{l,t}^{red} + lCut_{l,t}\phi_{l,t}^{cut} + lENS_{l,t}\phi_{l,t}^{ENS}$$

$$stor = \sum_{s}^{S} \sum_{t}^{T} sDch_{s,t}\phi_{s,t}^{dch} - sCh_{s,t}\phi_{s,t}^{ch} + sMinRelax_{s,t} \times \phi_{s,t}^{sMin}$$

$$v2g = \sum_{v}^{V} \sum_{t}^{T} vDch_{v,t}\phi_{v,t}^{dch} - vCh_{v,t}\phi_{v,t}^{ch} + vMinRelax_{v,t} \times \phi_{v,t}^{vMin}$$

$$rest = \sum_{t}^{T} pImp_t\phi_t^{buy} - pExp_t\phi_t^{sell}$$

$$obj = gens + loads + stor + v2g + rest \tag{28}$$

To aid the convergence process of the HyDE-DF meta-heuristic, a penalty was added to the whenever there is a period with excessive import or export power. Equation 29 illustrates the adjusted objective function with the added $penalty$ component.

$$obj = gens + loads + stor + v2g + rest + \sum_{t}^{T} penalty_t \tag{29}$$

III. Results

Considering the MILP approach, a value of 159.3330 was achieved on the described scenario, against the obtained best of 153.4883 by the HyDE-DF method. The most significant difference between approaches is the time execution, with MILP achieving a solution in just 0.1898s and HyDE-DF requiring an average of 1769.1994s. Figure 1 depicts the evolution of ten runs of HyDE-DF over the course of 10000 iterations, with a population size of 50.

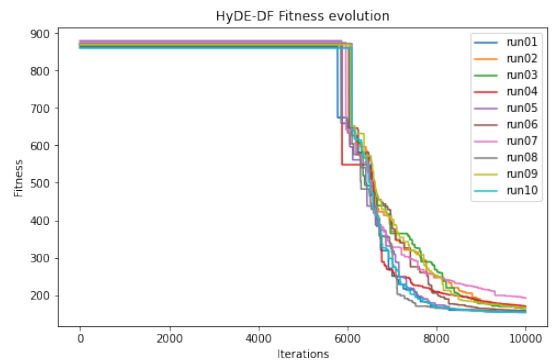

Fig. 1. HyDE-DF best solution fitness evolution

Table I summarizes the results of both approaches, contemplating total generation, total load reduced and curtailed, total battery and vehicle usage, as well as imports and exports.

TABLE I
TOTAL VARIABLE AMOUNTS

	MILP	HyDE-DF
pImp	829.5	629.8
pExp	0.0	0.0
gAct	1084.3	1151.1
gExcAct	0.0	0.0310
lRed	0.0	1.0330
lCut	0.0	0.0
lENS	0.0	0.1260
sDch	15.0	58.8
sCh	3.2	0.0
vDch	0.0	0.0
vCh	104.0	128.8

Regarding generation, the total of amount generated energy amounted to 1084.3 kWh and 1151.1 kWh for the MILP and HyDE-DF approaches. Figure 2 presents the values for the sum of $gAct$ and $gExcAct$ throughout each timestep for both methods.

In terms of imports and exports, MILP imported a total amount of 829.5087 kW, while HyDE-DF imported 629.7680 kW. Both approaches did not export any energy. Figure 3 represents the volume of imports and exports acquired by both methods. The total expenses on imports were of 71.1080 € for MILP and 55.9148 € for HyDE-DF.

IV. Conclusion

This work presented a comparison of two approaches on a scenario consisting of a micro-grid composed by seven generators, six loads, three batteries and five vehicles, as well as allowing for import and export of power. This work also

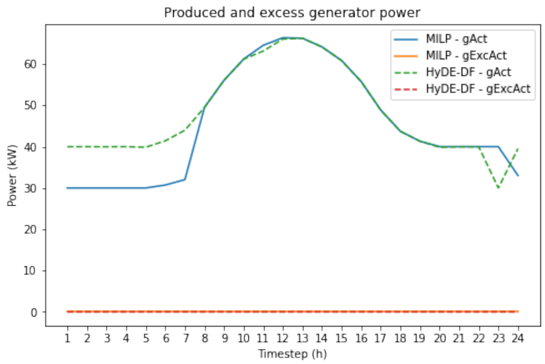

Fig. 2. Produced and excess generation values

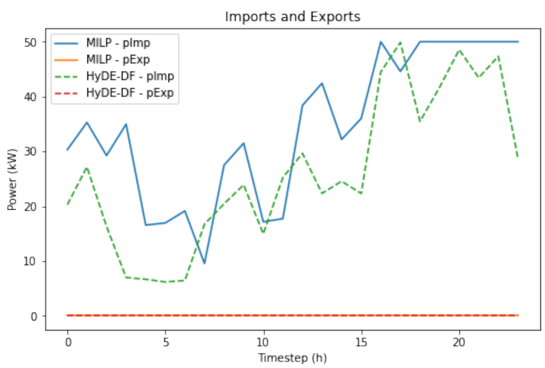

Fig. 3. Imports and export quantities

presents a contribution in the form of a *Python* implementation of HyDE-DF. The two considered approaches, Mixed Integer Linear Programming and the HyDE-DF metaheuristic produced very similar results, with the metaheuristic achieving a lower objective function value, as well as making more use of available storage resources, generators, while achieving a lower import cost through the scenario.

ACKNOWLEDGMENT

This work received funding from the European Union´s Horizon Europe research and innovation programme under grant agreement no. 101056765. Views and opinions expressed in this document are however those of the authors only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the grating authority can be held responsible for them. The authors received funding from the Portuguese Foundation for Science and Technology (FCT) under grant 2021.07754.BD, CEECIND/01179/2017, UIDB/50009/2020 and UIDB/50021/2020.

REFERENCES

[1] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, A. Luccioni, T. Maharaj, E. D. Sherwin, S. K. Mukkavilli, K. P. Kording, C. Gomes, A. Y. Ng, D. Hassabis, J. C. Platt, F. Creutzig, J. Chayes, and Y. Bengio, "Tackling climate change with machine learning."

[2] European climate law. [Online]. Available: https://ec.europa.eu/clima/eu-action/european-green-deal/european-climate-law_en

[3] E. Union. Roadmap 2050. [Online]. Available: https://www.roadmap2050.eu/project/roadmap-2050

[4] J. Stork, A. E. Eiben, and T. Bartz-Beielstein, "A new taxonomy of global optimization algorithms," vol. 21, no. 2, pp. 219–242.

[5] G. M. Tina, C. Ventura, S. Ferlito, and S. De Vito, "A state-of-art-review on machine-learning based methods for PV," vol. 11, no. 16, p. 7550.

[6] M. U. Hashmi, L. Pereira, and A. Bušić, "Energy storage in madeira, portugal: co-optimizing for arbitrage, self-sufficiency, peak shaving and energy backup," in *2019 IEEE Milan PowerTech*, pp. 1–6.

[7] E. Gomes and L. Pereira, "PB-NILM: Pinball guided deep non-intrusive load monitoring," vol. 8, pp. 48 386–48 398, conference Name: IEEE Access.

[8] H. Morais, P. Kádár, P. Faria, Z. A. Vale, and H. M. Khodr, "Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming," vol. 35, no. 1, pp. 151–156. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148109001001

[9] T. Soares, L. Carvalho, H. Morais, R. J. Bessa, T. Abreu, and E. Lambert, "Reactive power provision by the dso to the tso considering renewable energy sources uncertainty," *Sustainable Energy, Grids and Networks*, vol. 22, p. 100333, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352467719304886

[10] L. Gomes, H. Morais, C. Gonçalves, E. Gomes, L. Pereira, and Z. Vale, "Impact of forecasting models errors in a peer-to-peer energy sharing market," vol. 15, no. 10, p. 3543, number: 10 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1996-1073/15/10/3543

[11] G. Lorenzi and C. A. S. Silva, "Comparing demand response and battery storage to optimize self-consumption in PV systems," vol. 180, pp. 524–535.

[12] R. Faia, J. Soares, Z. Vale, and J. M. Corchado, "An optimization model for energy community costs minimization considering a local electricity market between prosumers and electric vehicles," vol. 10, no. 2, p. 129, number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[13] W. L. Schram, T. AlSkaif, I. Lampropoulos, S. Henein, and W. G. van Sark, "On the trade-off between environmental and economic objectives in community energy storage operational optimization," vol. 11, no. 4, pp. 2653–2661, conference Name: IEEE Transactions on Sustainable Energy.

[14] S. Zhou, Z. Hu, W. Gu, M. Jiang, and X. Zhang, "Artificial intelligence based smart energy community management: A reinforcement learning approach," vol. 5, no. 1, pp. 1–10, conference Name: CSEE Journal of Power and Energy Systems.

[15] F. Lezama, J. Soares, R. Faia, and Z. Vale, "Hybrid-adaptive differential evolution with decay function (HyDE-DF) applied to the 100-digit challenge competition on single objective numerical optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. Association for Computing Machinery, pp. 7–8.

[16] E. Gomes. eso_project. [Online]. Available: https://github.com/ECGomes/eso_project

[17] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: modeling and solving mathematical programs in python," vol. 3, no. 3, p. 219.

[18] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," vol. 11, no. 4, pp. 341–359.

[19] J. Brest, A. Zamuda, B. Boskovic, M. S. Maucec, and V. Zumer, "Dynamic optimization using self-adaptive differential evolution," in *2009 IEEE Congress on Evolutionary Computation*, pp. 415–422, ISSN: 1941-0026.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," vol. 220, no. 4598, pp. 671–680, publisher: American Association for the Advancement of Science.

[21] F. Lezama, "HyDEDF_source," original-date: 2019-09-22T19:41:40Z. [Online]. Available: https://github.com/fernandolezama/HyDEDF_Source

[22] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: The best of both worlds," vol. 13, no. 2, pp. 31–39.

[23] M. Rodrigues, "Impact of dynamic contracts in power systems." [Online]. Available: https://fenix.tecnico.ulisboa.pt/cursos/meec/dissertacao/846778572212781