DARKO

Release 0.1

Matija Pavičević

Nov 22, 2021

CONTENTS:

DARKO is acronym for the Day-ahead Market Optimization package.

CHAPTER ONE

DESCRIPTION

DARKO is a small energy market analysis toolkit (mostly day ahead, but can also be multi day, weekly, monthly or annual). It is based in Python and solved in GAMS.

Main features:

- Demand orders
- Simple orders
- Block orders
- Flexible orders
- Storage orders
- NTC's (ramping limits: both hourly and per period)
- Net positions (ramping limits: both hourly and per period)

The main purpose of this package is simulation of energy markets with multiple players and interconnected zones

CHAPTER

TWO

GET INVOLVED

This project is an open-source project. Interested users are therefore invited to test, comment or contribute to the tool. Submitting issues is the best way to get in touch with the development team, which will address your comment, question, or development request in the best possible way. We are also looking for contributors to the main code, willing to contibute to its capabilities, computational-efficiency, formulation, etc. Finally, we are willing to collaborate with national agencies, reseach centers, or academic institutions on the use on the model for different data sets relative to EU countries.

CHAPTER THREE

MAIN DEVELOPERS

Currently the main developers of the DARKO package are the following:

- Matija Pavičević (KU Leuven, Belgium)

CHAPTER FOUR

CONTENTS

4.1 Overview

Version (0.1)

Date Nov 22, 2021

4.1.1 Features

- Demand orders
- Simple orders
- Block orders
- Flexible orders
- Storage orders
- NTC's (ramping limits: both hourly and per period)
- Net positions (ramping limits: both hourly and per period)

4.1.2 Libraries used and requirements

- Python 3.7
- pandas for input and result data handling
- matplotlib for plotting
- GAMS_api for the communication with GAMS

the above are auto installed in a conda environment if you follow the instructions of the Quick start.

4.1.3 DARKO in the scientific literature

- Micro-scale heat market[1].

4.1.4 Ongoing developments

The DARKO project is relatively recent, and a number of improvements will be brought to the project in a close future:

- Results analysis (outputs, plots…)

4.1.5 Licence

DARKO is a free software licensed under the “European Union Public Licence” EUPL v1.2. It can be redistributed and/or modified under the terms of this license.

4.1.6 Main Developers

- Matija Pavičević (KU Leuven, Belgium)

4.1.7 References

4.2 Model Description

The model is expressed as a MILP problem.

4.2.1 Variables

Sets

Name	Description
u	Units Supply Side
d	ProsumersConsumers Demand Side
o	Order type
n	Nodes
l	Lines
t	Technologies
tr(t)	Renewable technologies
f	Fuel types
s(u)	Storage Technologies
h	Hours
i(h)	Subset of simulated hours for one iteration
z(h)	Subset of all simulated hours
sk	Sectors

[1] Sebestyén, T.T. and Pavičević, M. and Dorotić, H. and Krajačić, G. (2020). The establishment of a micro-scale heat market using a biomass-fired district heating system. *Energy, Sustainability and Society*, doi:10.1186/s13705-020-00257-2

Parameters

Name	Units	Description
Availabilities		
AccaptanceBlockOrdersMin(u)	[%]	Accaptance ratio for block orders
AvailabilityFactorDemandOrder(d,h)	[%]	Share of maximum demand in time period h
AvailabilityFactorSimpleOrder(u,h)	[%]	Share of maximum Simple order in time period h
AvailabilityFactorBlockOrder(u,h)	[%]	Share of maximum Block order in time period h
AvailabilityFactorFlexibleOrder(u)	[%]	Accaptance ratio for block orders
Node data		
LocationDemandSide(d,n)	[n.a.]	Location {1 0}
LocationSupplySide(u,n)	[n.a.]	Location {1 0}
NodeHourlyRampUp(n, h)	[%hn]	Node ramp up limit
NodeHourlyRampDown(n, h)	[%hn]	Node ramp down limit
NodeDailyRampUp(n)	[%24hn]	Node daily ramp up limit
NodeDailyRampDown(n)	[%24hn]	Node daily ramp up limit
NodeInitial(n)		
Prices		
PriceDemandOrder(d,h)	[€MW]	Price ofer of the consumer d in time period h
PriceSimpleOrder(u,h)	[€MW]	Price ofer of the simple order u in time period h
PriceBlockOrder(u)	[€MW]	Default block order price
PriceFlexibleOrder(u)	[€MW]	Default block order price
Interconection lines		
LineNode(l,n)	[n.a.]	Incidence matrix {-1 +1}
FlowMaximum(l,h)	[MW]	Line limits
FlowMinimum(l,h)	[MW]	Minimum flow
LineHourlyRampUp(l, h)	[%hn]	Interconection line ramp up limit
LineHourlyRampDown(l, h)	[%hn]	Interconection line ramp down limit
LineDailyRampUp(l)	[%24hn]	Interconection daily line ramp up limit
LineDailyRampDown(l)	[%24hn]	Interconection daily line ramp down limit
LineInitial(l)		
Units / demands		
MaxDemand(d)	[MWu]	Maximum demand
PowerCapacity(u)	[MWu]	Installed capacity
OrderType(u,o)	[n.a.]	Order type {1 0}
Technology(u,t)	[n.a.]	Technology type {1 0}
Fuel(u,f)	[n.a.]	Fuel type {1 0}
Sector(d,sk)	[n.a.]	Demand sector type {1 0}
UnitRampUp(u)	[%hu]	Unit ramp up limit
UnitRampDown(u)	[%hu]	Unit ramp down limit
LinkedBlockOrderIncidenceMatrix(u)		
MinimumIncomeFixed(u)		
MinimumIncomeVariable(u)		
Storage		
StorageChargingCapacity(s)	[MWu]	Storage capacity
StorageChargingEfficiency(s)	[%]	Charging efficiency
StorageSelfDischarge(s)	[%day]	Self-discharge of the storage units
StorageCapacity(s)	[MWhu]	Storage capacity
StorageDischargeEfficiency(s)	[%]	Discharge efficiency
StorageOutflow(s,h)	[MWhu]	Storage outflows

continues on next page

Table 1 – continued from previous page

Name	Units	Description
StorageInflow(s,h)	[MWhu]	Storage inflows (potential energy)
StorageInitial(s)	[MWh]	Storage level before initial period
StorageProfile(s,h)	[%]	Storage level to be resepected at the end of each horizon
StorageMinimum(s)	[MWh]	Storage minimum
StorageFinalMin(s)	[MWh]	Minimum storage level at the end of the optimization horizon

NB: When the parameter is expressed per unit ("/u"), its value must be provided for one single unit.

Positive Optimization Variables

Name	Units	Description
AcceptanceRatioOfDemandOrders(d,h)	[%]	Acceptance ratio of demand orders
AcceptanceRatioOfSimpleOrders(u,h)	[%]	Acceptance ratio of simple orders
AcceptanceRatioOfBlockOrders(u)	[%]	Acceptance ratio of block orders
Flow(l,h)	[MW]	Flow through lines
StorageInput(s,h)	[MW]	Charging input for storage units
StorageOutput(s,h)	[MW]	Discharging output for storage units
StorageLevel(s,h)	[MWh]	Storage level of charge
spillage(s,h)	[MW]	Spillage from reservoirs
WaterSlack(s)	[MWh]	Unsatisfied water level constraint at end of optimization period
SystemCost(h)	[EUR]	Hourly system cost

Binary Variables

Name	Units	Description
ClearingStatusOfBlockOrder(u)	•	Binary variable
ClearingStatusOfFlexibleOrder(u,h)	•	Binary variable

Free Variables

Name	Units	Description
TotalWelfare	[EUR]	Total welfate
NetPositionOfBiddingArea(n,h)	[EUR]	Net position of bidding area
TemporaryNetPositionOfBiddingArea(n,h)	[EUR]	Temporary net position of bidding area
DailyNetPositionOfBiddingArea(n)	[EUR]	Daily net position of biding area

4.2.2 Optimisation model

Objective function

The goal of the day-ahead market problem is to maximize the total welfare.

$$maxTotalWelfare = \sum_i SystemCost_i - \sum_s StorageSlack_s \tag{4.1}$$

System costs

Hourly system costs are defined as follows:

$$\begin{aligned}
&SystemCost_i = \\
&\sum_d (AcceptanceRatioOfDemandOrders_{d,i} \cdot AvailabilityFactorDemandOrder_{d,i} \cdot \\
&MaxDemand_d \cdot PriceDemandOrder_{d,i}) \\
&- \sum_u (AcceptanceRatioOfSimpleOrders_{u,i} \cdot AvailabilityFactorSimpleOrder_{u,i} \cdot \\
&PowerCapacity_u \cdot PriceSimpleOrder_{u,i}) \\
&- \sum_u (AcceptanceRatioOfBlockOrders_{u,i} \cdot AvailabilityFactorBlockOrder_{u,i} \cdot \\
&PowerCapacity_u \cdot PriceBlockOrder_u) \\
&- \sum_u (AcceptanceRatioOfFlexibleOrders_{u,i} \cdot AvailabilityFactorFlexibleOrder_{u,i} \cdot \\
&PowerCapacity_u \cdot PriceFlexibleOrderOrder_u)
\end{aligned}$$

Power Balances

The main constraint to be met is the supply-demand balance, for each period and each zone, in the day-ahead market (equation). According to this restriction, the sum of all the power produced by all the units present in the node (including the power generated by the storage units), the power injected from neighbouring nodes is equal to the load in that node plus the power consumed for energy storage

Net position of each area:

$$
\begin{aligned}
&NetPositionOfBiddingArea_{n,i} = \\
&\sum_u (AcceptanceRatioOfSimpleOrders_{u,i} \cdot AvailabilityFactorSimpleOrder_{u,i} \cdot \\
&PowerCapacity_u \cdot LocationSupplySide_{u,n}) \\
&+ \sum_u (AcceptanceRatioOfBlockOrders_{u,i} \cdot AvailabilityFactorBlockOrder_{u,i} \cdot \\
&PowerCapacity_u \cdot LocationSupplySide_{u,n}) \\
&+ \sum_u (AcceptanceRatioOfFlexibleOrders_{u,i} \cdot AvailabilityFactorFlexibleOrder_{u,i} \cdot \\
&PowerCapacity_u \cdot LocationSupplySide_{u,n}) \\
&- \sum_d (AcceptanceRatioOfDemandOrders_{d,i} \cdot AvailabilityFactorDemandOrder_{d,i} \cdot \\
&MaxDemand_d \cdot LocationDemandSide_{d,n}) \\
&- \sum_s (StorageInput_{s,i} \cdot LocationSupplySide_{s,n}) \\
&+ \sum_s (StorageOutput_{s,i} \cdot LocationSupplySide_{s,n})
\end{aligned}
$$

Temporary net position due to flows between two neighbouring areas:

$$TemporaryNetPositionOfBiddingArea_{n,i} = -\sum_l (Flow(l,i) \cdot LineNode_{l,n})$$

Net position due to flows between two neighbouring areas:

$$
\begin{aligned}
&NetPositionOfBiddingArea_{n,i} - TemporaryNetPositionOfBiddingArea_{n,i} = \\
&-\sum_l (Flow_{l,i} \cdot LineNode_{l,n})
\end{aligned}
$$

Block orders

Lower and upper bounds on block orders:

$$
\begin{aligned}
AccaptanceBlockOrdersMin_u \cdot ClearingStatusOfBlockOrder_u \cdot OrderType_{u,"Block"} \leq \\
AcceptanceRatioOfBlockOrders_u
\end{aligned}
$$

$$
\begin{aligned}
AcceptanceRatioOfBlockOrders_u \leq \\
AccaptanceBlockOrdersMin_u \cdot ClearingStatusOfBlockOrder_u \cdot OrderType_{u,"Block"}
\end{aligned}
$$

Flexible orders

Limits of flexible orders

$$\sum_i (ClearingStatusOfFlexibleOrder_{u,i} \cdot OrderType_{u,"Flexible"}) \leq 1$$

Flow limits

Flows are above minimum values

$$FlowMinimum_{l,i} \leq Flow_{l,i}$$

Flows are below maximum values

$$Flow_{l,i} \leq FlowMaximum_{l,i}$$

Flows are within hourly ramping limits

$$Flow_{l,i} - Flow_{l,i-1} - LineInitial_{l,i=1} \leq LineHourlyRampUp_{l,i}$$

$$-Flow_{l,i} + Flow_{l,i-1} + LineInitial_{l,i=1} \leq LineHourlyRampDown_{l,i}$$

Flows are within daily ramping limits

$$\sum_i (Flow_{l,i}) \leq LineDailyRampUp_l$$

$$-\sum_i (Flow_{l,i}) \leq LineDailyRampDownl;$$

Net position limits

Net position is within hourly limits

$$\begin{aligned} NetPositionOfBiddingArea_{n,i} - NetPositionOfBiddingArea_{n,i-1} \\ \leq NodeHourlyRampUp_{n,i} \end{aligned}$$

$$\begin{aligned} -NetPositionOfBiddingArea_{n,i} + NetPositionOfBiddingArea_{n,i-1} \\ \leq NodeHourlyRampDown_{n,i} \end{aligned}$$

Net position is bounded by net position ramping limits

$$\sum_i (NetPositionOfBiddingArea_{n,i}) \leq NodeDailyRampUp_n$$

$$-\sum_i (NetPositionOfBiddingArea_{n,i}) \leq NodeDailyRampDown_n$$

Net position is within daily limits

$$DailyNetPositionOfBiddingArea_n \leq \sum_i (NetPositionOfBiddingArea_{n,i})$$

Ramping rates

Ramping rates are bound by maximum ramp up and down MW/min

$$\begin{aligned} & AcceptanceRatioOfSimpleOrders_{u,i} \cdot AvailabilityFactorSimpleOrder_{u,i} \\ & \cdot PowerCapacity_u \\ & - AcceptanceRatioOfSimpleOrders_{u,i-1} \cdot AvailabilityFactorSimpleOrder_{u,i-1} \\ & \cdot PowerCapacity_u \\ & \leq UnitRampUp_u \cdot PowerCapacity_u \end{aligned}$$

$$\begin{aligned}&AcceptanceRatioOfSimpleOrders_{u,i-1} \cdot AvailabilityFactorSimpleOrder_{u,i-1}\\&\cdot PowerCapacity_u\\&- AcceptanceRatioOfSimpleOrders_{u,i} \cdot AvailabilityFactorSimpleOrder_{u,i}\\&\cdot PowerCapacity_u\\&\leq UnitRampDown_u \cdot PowerCapacity_u\end{aligned}$$

Storage-related constraints

Generation units with energy storage capabilities (large hydro reservoirs, pumped hydro storage units, hydrogen storage units or batteries) must meet additional restrictions related to the amount of energy stored. Storage units are considered to be subject to the same constraints as non-storage power plants. In addition to those constraints, storage-specific restrictions are added for the set of storage units (i.e. a subset of all units). These restrictions include the storage capacity, inflow, outflow, charging, charging capacity, charge/discharge efficiencies, etc. Discharging is considered as the standard operation mode and is therefore linked to the Power variable, common to all units.

The first constraint imposes that the energy stored by a given unit is bounded by a minimum value:

$$StorageMinimum_s \leq StorageLevel_{s,i}$$

In the case of a storage unit, the availability factor applies to the charging/discharging power, but also to the storage capacity. The storage level is thus limited by:

$$StorageLevel_{s,i} \leq StorageCapacity_s$$

The energy added to the storage unit is limited by the charging capacity. Charging is allowed only if the unit is not producing (discharging) at the same time (i.e. if Committed, corresponding to the normal mode, is equal to 0).

$$StorageInput_{s,i} \leq StorageChargingCapacity_s$$

Discharge is limited by the level of charge of the storage unit:

$$\frac{StorageOutput_{s,i}}{StorageDischargeEfficiency_s} \leq StorageLevel_{s,i-1} + StorageInflow_{s,i}$$

It is worthwhile to note that StorageInflow and StorageOuflow must be multiplied by the number of units because they are defined for a single storage plant. On the contrary StorageLevel, Spillage and Power are defined for all units s. StorageInflow and Storage Outflow are predefined time series, whose meaning depends on the type of storage units: for hydro units, it is the natural water flows. For hydrogen units, StorageInflow is 0 at all times, but StorageOutflow represents the hydrogen demand (for fuel cell vehicles, industries,…). For batteries, both parameters are null at all times.

Charge is limited by the level of charge of the storage unit:

$$\begin{aligned}&StorageInput_{s,i} \cdot StorageChargingEfficiency_s\\&\leq StorageCapacity_s - StorageLevel_{s,i} + StorageOutflow_{s,i}\end{aligned}$$

Besides, the energy stored in a given period is given by the energy stored in the previous period, net of charges and discharges. This is storage balance equation:

$$\begin{aligned}&StorageLevel_{s,i-1} + StorageInflow_{s,i} + StorageInput_{s,i} \cdot StorageChargingEfficiency_s\\&= StorageLevel_{s,i} + StorageOutflow_{s,i} + Spillage_{wat,i} + \frac{StorageOutput_{s,i}}{StorageDischargeEfficiency_s}\end{aligned}$$

Some storage units are equiped with large reservoirs, whose capacity at full load might be longer than the optimisation horizon. Therefore, a minimum level constraint is required for the last hour of the optimisation, which otherwise would systematically tend to empty the reservoir as much a possible. An exogenous minimum profile is thus provided and the following constraint is applied:

$$StorageFinalMin_s \leq StorageLevel_{s,i} + StorageSlack_s$$

where N is the last period of the optimization horizon, StorageProfile is a non-dimensional minimum storage level provided as an exogenous input and StorageSlack is a variable defining the unsatified water level. The price associated to that water is very high.

4.2.3 Rolling Horizon

The mathematical problem described in the previous sections could in principle be solved for a whole year split into time steps, but with all likelihood the problem would become extremely demanding in computational terms when attempting to solve the model with a realistically sized dataset. Therefore, the problem is split into smaller optimization problems that are run recursively throughout the year.

The following figure shows an example of such approach, in which the optimization horizon is two days, including a look-ahead (or overlap) period of one day. The initial values of the optimization for day j are the final values of the optimization of the previous day. The look-ahead period is modelled to avoid issues related to the end of the optimization period such as emptying the hydro reservoirs, or starting low-cost but non-flexible power plants. In this case, the optimization is performed over 48 hours, but only the first 24 hours are conserved.

##.. image:: figures/rolling_horizon.png

The optimization horizon and overlap period can be adjusted by the user in the DARKO configuration file. As a rule of thumb, the optimization horizon plus the overlap period should at least be twice the maximum duration of the time-dependent constraints (e.g. the minimum up and down times). In terms of computational efficiency, small power systems can be simulated with longer optimization horizons, while larger systems should reduce this horizon, the minimum being one day.

4.2.4 References

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search