AI Acceleration for the Branch and Cut Optimization Algorithm

July 29, 2024

In this paper I discuss a method to optimize fleet decarbonization strategies and enable fleet owners to make informed decisions that align with their energy transition objectives and business outcomes.

I achieve this by first outlining an AI assisted approach to Mixed Integer Linear Programming (MILP) using data provided during the *Shell.ai Hackathon for Sustainable and Affordable Energy 2024*. The AI portion of the approach is called learning-to-configure-separators and is the result of research described in this MIT News article and this github article.

The key to the branch and cut algorithm is determining which solutions to eliminate. The solution space is represented by a tree and so, large portions of the solution space can be eliminated by eliminating entire branches.

The configure-separators algorithm accelerates the standard branch and cut algorithm by using a deep learning model to make better decisions about which branches to eliminate. Further acceleration is obtained by examination of possible solutions in parallel. Using a 96-processor machine with 4 GPUs, this algorithm is able to reduce the time to solve a problem by over 80% and also reducing carbon emissions created by running optimization solvers by a proportionate amount.

Detailed Problem Formulation:

The formal problem definition is described on the HackerEarth website and is copied in the Appendix.

In order to formulate the problem as a MILP, the decision variables were defined as all possible combinations of the following variables:

- veh_id: – defined by the ID column in the dataset defined in vehicles.csv
- purchase_year: – availability constrained according to vehicles.csv
- sell_year – for any given year up to 10 years after vehicle was purchased
- demand_year – for any year after vehicle was purchased and before it was sold.

- fuel_type – constrained according to vehicles_fuels.csv
- demand_distance – distance identifier constrained according to Constraints section of problem definition.
- size – constrained according to the problem definition to match demand.csv

The concatenation of the decision possibilities enumerates the possible decision space creating 43,688 variables.

Because purchase variables did not depend on fuel type, the problem was divided into two constraint categories, usage and purchase. Constraints were defined according to a format specification defined in the SCIOPT Documentation.

Solution Execution

Cost is defined according to the problem definition rules as the sum of **fuel cost**, **vehicle purchase cost** and **ownership costs** (insurance and maintenance less sales costs). The objective is to minimize costs without violating the constraints which are also defined in the supplied problem definition document.

Once the problem is formulated with an objective function, variable definitions and constraints, the problem is saved to a file and then the possible separator space is enumerated using the repository learning-to-configure-separators repository.

The steps of the solution are as follows:

Step	Desc.	File	Source Location	Input	Output
1	Define Linear Program	Shell_v0.2.ipynb	sign-of-fourier	Decisions, Objective Function and Constraints from HackerEarth problem statement	Python (SCIP/PuLP) Linear Program and LP Variables
2	Build Accelerator	find_best_in agnostic.py	L2config	Python defined Linear Program	Separators
3		explore subspace.py	L2config	Separators	Configured Separators
4		get_actions.py	L2config	Configured Separators	Training File
5	Train DL model	train.ipynb	sign-of-fourier	Single Configured Separator File	Trained DL model to estimate reward for separator configuration
6	Solve final problem with acceleration	solve.ipynb	sign-of-fourier	Trained DL model Original Problem Separators	Solved MILP

Steps 1 – 3 are defined using standard Python Linear Programming format. The output of this step is a standard Mixed Integer Linear Program.

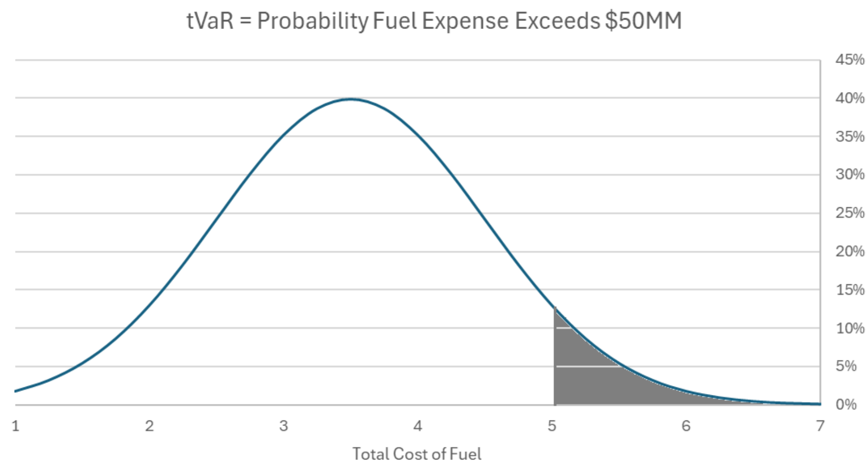
Steps 4 and 5 are performed using code provided by the learning to configure separators project. The output of step 4 is a sample of the configuration space of separators.

Fuel Uncertainty

Reliability Based Design Optimization (RBDO) is a technique in which reliability is a component of a optimization problem. Reliability of a proposed solution *s*, to an optimization problem is defined as

Reliability(s) = 1 – probability of failure

This equation can either be the objective function or added as one of the constraints. In financial risk management it is common to use a measure of tail risk of an asset to manage financial risk. For example, given a budget limit, a risk measure could be the amount that the fleet management strategy is expected to exceed a given limit.



The limit can be set according to actual market dynamics in order to establish a book value for the risk and to [manage the fuel price risk.]

$$E\left[B20_{t_i}B20_{t_j}\right] - E\left[B20_{t_i}\right]E\left[B20_{t_j}\right]$$

$$\int F_t \mathcal{N}\left(log(F_t), \mu_t, \sigma\sqrt{t}\right) dF_t$$

$$\mathcal{N}(x, \mu, \sigma)$$

Is the probability distribution function for the normal distribution for x with mean μ, and standard deviation σ.

An example of a constraint for B20 is

$$\sum_t E\left[B20_t 1_{\{B20_t > B_t\}}\right]$$

$$E[B20_t > B_t] = \int_{B_t}^{\infty} F_t \mathcal{N}\left(log(F_t), \mu_t, \sigma\sqrt{t}\right) dF_t$$

The risk of an instance is the sum of the reliability of the total fuel expenditure for the strategy. This is expressed as a sum and added as a constraint to linear program.

In RBDO, a measure of reliability is either added as a constraint or as the objective function. In the case of financial risk management, tail risk is calculated in much the same way as reliability and thus, the use of RBDO techniques is straightforward. Furthermore, because tail risk concepts are often used in the pricing of financial derivatives, it is straightforward to define a fair market value of this additional constraint and interpret the problem as an optimization of the tradeoff between risk and hedging costs.

Appendix

Introduction -

Welcome to the fifth edition of the Shell.ai Hackathon for Sustainable and Affordable Energy. Shell.ai Hackathon brings together brilliant minds passionate about digital solutions and AI, to tackle real energy challenges and help build a lower-carbon world where everyone can access and afford energy.In the previous four editions, we addressed some of the digital challenges around the energy transition: windfarm layout optimization (2020), irradiance forecasting for solar power generation (2021), optimal placement of electric vehicle (EV) charging stations (2022), and supply chain optimization for biorefineries (2023). This year, we focus on fleet decarbonization - a transition problem in the mobility sector.

Challenge -

Professional, delivery and operational fleets are a significant contributor to global greenhouse emissions. Fleet owners aspire to achieve net-zero emissions promptly; however, the transition presents a complex dilemma. Balancing the urgency of achieving net-zero emissions with business sustainability and customer satisfaction requires a decision-making framework that considers factors such as timing,location, and approach. In this hackathon, you will have a chance to develop mathematical models to optimize fleet decarbonization strategies, to help fleet owners make informed decisions that align with their energy transition objectives and business outcomes. By harnessing the power of data and mathematical models, you will navigate the complexities of demand forecasts, dissect emission profiles,and find ways to meet ambitious emission targets. The end game is to develop ingenious solutions that strike a balance between operational effectiveness and environmental impact.

Problem Statement -

Road transport is the backbone of supply chain, playing a pivotal role in moving goods and bolstering the economy. This mode of transport’s advantages are flexibility, door-to-door service, and connectivity between cities, towns, and villages. While it comes with convenience and advantage, professional, delivery and operational fleets are a significant contributor to global greenhouse emissions. Fleet owners aspire to achieve net-zero emissions promptly; however, the transition presents a complex dilemma. Balancing the urgency of achieving net-zero emissions with business sustainability and customer satisfaction requires a decision-making framework that considers factors such as timing, location, and approach. In this hackathon, you will have a chance to develop mathematical models to optimize fleet decarbonization strategies, to help fleet owners make informed decisions that align with their energy transition objectives and business outcomes. By harnessing the power of data and mathematical models, you will navigate the complexities of demand forecasts, dissect emission profiles, and find ways to meet ambitious emission targets. The end game is todevelop ingenious solutions that strike a balance between operational effectiveness and environmental impact.We will provide various yearly 'demand' data from a fleet operator that must be met. The demand data is further divided into various size and distance buckets which indicate what vehicle sizes should be used and how much distance per day they can cover. These are some additional constraints imposed on meeting the customer demand. We provide various vehicles from the following 3 drivetrains: Diesel, LNG, and BEV (Battery Electric Vehicle). For each of these vehicles, the cost, operational yearly range, distance bucket they can cover, and the vehicle ID (unique identifier which helps you reference them in your solution) is provided. We include the information on the fuel consumption by every model of the vehicle and the corresponding fuel types. Furthermore, we also include the cost for every fuel type along with the amount of carbon emissions by each of them, for every single year. Finally, you are also provided with the total carbon emission limits that must not be exceeded every year. All the data provided spans the years 2023 to 2038 (both years inclusive), for a total of 16 years

Your solution should provide an optimal fleet composition over the years, which meets all supply-chain demand and constraints while abiding by the carbon emission limits for every year and has the lowest overall cost possible. The data provided to you and the solution expected from you has been further explained in the next section.

Data description -

The dataset contains the following:

- **demand.csv**: This file gives you the total yearly distance demand (in kms) that needs to be satisfied with vehicles of size Sx (size bucket) which can travel at least a minimum of Dx (distance bucket) per day. For example, row 1 indicates that there is a yearly demand of 869181 km for the S1 sized vehicles which can travel at least a minimum of D1 distance bucket per day.
- **vehicles.csv**: This file gives you the vehicle ID (model), type of vehicle (drivetrain), size bucket, year in which you can purchase it, purchase cost, yearly range (in kms), and the daily maximum distance bucket it can travel.
- **vehicles_fuels.csv**: This file gives you the fuel consumption (unit of fuel consumed/km) for every vehicle ID using a certain type of fuel.
- **fuels.csv**: There are 5 fuel types, and for each, this table provides the carbon emission per unit fuel and the (median) cost per unit fuel across all the years. It also includes the uncertainty in the fuel cost.
- **carbon_emissions.csv**: Provides the total carbon emissions limits that should not be violated for every year. It is a decreasing profile over the years.
- **sample_submission.csv**: Provides sample format for submission

Notations -

$C_{total} \leftarrow$ Total cost of fleet ownership and operations across all the years.

$C_{buy}^{yr} \leftarrow$ Total cost of buying vehicles in year yr.

$C_{ins}^{yr} \leftarrow$ Total insurance cost incurred on the vehicles in the fleet for the year yr.

$C_{mnt}^{yr} \leftarrow$ Total maintenance cost incurred on the vehicles in the fleet for year yr.

$C_{fuel}^{yr} \leftarrow$ Total fuel cost incurred on the operating fleet in the year yr.

$C_{sell}^{yr} \leftarrow$ Amount received by selling some vehicles in the fleet in the year yr.

$V_{yr} \leftarrow$ Set of all vehicles purchased in the year yr.

$C_{v_{yr}} \leftarrow$ Purchase cost of a single vehicle with ID v_{yr}.

$N_{v_{yr}} \leftarrow$ Number of vehicles of ID v_{yr} that have been purchased.

$F_{yr} \leftarrow$ Fleet of vehicles in the year yr.

$C_{v_{yrp}} \leftarrow$ Cost of vehicle in fleet purchased in the year yrp. yrp is the year of purchase and yr is the year of operation such that $yrp \leq yr$.

$N_{v_{yrp}} \leftarrow$ Number of vehicles currently in the fleet in the year yrp.

$I_{(yr-yrp)}^{v_{yrp}} \leftarrow$ Insurance cost in the year yr for vehicle v_{yrp} purchased in the year yrp.

$M_{(yr-yrp)}^{v_{yrp}} \leftarrow$ Maintenance cost in the year yr for vehicle v_{yrp} purchased in the year yrp.

$D_{(yr-yrp)}^{v_{yrp}} \leftarrow$ Depreciation cost in the year yr for vehicle v_{yrp} purchased in the year yrp.

$U_{yr} \leftarrow$ Vehicles being used (driven) in the year yr. This is a subset of F_{yr}.

$F_v \leftarrow$ All fuel types applicable for vehicle v.

$Ds_v^f \leftarrow$ Distance travelled by vehicle v using fuel f.

$N_v^f \leftarrow$ Number of vehicles of type v driving fuel type f.

$m_v^f \leftarrow$ Fuel Consumption of vehicle type v driving with fuel type f.

$C_{uf,f}^{yr} \leftarrow$ Cost of unit fuel of type f in the year yr.

$N_{yr,v_{yrp}}^{sell} \leftarrow$ Number of vehicles v_{yrp} to be sold in the year yr.

$Carbon_{tot}^{yr} \leftarrow$ Total carbon emission in the year yr.

$CE^f \leftarrow$ Carbon emission for the fuel type f.

Objective -

$$C_{total} = \sum_{yr=2023}^{2038} C_{buy}^{yr} + C_{ins}^{yr} + C_{mnt}^{yr} + C_{fuel}^{yr} - C_{sell}^{yr}$$

$$C_{buy}^{yr} = \sum_{v_{yr} \epsilon V_{yr}} C_{v_{yr}} * N_{v_{yr}}$$

$$C_{ins}^{yr} = \sum_{v_{yrp} \epsilon F_{yr}} C_{v_{yrp}} * I_{(yr-yrp)}^{v_{yrp}} * N_{v_{yrp}}$$

$$C_{mnt}^{yr} = \sum_{v_{yrp} \epsilon F_{yr}} C_{v_{yrp}} * M_{(yr-yrp)}^{v_{yrp}} * N_{v_{yrp}}$$

$$C_{fuel}^{yr} = \sum_{v \epsilon U_{yr}} \sum_{f \epsilon F_v} Ds_v^f * N_v^f * m_v^f * C_{uf,f}^{yr}$$

$$C_{sell}^{yr} = \sum_{v_{yrp} \epsilon F_{yr}} C_{v_{yrp}} * D_{(yr-yrp)}^{v_{yrp}} * N_{yr,v_{yrp}}^{sell}$$

Your goal is to minimize C_{total} while abiding by the constraints detailed below.

Constraints -

- Vehicle of size Sx can only cater to the demand of size bucket Sx.
- Vehicle belonging to distance bucket Dx can satisfy all demands for distance bucket D1 to Dx. For example, vehicle belonging to distance bucket D4 can satisfy demand of D1, D2, D3, D4buckets; similarly, D3 can satisfy D1, D2, D3 but NOT D4.
- Total carbon emitted by fleet operations each year should be within the respective year's carbon emissions limits provided in carbon_emissions.csv. Total carbon emissions for a year is calculated using:

$$Carbon_{tot}^{yr} = \sum_{v \epsilon U_{yr}} \sum_{f \epsilon F_v} Ds_v^f * N_v^f * m_v^f * CE^f$$

- Total yearly demand for each year must be satisfied for each distance and size buckets.
- Vehicle model of year 20xx can only be bought in the year 20xx. For example, Diesel_S1_2026 can only be bought in 2026 and not in any subsequent or previous years.
- Every vehicle has a 10-year life and must be sold by the end of 10th year. For example, a vehicle bought in 2025 must be sold by the end of 2034.
- You cannot buy/sell a vehicle mid-year. All buy operations happen at the beginning of the year and all sell operations happen at the end of the year.
- Every year at most 20% of the vehicles in the existing fleet can be sold.

Evaluation -

There will be 2 rounds of evaluation which are as follows:

- Public
- Private

The score generated for the public round will be visible on the leaderboard till the first round is over.

You are expected to provide the solution in a .csv format file. The column names that should exist in the .csv along with "valid" entries are provided in the table below.

Columns	**Valid Entries**
Year	2023, 2024, …., 2038
ID	Should be among list of IDs provided in vehicles.csv
Num_Vehicles	>=1
Type	Should be among "Buy", "Use", "Sell".
Fuel	Should be among "Electricity", "B20", "LNG", "BioLNG", "HVO".
Distance_bucket	Should be among D1, D2, D3, D4.
Distance_per_vehicle(km)	Should >= 0 and <= Yearly range of that model.

Note: Distance bucket of the solution file corresponds to the distance bucket in the demand.csv file. Note that this is not the distance bucket of the vehicle itself (we get that from vehicles.csv anyways using the provided ID).

Additional Info -

- Vehicle resale value, insurance cost and maintenance costs as a percentage of its purchase cost is given below for each year after the purchase of vehicle.

End of Year	% of Purchase Cost		
	Resale Value (%)	Insurance Cost (%)	Maintenance Cost (%)
1	90%	5%	1%
2	80%	6%	3%
3	70%	7%	5%
4	60%	8%	7%
5	50%	9%	9%
6	40%	10%	11%
7	30%	11%	13%
8	30%	12%	15%
9	30%	13%	17%
10	30%	14%	19%

To illustrate the calculations, let us take an example of purchase cost = $100 for a vehicle bought on Jan 1st , 2025. Using the percentages in the above table, the values can be calculated as follows:

Year of Operation	End of Year	Resale Date	Resale Value (RV) ($)	Insurance Cost (IC) ($)	Maintenance Cost (MC) ($)	Insurance & Maintenance Period
2025	1	31st Dec, 2025	90	5	1	1st Jan – 31st Dec, 2025
2026	2	31st Dec, 2026	80	6	3	1st Jan – 31st Dec, 2026
2027	3	31st Dec, 2027	70	7	5	1st Jan – 31st Dec, 2027
2028	4	31st Dec, 2028	60	8	7	1st Jan – 31st Dec, 2028
2029	5	31st Dec, 2029	50	9	9	1st Jan – 31st Dec, 2029
2030	6	31st Dec, 2030	40	10	11	1st Jan – 31st Dec, 2030
2031	7	31st Dec, 2031	30	11	13	1st Jan – 31st Dec, 2031
2032	8	31st Dec, 2032	30	12	15	1st Jan – 31st Dec, 2032
2033	9	31st Dec, 2033	30	13	17	1st Jan – 31st Dec, 2033
2034	10	31st Dec, 2034	30	14	19	1st Jan – 31st Dec, 2034

- Distance bucket mappings -

Distance bucket	Name of category
Up to 300 km daily	D1
Up to 400 km daily	D2
Up to 500 km daily	D3
Up to 600 km daily	D4

- Vehicle size bucket mappings -

Vehicle size	Name of category
17 tons	S1
44 tons	S2
50 tons	S3
64 tons	S4

- Please note that insurance and maintenance costs should be calculated for all vehicles in the fleet (irrespective of whether they are used in a particular year), whereas fuel costs are only for those vehicles that will be used (driven) in that year.

Scoring -

If your solution satisfies all constraints, we will first calculate your solution's total cost of fleet ownership and operations using the overall cost function. Your cost (the lower the better) will then be converted to a score (the higher the better) between 30 to 100 using the following transformation function for the leaderboard ranking:

$leaderboard\ score = max[30,(100-70 * costreference\ cost)]$

Scores between 0 to 26 are reserved for the error codes detailed below.

As you can see from the fuels.csv, there is a median fuel cost along with an uncertainty band that has been provided. The final cost (which will also comprise of fuel costs), will be a statistical aggregate over 1000 randomly drawn samples from the provided fuel distribution. The public leaderboard is shown to you during the entire duration of the competition. Here, you will be evaluated until the year 2028. The private leaderboard will not be shown to you and will only be revealed at the end of the competition. This is evaluated for the full course of 16 years from 2023 to 2038. The reference cost for private leaderboard is 172,000,000, and for public leaderboard it is 65,000,000.

Note: Error scores - if the candidate gets any integer score between 0-26, it means it encountered an error during evaluation.

0 - Any unaccounted error code

1 - Year entry must be of type integer.

2 - ID entry must be a string.

3 - Num_Vehicles entry must be of type integer.

4 - Type entry must be a string.

5 - Fuel entry must be a string.

6 - Distance_bucket entry must be a string.

7 - Distance_driven_per_vehicle(km) must be of type float.

8 - Year entry must be >=2023 and <=2038.

9 - ID must be from the given list of vehicle IDs.

10 - Num_Vehicles must be > 0.

11 - Type must take values among “Buy”, “Sell”, “Use”.

12 - Fuel must be among “Electricity”, “LNG”, “BioLNG”, “HVO”, “B20”.

13 - Distance bucket must be among “D1”, “D2”, “D3”, “D4”.

14 - Distance driven must be >= 0 and <= Yearly range for that model.

15 - Constraint 2 violations.

16 - Constraint 5 violation. Vehicle bought in year YYYY should have YYYY in its ID.

17 - Constraint 6 violation. Vehicle purchased in YYYY can only be used for 10 years.

18 - Demand.csv has the demand pertaining to 16 combinations (S1_D1, S1_D2, ..., S4_D4) for each year. Your solution.csv file should also have all these combinations present.

19 - Constraint 4 violation. For example, in demand.csv for year 2023 S1_D1 demand is 869181 km. In the solutions you provide, sum of all distance travelled in the year 2023 by vehicles with S1 size which satisfy D1 distance demand must be >= 869181 km

20 - You should use the right type of fuel for a given vehicle.

21 - Constraint 8 violation. You must sell only a max of 20% of the fleet every year and nothing more.

22 - You can only "Use" vehicle IDs that you have in fleet.

23 - You can only "Use" as many vehicles that you have in fleet for each ID.

24 - You can only "Sell" vehicle IDs that you have in fleet.

25 - You can only "Sell" as many vehicles that you have in fleet for each ID.

26 - Constraint 3 violation. Total Carbon emission by fleet operations <= Carbon budget for that year.

You do not need to submit your source code files. When you submit your solution, you can ignore the "Upload source file" field.

Arthur Charpentier. An introduction to multivariate and dynamic risk measures. 2018. ffhal-01831481f

Tomer Shushi, Jing Yao, Multivariate risk measures based on conditional expectation and systemic risk for Exponential Dispersion Models, Insurance: Mathematics and Economics, Volume 93, 2020, Pages 178-186, ISSN 0167-668

Cai, J., Jia, H., & Mao, T. (2021). A multivariate CVaR risk measure from the perspective of portfolio risk management. *Scandinavian Actuarial Journal*, *2022*(3), 189–215. https://doi.org/10.1080/03461238.2021.1944905