

The dynamic electric boat charging problem

Camilo Vélez^{a*} , Alejandro Montoya^a 

^aUniversidad EAFIT, Departamento de Ingeniería de Producción, Medellín, Colombia

*cvelezg10@eafit.edu.co

Abstract

Paper aims: This paper proposes a new optimization problem named the dynamic electric boat charging problem (DEBCP).

Originality: The problem dynamically determines the speed and charging decisions of an electric boat (EB) to be performed in photovoltaic charging stations (PVCSS).

Research method: The objective function is to minimize the charging plus the battery degradation costs. Considering that the energy consumption of the EB and the solar irradiance for the PVCSS are uncertain due to factors external to the operation, the dynamic component of the DEBCP recalculates the solution to the problem as new information related to these variables is known. To solve the problem, we propose a rolling horizon genetic algorithm. This method constantly reevaluates the speed and charging decisions of the operation. Such decisions are made with a genetic algorithm.

Main findings: Our results show that the recalculations help either reducing the cost of the solution when possible or correcting the operation when needed.

Implications for theory and practice: To evaluate our solution method, we built some test instances based on a future fluvial transport operation with an EB that will be implemented in Colombia. We assess the impact of the dynamic recalculations by comparing the results of the DEBCP problem to a scenario following a static solution.

Keywords

Electric boat. Speed decisions. Charging decisions. Photovoltaic charging stations. Dynamic recalculations.

How to cite this article: Vélez, C., & Montoya, A. (2023). The dynamic electric boat charging problem. *Production, 33*, e20220114. <https://doi.org/10.1590/0103-6513.20220114>

Received: Nov. 3, 2022; Accepted: Sep. 11, 2023.

1. Introduction

The electric vehicle (EV) market has been constantly growing (and is expected to continue doing so) due to some environmental policies and the acceptance from industries and consumers (International Energy Agency, 2020). However, EVs face certain technical challenges such as having overall less autonomy than internal combustion vehicles, lengthy charging times and expensive batteries. Even though the cost of these batteries has been decreasing with the years, it can still account for around 30% of the total cost of an EV (Bullard, 2019). This issue takes even more importance when considering another problem of such batteries, their degradation. Battery degradation is a phenomenon in which a battery's maximum capacity decreases due to its charging cycles and its storing conditions (Barré et al., 2013). Transport operations with EVs must account for these limitations to be technically feasible and economically viable.

Most EVs operate in urban areas where their technical limitations are less impactful than in rural ones. Additionally, urban areas tend to have more reliable energy grids than rural ones (Krupp, 2010), which makes it easier for EVs to be charged. However, recent technological advances such as improved battery energy densities (Choi & Aurbach, 2016) (which increase the driving range of EVs) or the use of Photovoltaic Charging Stations (PVCSS) allow EVs to operate in rural areas. PVCSS as their name implies are charging stations that use photovoltaic energy during their operation (Shepero et al., 2020). This energy may supplement the one from the electric grid to help mitigating the reliability issues on rural areas. One interesting application of electric transportation in rural areas is the fluvial one.



This type of transportation is crucial in certain regions where terrestrial transportation is infeasible, such as certain areas of the Amazonía (Jaimurzina et al., 2017). Some electric fluvial operations have been implemented in Latin America in recent years. Jaimurzina et al. (2017) evaluated the feasibility of using Electric Boats (EBs) for passenger transportation in the Putumayo River in routes of up to 60 km. Project Kara Solar connected multiple indigenous communities in Ecuador using a solar powered EB and aim at installing another one in Peru (Giménez, 2017).

Given that some electric fluvial operations have been implemented in the past couple of years, optimization problems in transportation regarding EBs have recently been researched in the literature. These problems can be faced from two different perspectives, a strategic and an operational one. Strategic problems determine the required infrastructure to support electric fluvial operations, making decisions such as the location and sizing of charging stations or the battery capacity of an EB. From this perspective, Zhang et al. (2017) used a mixed integer programming formulation to determine the location and service capacity of some charging stations to supply a given demand of EBs. Considering PVCs, Villa et al. (2020) used a mixed integer linear programming formulation to install non-grid connected PVCs to supply an EB during a given route, while also selecting the battery capacity of the boat. Solving a similar problem, Vélez et al. (2020) used a constructive heuristic to also select the battery capacity of an EB and locate the same type of PVCs, while also sizing their photovoltaic (PV) components. Expanding on these works, Vélez & Montoya (2023) used a two-stage simulation-based branch-and-bound algorithm to determine the battery capacity of an EB, the location and sizing of PVCs and non-PV charging stations. Their objective function was the sum of the investment costs plus the ones needed to operate the system in a simulation of multiple years-worth of the EB's operation. For that simulation, the authors considered the number of passengers, the solar irradiance and the possibility of having power outages as stochastic variables. Operational problems determine how to operate an EB regarding decisions like its traveling speed, charging decisions or the scheduling of the operation, given an existing infrastructure. To our best knowledge, the first work to consider this perspective was Villa et al. (2019), who proposed the Electric Boat Charging Problem (EBCP). This problem determines the average speeds for an EB when traversing a set of segments in which a fluvial route is divided and the charging decisions of the EB. The charging decisions consist of where and when to charge, how much energy to charge, and which charging power to use. The objective function of the problem is to minimize the charging and the EB's battery degradation costs. For the battery degradation, only the component related to the cycling of the battery is considered, as it is the one that depends on the EB's operation. The EBCP solves a static scenario of the fluvial operation before the beginning of such operation. One of the main parameters of the problem is an estimation of the energy consumption of the EB as a function of its speed. However, the actual energy consumption that an EB experiences can be significantly affected by external factors that are hard to predict, such as waves generated by other boats or sediment in the water. Additionally, the authors considered only non-PV charging stations. However, given that fluvial transport operations are usually performed in rural areas where electric grids may be unreliable, PVCs are more common and applicable in these areas. Due to these two reasons, the solution of the EBCP may not be properly translated to the actual operation of an EB.

To face the aforementioned issues of the EBCP, in this research we propose the Dynamic Electric Boat Charging Problem (DEBCP). This problem dynamically reevaluates its solution, updating the energy level of the EB's battery as new information of the energy consumption is known. Additionally, the problem considers grid connected PVCs to charge the EB. As the solar irradiance is a stochastic variable due to some weather phenomena, we also reevaluate the solution as new information of such irradiance is available. For this problem we consider the same objective function of the EBCP, to minimize the charging and battery degradation costs. Our charging cost is given by the amount of energy to purchase from the grid. A feasible solution to the problem must meet the energy autonomy of the EB and comply with a set of time windows for the EB to depart from certain nodes of the fluvial route and a maximum route time. For this problem we considered that such route is a round trip. To solve this problem, we propose a rolling horizon genetic algorithm. The rolling horizon algorithm dynamically recalculates the solution of the DEBCP and the Genetic Algorithm (GA) makes the speed and charging decisions. For the charging decisions, the GA uses an embedded heuristic method with some charging policies. We perform some computational experiments to evaluate our solution method. Considering that the speed and charging decisions are made by the GA we first compare its performance against a Mixed Integer Linear Programming (MILP) formulation based on that of Villa et al. (2019). The results show that the GA provides competitive solutions. Then, we assess the impact of the recalculations of the solution in the DEBCP by comparing it to a scenario where a static solution is followed. Results show that such recalculations can either reduce the cost of the solution when possible or correct the operation when required.

The remainder of the paper is organized as follows. Section 2 presents a literature review of optimization problems with dynamic decisions in transportation topics as well as problems that make charging and speed decisions for EVs. Section 3 describes some important technical aspects of the problem. Section 4 formally introduces the DEBCP. Section 5 presents our solution method. Section 6 shows the computational experiments. Finally, Section 7 concludes the paper.

2. Literature review

To our best knowledge, operational optimization problems with EBs are relatively new. Because of that, this literature review focuses on transportation problems that consider components of the DEBCP, regardless of which type of vehicle the problem considers. Such components are the dynamic part of the problem, the charging, and the speeds decisions.

Dynamic transportation problems have been widely studied in the literature. For example, Pillac et al. (2013) made a review of dynamic vehicle routing problems. The authors mentioned four different perspectives to face the dynamic component of the problems: periodic or continuous reoptimization, stochastic modeling and sampling. A common approach to perform periodic reoptimization is using a rolling horizon algorithm, which is the approach we took on this research. These algorithms divide a problem into multiple time slots which are each solved separately using some information taken from previous time slots or future ones via predictions (Bischi et al., 2019). The following works used a rolling horizon algorithm to face the dynamic nature of their problems. For a dynamic green bicycle repositioning problem, Shui & Szeto (2018) used a hybrid algorithm of rolling horizon and artificial bee colony to minimize a penalty cost related the unmet demand of bicycles, the fueling and emission costs of the vehicle that relocates them. The authors define a set of time slots for the operation, solving a scenario with updated parameters in each time slot. For a bus dispatching problem, Gkiotsalitis & Van Berkum (2020) used a rolling horizon algorithm with a novel nonlinear formulation to minimize the variance of the headway for a set of bus routes prioritized by their number of passengers. The authors updated the solution of the problem each time a new route is about to be dispatched. Recently, Kumar & Khani (2021) used a rolling horizon with an optimal matching algorithm to maximize either the total number of matches or the vehicle-hours savings for a ridesharing and scheduled-based transit system. To save computational time, the authors run their algorithm only for new riders and drivers during each time slot.

One of the main characteristics of most transportation problems with EVs is the inclusion of charging decisions. These decisions may include when and where to charge, how much energy to charge or which charging power to use. A review of optimization problems in the topic of charging EVs was made by Rahman et al. (2016). The authors reviewed works with different types of objective functions such as charging station installations costs from a strategic perspective as well as the charging costs, life cycle costs of the stations, average energy level of the EVs being charged or maximizing the use of renewable energies from an operational perspective. The authors emphasize the use of renewable energy sources for future optimization problems in EV charging topics. Considering the large number of works in the topic of optimization of EV charging processes and the importance of renewable energy sources, we focused this topic of the literature review in problems that use PVCSSs, as we do in the present work. Aiming to minimize the charging costs of some groups of EVs, Seddig et al. (2017) used three different optimization methods to reschedule the charging operations of EVs with PVCSSs. They considered that the cost of the energy to purchase from the grid was hour-dependent and different available hours in which each EV could be rescheduled. Recently, Li et al. (2020) used a chance constrained programming method to also minimize the cost of using PVCSSs to charge EVs. The authors considered different charging strategies, such as using energy storage systems and selling energy back to the grid via PV generation or vehicle to grid. Accounting for the battery degradation of the EVs, Wu et al. (2020) used a Markov decision process to minimize the operational costs of PVCSS. Such costs yet again included selling energy to the grid from the PV generation or from vehicle to grid. The authors additionally considered a compensation cost for the EVs due to the degradation of their batteries when using the vehicle to grid alternative and a penalization cost for not properly satisfying the charging demands.

The speed of an EV plays a significant role on its energy consumption, which is why selecting such speed is a crucial decision in certain operational problems. In one of such problems, Betancur et al. (2017) used a GA to optimize the speeds of a PV powered EV aiming to minimize the time it took to finish a race. Such race followed a fixed route which the authors divided into multiple segments, selecting the speed with which to traverse each of them. Probably the most similar research to ours is the EBCP of Villa et al. (2019). As mentioned before, the authors selected the speeds and charging decisions of an EB aiming to minimize the recharging and battery degradation costs using a MILP formulation. In a similar work to Villa et al. (2019) and our current research, Deschênes et al. (2020) used a MILP formulation to select the speeds and charging decisions of an EV during a fixed route. However, the authors did not consider the battery degradation or the use of different charging powers. They did however consider that the charging stations are not part of the fixed route and therefore required detours to be visited.

As previously mentioned, our research extends the EBCP proposed by Villa et al. (2019) by adding the dynamic recalculation of the solution, time windows for the operation and the use of PVCSs. We consider that these additions allow the problem to respond to the inherent uncertainty of some estimated parameters of the operation while also reducing the cost of the solution and providing a more detailed schedule of the route timewise.

3. Technical aspects

To provide a better explanation of the DEBCP, in this section we perform a brief explanation of some technical aspects related to the problem. These aspects are the EB's energy consumption, the PVCSs, how we estimated the solar irradiance, the way the battery degradation is modeled and the nonlinear charging function of electric batteries.

3.1. Energy consumption

The limited autonomy of EVs when compared to internal combustion vehicles is more impactful for EBs than for electric cars as the former consume around 12 to 18 times more energy to traverse the same distance (Candela, 2021). The energy consumption of an EB depends on different variables, two of the most important being the speed and weight of the vehicle (Minami & Yamachika, 2004). The weight of the EB in this research is a parameter composed by the weight of the hull, the equipment of the EB (including its battery) and the number of passengers currently aboard it. In the present work only the number of passengers may vary during the operation, but it is a known parameter. On the other hand, the speed of the EB is a decision variable of ours. This speed is the one measured relative to the water. For the actual speed of the EB relative to the ground, the river current is either added or subtracted depending on the flow of water. The consumption rate of the EB at a given speed remains the same while traveling in favor or against the current. However, as the speed of the EB is higher when traveling in favor of the river current, the energy consumption would be lower due to shorter travel times.

Some transportation problems with EVs require an estimation of the energy consumption of the vehicles to be performed (Villa & Montoya, 2018). For the DEBCP we require such estimation to be in terms on the weight and speed of the EB. There are multiple energy consumption models for boats in the literature. The model to select depends on whether the boat has a displacement or planing hull. The former type always displaces the same amount of water to float. The latter generates a lift force at certain speeds which decreases the displaced water and by extension the generated drag (Molland et al., 2017). The EB that we considered for our experiments had a planing hull. Therefore, we used the model of Savitsky (1964) to estimate the EB's energy consumption. However, as such consumption is a parameter for the DEBCP, any other consumption model that depends on the speed and weight of the vehicle can be implemented in a future research.

3.2. Photovoltaic charging stations

As previously mentioned, PVCSs are an interesting addition to electric fluvial transport operations in rural areas where the electric grid is not reliable. This is because PVCSs can either supplement or entirely replace the energy from the grid with their own PV generation. The main components of a PVCS are its charging plug, PV panels, either a DC-DC converter or a DC-AC inverter and an optional battery which is often called energy storage system. Just like with every other type of charging station, PVCSs are divided into three levels given by their charging powers, with level 1 stations having the lowest powers and level 3 the highest ones (Forbes, 2021). As EBs need relatively large batteries due to their energy consumption, in this research we focused only on level 3 PVCSs. All level 3 charging stations perform their charging in DC, therefore the following explanation reflects that. For the PV components, the PV panels are grouped up in arrays which are then connected to DC-DC converters. The output voltage of the PV arrays must match the charging voltage of the EV, something which can be achieved by either having large PV arrays, or by connecting them in series as in Macellari et al. (2013). Energy storage systems are sometimes used in PVCSs to store any surplus from the PV generation, which is crucial for non-grid connected PVCSs. However, it is not necessary for grid connected PVCSs to have such components, specially considering that they require a large initial investment and maintenance (Bhatti et al., 2016). Given that the PVCSs in the present work are indeed grid connected, we did not consider energy storage systems. For the energy flow, PVCSs prioritize using their own PV generation, buying energy from the grid only when required to supply the current energy demand of the EB. If the PV generation is large enough to supply the current charging demand at a given time, no energy is purchased from the grid. However, depending on the charging power, a great number of PV panels would be required for doing so.

3.3. Solar irradiance

Solar irradiance (which is the input energy for PV generation) is the amount of light energy from the sun that hits a square meter of the Earth each second (National Aeronautics and Space Administration, 2009). The fraction of solar irradiance that reaches the Earth's surface is stochastic due to weather related variables such as cloudiness (Ramakrishna & Scaglione, 2016). Because of that, such irradiance is often forecasted, which is what we do in this research. For doing so, we used a bidirectional long short-term memory artificial neural network implemented by Singh (2020). This is a type of memory based neural network that is trained using data presented both in a forward and reversed sequence to consider past and future information for each datum. Bidirectional long short-term memory artificial neural networks have given good results in the literature when forecasting weather variables such as solar irradiance (Patel et al., 2018). The training data for the neural network were taken from the National Solar Radiation Database (2023).

3.4. Battery degradation

As previously mentioned, battery degradation refers to a decrease in a battery maximum capacity. The literature divides battery degradation into two components referred to as calendar and cycling aging Barré et al. (2013). The former refers to the degradation given by the battery's storage condition while the latter is due to its charging cycles. We only considered the cycling aging on this paper as it is the one that depends on the operation of the EB.

To include the battery degradation in our model, we implemented an approach similar to the one used by Villa et al. (2019). In that work the battery degradation depends on two components: the depth of discharge, and the charging power. The first component is explained in detail by Pelletier et al. (2017). They propose an approach to cost the battery degradation based on an experimental study by Han et al. (2014). The latter studied the number of charging cycles that a battery is able to perform for different depths of discharges. Then, the authors estimated a wear cost function per unit of energy either charged or discharged from the battery while its energy level is in between a set of intervals. Each interval is given a different degradation cost. This approach assumes that charging operations are performed using a domestic charger (i.e. it does not account for multiple charging powers).

In the topic of degradation due to different charging powers, Omar et al. (2014) studied the impact of using higher charging powers in a battery's charging cycles when compared to a domestic one. To account for both the impact of the depth of discharge as well as the charging powers, Villa et al. (2019) penalizes the wear cost function proposed by Han et al. (2014) for each different charging power based on the findings of Omar et al. (2014). That is, different parameters of the wear cost function are considered for each charging power. As stated by Villa et al. (2019), this is a limited approach as it mixes two independent battery degradation studies. Nonetheless, our intention is simply to account for the impact of using different charging powers.

Figure 1 shows an example of a wear cost function for 16 kWh and 4,800 USD battery charged with either a 11 kW or 44 kW power. 10 intervals were selected, having each an equal length of 1.6 kWh. Values in the graph represent the degradation costs of charging 1 kWh with either of the charging powers when the current energy level of the battery falls within such intervals.

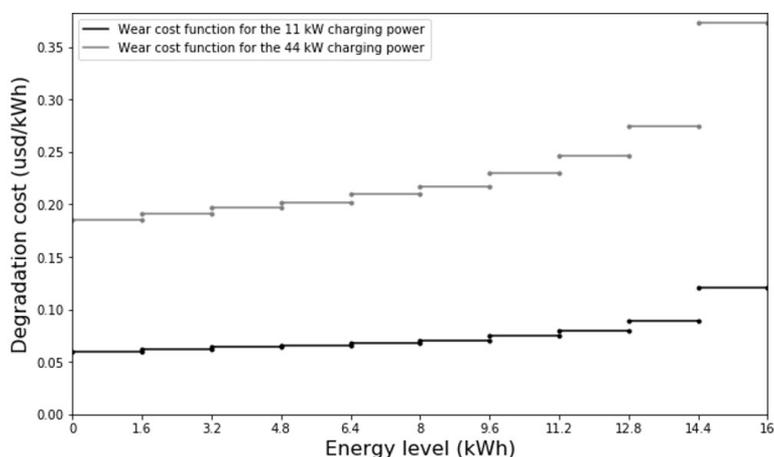


Figure 1. Wear cost function for a 16 kWh battery costing 4, 800USD charged with 11 kW and 44 kW charging powers.

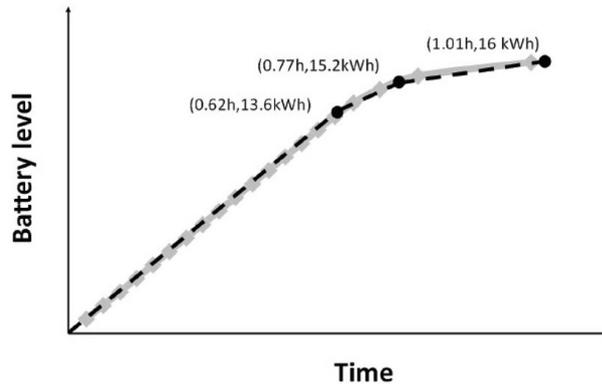


Figure 2. 16 kWh battery charged with a 22 kW power. Taken from Villa et al. (2019).

3.5. Nonlinear charging function

Charging functions for electric batteries model the relation between the charging time and energy level during a charging operation. To prevent such batteries from degrading due to overcharging, their charging process follows a nonlinear function called as constant current - constant voltage. During the constant current phase, the charging power is kept constant causing the battery's energy level to increase in a linear way with respect to time. This phase finishes when the battery's energy level is around 80% of its total capacity. Then, during the constant voltage phase the charging power decreases in an exponential way causing the energy level to increase following a decreasing curve with respect to time (Pelletier et al., 2015). In this research we modeled this charging process using the same approach as Montoya et al. (2017), which consists in approximating it via a piecewise linear function. Figure 2 shows an example of such approximation for a 16 kWh battery charged with a 22 kW power.

4. Problem description

Let $s = \{0, \dots, n\}$ be the set of segments in which a round trip is divided, with 0 being the segment in which the outward trip starts and n the one in which it finishes. Let also S_p be a subset of S having the segments with a port at the end of them where passengers may board or disembark from the EB. Each port also has a PVCS where the EB may charge. A PVCS located at the end of segment i during the outward trip is also located at the end of segment $n - (i + 1)$ during the return one. The EB is allowed to charge at such PVCS during both trips. We consider that the EB starts the round trip fully charged, and given that set S_p contains the segments with a PVCS at the end of them, we note that $n \notin S_p$. To properly plan the operation, we consider a booking system for passengers to buy their tickets. Because of that, the number of passengers aboard the EB at all times is known in advance. As previously mentioned, this number of passengers plays a significant role in the energy consumption of the EB. To have a proper quality service towards the passengers in relation to the time of their bookings, let l_s and u_s be the lower and upper time window in between which the EB must depart from the port at the end of segment $s \in S_p$. Additionally, let T_{max} be the maximum time in which the EB must complete the round trip. Let then p_s be the number of passengers aboard the EB when travelling towards the node at the end of segment $s \in S_p \cup \{n\}$. Let V be the set of speeds to consider for the EB. Each segment $s \in S$ has an associated travel time t_{sv} and real energy consumption $e(s, v, p_s)$ when traversing at speed $v \in V$ with p_s passengers. However, such energy consumption is uncertain before the start of the fluvial operation due to external factors such as the presence of other boats or sediment in the water. For that reason, let $\bar{e}(s, v, p_s)$ be the estimated energy consumption of the EB when traversing the segment $s \in S$ with a speed $v \in V$ and p_s passengers. For the operation of the PVCSs, let H be the set of charging powers to consider. Let also f_{sh} be a binary parameter equal to 1 if the PVCS at the end of segment $s \in S_p$ has access to charging power $h \in H$. We considered that PVCSs have the same owner of the EB, so there are not different energy tariffs based on the charging powers or the location of each PVCS, but rather a unified cost of buying energy from the grid. For the PV generation, let α_s be the number of PV panels of the PVCS at the end of segment $s \in S_p$. Let T be a set of time intervals of a day and g_j the real solar irradiance during time interval $j \in T$. Similarly to $e(s, v, p_s)$, g_j is also uncertain due to external factors. Therefore, let \bar{g}_j be the estimated solar irradiance during interval $j \in T$.

The EB has a battery with capacity Q . For the degradation of such battery, let M be a set of energy intervals in which to divide the EB's battery (e.g. $M = \{0kWh, 5kWh, 5kWh - 10kWh, \dots\}$). Let d_m be the battery degradation cost when the EB's battery is being discharged and its energy level falls in interval $m \in M$. Finally let c_{mh} be the battery degradation cost when the EB's battery is being charged with power $h \in H$ and its energy level is within interval $m \in M$.

The goal of the DEBCP is to dynamically determine the speed and charging decisions (where and when to charge, how much energy to charge and which power to use) of an EB performing a round trip. The dynamic component refers to recalculating the solution to the problem as new information becomes available during the operation. The objective function is to minimize the sum of the cost of the energy to purchase from the grid while charging the EB plus its battery degradation costs. For a solution to be feasible, the EB must comply with its time windows and maximum route time while meeting its energy autonomy. For the latter, the energy level of the battery must always be equal to or greater than a minimum value, both for safety and for battery health reasons (Pelletier et al., 2018). Lastly, if the reader wants to see a MILP formulation of the problem (without the dynamic recalculations of the solution), please refer to the one by Villa et al. (2019).

5. Rolling horizon genetic algorithm

We use a rolling horizon algorithm to face the dynamic component of DEBCP by recalculating the solution of the problem when an event is triggered. Events are triggered when a significant deviation between the real and expected values of the energy consumption, or the solar irradiance is experienced or when the EB arrives at a PVCS. The solution is recalculated using a GA that determines the speeds of the EB with some embedded charging policies making the charging decisions. Then, such policies allow the solution to be evaluated in terms of its feasibility and its cost to be calculated. We used a GA to recalculate the solution DEBCP as it allows such calculations to be performed with low computational times. We selected this metaheuristic as it has given good results in other works that determine the travelling speeds of EVs in the literature. For example, Saini et al. (2016) used a GA for a gear shift strategy for EVs aiming to minimize the consumption and the difference between the desired and actual speed of the vehicles. As mentioned in the literature review, Betancur et al. (2017) used a GA to optimize the speeds of an electric car aiming to minimize its traveling time.

5.1. Rolling horizon algorithm

For our rolling horizon algorithm, we divide the EB's operation in multiple time slots. During each of them, the current energy level of the EB, ϕ , and the estimation of the solar irradiance, \bar{g}_j , are updated and a new solution for the remaining part of the round trip is calculated using the GA. The time slots are not evenly spaced in time as they are triggered by three different events, named θ_1, θ_2 and θ_3 , which are not always synchronous. Event θ_1 is triggered every time the EB arrives at a PVCS. Event θ_2 is triggered when a significant deviation between the most recent measured values of $e(s, v, p_s)$ and $\bar{e}(s, v, p_s)$ is experienced. We compare the moving averages and not the actual values of the variables to look for significant differences rather than outlier values. For that reason, let μ_e and $\mu_{\bar{e}}$ be the moving averages of the last b_2 values of $e(s, v, p_s)$ and $\bar{e}(s, v, p_s)$. Event θ_2 is triggered if $\frac{|\mu_e - \mu_{\bar{e}}|}{\mu_e} > r_2$, with r_2 been a given percentage value to determine if the difference is significant. This comparison is performed every o_2 units of time, resetting the timer when the EB is at a PVCS. Variable θ_3 is analogous to θ_2 but regarding the solar irradiance. Let μ_g and $\mu_{\bar{g}}$ be the moving averages of the last b_3 values of g_j and \bar{g}_j . The event is triggered if $\frac{|\mu_g - \mu_{\bar{g}}|}{\mu_g} > r_3$. This comparison is performed every o_3 units of time. When either event θ_2 or θ_3 is triggered, the EB continues travelling following the previous solution up until the recalculation is ready. For event θ_1 , we assume the recalculations are performed just before the arrival of the EB to a PVCS so that their execution times do not affect the operation. The fact that a new solution is not implemented immediately during events θ_2 and θ_3 as well as the differences between $e(s, v, p_s)$ and $\bar{e}(s, v, p_s)$ may cause the EB to incur in greater consumption than expected. Therefore, even though the EB's battery minimum energy level is a hard constraint when calculating each individual solution, we allow such minimum to be surpassed during the rolling horizon algorithm as long as the energy level remains positive.

An illustrative example of the rolling horizon algorithm is shown in Figure 3. The dotted lines crossing the route (which is the spline curve) represent the limits of each segment $s \in S$ and the symbols above the PVCSs represent the amounts of energy charged. Figure 3a represents the solution of the initial time slot of the rolling horizon algorithm before the start of the operation. Figure 3b shows the recalculation of the solution during the second time slot when event θ_2 is triggered. The event was triggered when the EB was located at the black circle while traversing the third segment of the route. Before that, the EB was travelling such segment with speed 1.

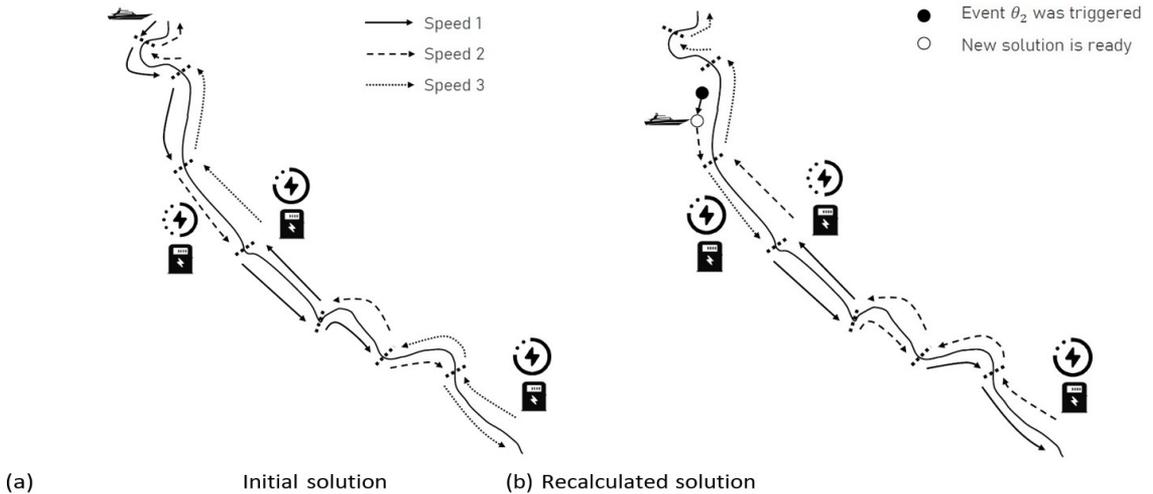


Figure 3. Illustrative example of the rolling horizon algorithm.

The solution of the DEBCP started being recalculated as soon as the event was triggered. However, due to the required computational time, by the time the new solution was ready the EB was located at the white circle. During the time it took for the solution to be recalculated, the EB kept travelling the third segment of the route with speed 1. Following the decisions of the new solution, the rest of such segment was traversed with speed 2.

5.2. Genetic algorithm

As previously mentioned, each time slot of the rolling horizon algorithm is solved using a GA. The method determines the speeds and charging decisions of the EB for the missing fraction of the fluvial operation. The speed decision is represented with two vectors, Γ_0 for the outward trip and Γ_1 for the return one. This was done so that certain operations during the generation of the initial population, the crossing and mutation operators are performed separately for each trip. Each of these vectors has a size equal to the number of segments which are yet to be finished in the respective trip. For example, consider that each trip of a given round trip is divided in three segments. If an event of the rolling horizon algorithm is triggered when the EB is traversing the second segment of the outward trip, new instances of Γ_0 and Γ_1 would be calculated. Vector Γ_1 would store three speeds while Γ_0 would only store two, one for the missing fraction of the second segment and another for the third segment.

When a new individual of the GA is created, a procedure named *ChargingPolicies*(\cdot) is executed. This procedure determines how much energy to charge, when and where to charge and which charging power to use. Let ψ represent all of these decisions clustered up. Having made such decisions, the feasibility of the individual is evaluated regarding all of the constraints of the problem and its cost is calculated. If an individual is deemed infeasible, its cost is set to infinity. We provide a detailed explanation of the procedure *ChargingPolicies*(\cdot) in Section 5.3. Our GA has all the classical components of that family of methods. We now describe how we implemented each of these components.

- (a) **Initial population:** Prior to designing the strategy to generate the initial population, we solved and analyzed some instances of the static version of our problem using a MILP formulation based on that of Villa et al. (2019). While it was not the case for every solution, some of them showed a tendency to have the same speeds over multiple consecutive segments as well as higher speeds during the trip against the river current. Therefore, we designed a method to generate the initial population which aims to favor similar characteristics to those of such results.

Algorithm 1 shows the structure of the initial population generation. The algorithm starts by initializing a list *population* to store the initial population (line 2) and two auxiliary indices ρ_0 and ρ_1 (line 3). The algorithm iterates trying to generate a *population* with at least one feasible individual, up until a maximum number of tries δ (lines 4–22). During each try, *population* is reinitialized as an empty list (line 5) and z individuals are generated (lines 6–15). To generate each individual, the algorithm first initializes vectors Γ_0 and Γ_1 (line 7).

As previously mentioned, the size of these vectors depends on the number of segments that are yet to be finished in the outward trip, \mathcal{G}_0 , and in the return one, \mathcal{G}_1 . The algorithm fills each Γ_k for the outward and return trips separately. To do so, first a *random* value is generated. If $random \leq \alpha$, with α been a given input value, Γ_k is filled with random speeds $v \in V$ by the procedure *RandomVector*(\mathcal{G}_k) (lines 9–10). Else, the procedure *FixedVector*($\mathcal{G}_k, V(\rho_k)$) fixes every position of Γ_k to the speed in index ρ_k of V (line 12) and the procedure *UpdateRho*(ρ_k) updates ρ_k (line 13). *UpdateRho*(ρ_k) performs different operations depending on which trip is against the river current. In the scenario that such trip is the outward one, the procedure checks if $\rho_0 = 1$. If so, ρ_0 is reset to its initial value $|V|$. Otherwise, ρ_0 is decreased by 1. The procedure performs the opposite operations for ρ_1 . On the other hand, in case that the trip against the river current is the return one, ρ_0 and ρ_1 must be switched in the previous explanation. When both speed vectors have been filled, the procedure *ChargingPolicies*($\phi, \Gamma_0, \Gamma_1, \bar{g}_j, f_{sh}$) is executed. This procedure returns ψ and the *cost* of the objective function (line 16). Then, a new individual is generated and added to *population* (line 17). When z individuals have been generated, procedure the *CheckForFeasibleIndividuals*(*population*) checks if at least one of them is feasible (line 19). If so, the algorithm breaks out of the outermost for

Algorithm 1. Initial population generation.

```

1: function GENERATEINITIALPOPULATION( $\alpha, z, V, \delta, \phi, \bar{g}_j, f_{sh}, \mathcal{G}_0, \mathcal{G}_1, \tau, \Gamma_0^{\tau-1}, \Gamma_1^{\tau-1}, \Gamma_0^0, \Gamma_1^0$ )
2:   population  $\leftarrow \langle \rangle$ 
3:    $\langle \rho_0, \rho_1 \rangle \leftarrow \langle |V|, 1 \rangle$ 
4:   for  $i=1$  to  $i=\delta$  do
5:     population  $\leftarrow list()$ 
6:     for  $j=1$  to  $j=z$  do
7:        $\langle \Gamma_0, \Gamma_1 \rangle \leftarrow \langle double[\mathcal{G}_0], double[\mathcal{G}_1] \rangle$ 
8:       for  $k=0$  to  $k=1$  do
9:         if  $random \leq \alpha$  then
10:             $\Gamma_k \leftarrow RandomVector(\Gamma_k, V)$ 
11:         else
12:             $\Gamma_k \leftarrow FixedVector(\mathcal{G}_k, V(\rho_k))$ 
13:             $\rho_k \leftarrow UpdateRho(\rho_k)$ 
14:         end if
15:       end for
16:        $\langle \psi, cost \rangle \leftarrow ChargingPolicies(\phi, \Gamma_0, \Gamma_1, \bar{g}_j, f_{sh})$ 
17:       population.add(new Individual( $\Gamma_0, \Gamma_1, \psi, cost$ ))
18:     end for
19:     if CheckForFeasibleIndividuals(population) = true then
20:       break
21:     end if
22:   end for
23:   if  $\tau \geq |\Pi|$  then
24:     population  $\leftarrow RemoveWorstIndividuals(\textit{population}, \Pi)$ 
25:     for each  $i \in \Pi$  do
26:        $\langle \psi, cost \rangle \leftarrow ChargingPolicies(\phi, \Gamma_0^i, \Gamma_1^i, \bar{g}_j, f_{sh})$ 
27:       population.add(new Individual( $\Gamma_0^i, \Gamma_1^i, \psi, cost$ ))
28:     end for
29:   end if
30:   return population
31: end function

```

loop (line 20). Else, *population* is discarded and the algorithm goes back to line 4. We note that the population is either discarded or accepted as a whole (i.e. even the infeasible individuals are accepted if there is at least one which is feasible). This was done as infeasible solutions may still provide good features when crossed with feasible individuals during the crossing stage.

The next part of the algorithm replaces the worst individuals of *population* with ones that have the speed vectors of solutions to previous time slots. This is done as such speeds have given good results in the past, and even if that were no longer the case, the new individuals would be eliminated during the rest of the GA's execution. For this reason, let Π be the set of all previous solutions of the GA up until the current time slot. Additionally, let Π' be a subset of Π having the solutions that will be used for replacing the worst individuals of *population*. Let also Γ_0^i and Γ_1^i be the speed vectors of the outward and return trip of solution $i \in \Pi'$. If the current time slot, τ , is greater than or equal to the cardinality of Π' (line 23), the algorithm replaces the worst $|\Pi'|$ individuals of population with new individuals having the speed vectors of previous solutions. To perform this replacement, procedure *RemoveWorstIndividuals*(*population*, Π') removes the worst $|\Pi'|$ individuals from *population* (line 24). Then, iterating over all elements $i \in \Pi'$, *ChargingPolicies*($\phi, \Gamma_0^i, \Gamma_1^i, \bar{g}_j, f_{sh}$) is executed generating a new ψ and *cost* (line 26). After that, a new individual is generated and added to *population* (line 27). Finally, the algorithm returns *population* (line 30).

- (b) **Crossover operator:** New individuals are generated in pairs. To do so, two different random parents are selected from the half of the population with the lowest objective function cost, as in Betancur et al. (2017). We use a single point crossover where the crossing point is randomly selected. The crossing is performed separately in between the pair of Γ_0 and the pair of Γ_1 . This process is repeated until the number of descendants is greater than or equal to λ percent of z .
- (c) **Mutation operator:** During the crossover, each speed vector has a probability γ of mutating. Such mutation consists of having a random speed of the vector changed to a random value $v \in V$.
- (d) **Population for the next generation:** We keep the *population* size z constant throughout the execution. For doing so, the top z individuals with lowest costs are passed from one generation to the next.
- (e) **Stopping criterion:** The algorithm stops after β generations have been performed.

5.3. Charging policies procedure

In this section we describe the procedure *ChargingPolicies*($\phi, \Gamma_0, \Gamma_1, \bar{g}_j, f_{sh}$). As previously mentioned, each time an individual is generated, this procedure makes the charging decisions ψ , which consist of how much energy to charge, when and where to charge and which charging power to use. After determining ψ , the feasibility of the individual is evaluated and its cost is calculate. To design this procedure we yet again analyzed the optimal solutions to the static version of the problem using the MILP formulation of Villa et al. (2019) looking for the best strategy to face these charging decisions. During each different time slot of the rolling horizon, only the fraction of the round trip that is yet to be traversed is evaluated.

Algorithm 2 shows the general structure of the *ChargingPolicies*($\phi, \Gamma_0, \Gamma_1, \bar{g}_j, f_{sh}$) procedure. The algorithm starts by initializing the *bestCost* and ψ (lines 2-3). Let ζ be a vector of size $|s_p|$ that stores the charging power to use in every PVCS during both trips. Procedure *SetHighestChargingPowers*(f_{sh}) as its name implies sets each position of ζ to the highest available charging power at each PVCS (line 4). This is because using such powers results in the best-case scenario in terms of charging times. Therefore, if such scenario were infeasible, no other scenario would be feasible. The algorithm starts iterating over an infinite loop (line 5) which is stopped by either one of two conditions that are explained later. The algorithm executes the procedure *EvaluateSolution*($\phi, \Gamma_0, \Gamma_1, \bar{g}_j, f_{sh}$) which evaluates the operation of the EB accounting for the traveling and charging times, the energy consumption and the constraints of the DEBCP (line 6). In this procedure, the EB charges at the last PVCS before its energy level would drop below its minimum value and charges the minimum amount of energy required to arrive at the following PVCS. The procedure returns the cost of the EB's operation given the current decisions, a boolean value *isFeasible* which indicates if the operation was feasible and three vectors, φ, ε and Ω . These vectors store the energy charged at each charging operation and the initial and final times of such operations, respectively. Vectors $\varphi, \varepsilon, \Omega$, and ζ must have an equal size $|s_p|$. For that reason, if a charging operation is not performed at the PVCS at the end of segment $j, \varphi[j]=0kWh$ and $\Omega[j]=\varepsilon[j]$. If the evaluation was feasible and its cost is less

Algorithm 2. Charging policies procedure.

```

1: function CHARGINGPOLICIES( $\phi, \Gamma_0, \Gamma_1, \bar{g}_j, f_{sh}$ )
2:    $bestCost \leftarrow \infty$ 
3:    $\psi \leftarrow \langle \ \ \rangle$ 
4:    $\leftarrow SetHighestChargingPowers(f_{sh})$ 
5:   while true do
6:      $\langle cost, isFeasible, \varphi, \varepsilon, \Omega \rangle \leftarrow EvaluateSolution(\phi, \Gamma_0, \Gamma_1, \bar{g}_j, \zeta)$ 
7:     if  $isFeasible = true$  and  $cost < bestCost$  then
8:        $bestCost \leftarrow cost$ 
9:        $\psi \leftarrow \langle \varphi, \varepsilon, \Omega, \zeta \rangle$ 
10:    else if  $isFeasible = false$  then
11:      break
12:    end if
13:     $\langle \kappa, \nu \rangle \leftarrow \langle -1, \infty \rangle$ 
14:    for  $j = 0$  to  $j = |S_p|$  do
15:      if  $0 < \Omega[j] - \varepsilon[j] < \nu$  and  $LowerChargingPower(j, \zeta, f_{sh}) = true$  then
16:         $\langle \kappa, \nu \rangle \leftarrow \langle j, \Omega[j] - \varepsilon[j] \rangle$ 
17:      end if
18:    end for
19:    if  $\kappa = 1$  then
20:      break
21:    else
22:       $\zeta \leftarrow DecreaseChargingPower(\zeta, \kappa)$ 
23:    end if
24:  end while
25:  return  $\langle \psi, bestCost \rangle$ 
26: end function

```

than $bestCost, \psi$ and $bestCost$ are updated (lines 7-9). Else, if the evaluation was infeasible the infinite loop stops (lines 10-11). The algorithm then searches for a candidate charging operation in which to evaluate a lower power (lines 13-23) looking to decrease the degradation costs. For doing so, the algorithm initializes two auxiliary variables, κ and ν (line 13). The former stores which PVCS must have its charging power decreased, while the latter saves the charging time of the operation at such PVCS. Let $LowerChargingPower(i, \zeta, f_{sh})$ be a procedure that returns a boolean value true if the PVCS at the end of segment j has a lower available charging power than the one being considered in ζ . The algorithm uses this procedure to look for the charging operation that took the least amount out of the ones performed in PVCSs that have at least one lower available charging power than the ones in ζ , and stores its index in κ (lines 14-18). If no charging operation has a lower available charging power, the infinite loop stops (lines 19-20). Else, the charging operation at the PVCS with index κ gets its charging power decreased to the next highest available power at its PVCS using a procedure named $DecreaseChargingPower(\zeta, \kappa)$. The algorithm continues iterating until either of the two stopping conditions (finding an infeasible solution or evaluating a scenario charging only with the lowest available power at each PVCS) are met. Finally, the algorithm returns ψ and $bestCost$ (line 25).

6. Computational experiments

In this section we present the two types of computational experiments we performed for the DEBCP. Considering that we used a metaheuristic as our solution method, we first compared its performance against a MILP formulation based on that of Villa et al. (2019) when solving a static version of the DEBCP. With the goal of evaluating the importance of dynamically recalculating the solution of the problem, we then compare

the DEBCP against a static scenario without the solution recalculations. The algorithm was implemented on Java (java/jdk-1.8.0_112). Experiments were run on a computing cluster with Intel Xeon E5-2683 v4 processor (with 32 cores at 2.1 GHz) and 64 GB of RAM running on Linux Rocks 6.2 - 64 bits - Centos 6.6.

6.1. Test instances

For these experiments, we built a set of test instances based on a future electric fluvial transport operation in Magangué Colombia that will be implemented by the alliance “ENERGETICA 2030” (Energética 2030, 2023). Such location was selected as it is an important port in the Magdalena River (Colombia’s principal river). Most of the parameters of the instances were taken from a field research performed by the alliance. Some parameters are shared by all instances while others are instance dependent. We first describe the shared ones. The set of speeds which may be selected is $V = \{20, 21, \dots, 69, 70\} \text{ km/h}$. For the charging power, we set $H = \{65, 130\} \text{ kW}$. The EB’s battery has a capacity Q of 130 kWh and a cost of $24,595 \text{ USD}$ according to the supplier’s quotation (the name of the supplier was treated as undisclosed commercial information). For the battery degradation we considered 10 intervals for its energy level, so $M = \{0 - 13, 13 - 26, \dots, 117 - 130\} \text{ kWh}$. With the battery’s cost, we calculated the wear cost for each interval $m \in M$ as in Han et al. (2014). For the penalization due to different charging powers, we used the equations of Omar et al. (2014). We set the minimum energy level percentage of the battery to 10%. For the PVCS, we considered $a_s = 84$ panels and $f_{sh} = 1 \forall s \in S_p, \forall h \in H$. Each panel has an area of 2.24 m^2 and an efficiency of 20.71% (Solartex, 2020). The cost of energy to purchase from the grid was 0.18 USD/kWh (Empresas Públicas de Medellín, 2021).

The instance-dependent parameters are shown in Table 1. All instances have the round trip starting at Magangué and are divided in three groups of three instances each depending on where the outward trip ends and the return one starts. The first group of instances perform a round trip between Magangué and the city of Pinillos, the second group between Magangué and an inn on the bank of the Magdalena River, and the third one between Magangué and the city of Achí. Instances within a group have the same segments, route length, location and number of PVCSSs, but different T_{max} and time windows. The segments have a length of 1 km , except for segments $s \in S_p$ which were properly shorten according to the location of the PVCSSs. Such PVCSSs were located in actual settlements across the Magdalena River.

6.2. Solution method comparison

To evaluate the performance of the GA, in this section we compare its results to those of a MILP formulation based on that of Villa et al. (2019). We added the upper and lower time windows to depart from each charging station to the formulation as well as the minimum energy level for the EB’s battery. To maintain the variant of the MILP formulation as close as possible to that of Villa et al. (2019), we considered non-PV charging stations rather than PVCSSs. The MILP formulation was implemented using gurobipy with Python 3.7. For the GA, we set the population size (z) to 720 individuals, the probability of initializing a speed vector as random (α) to 99%, the percentage of descendants per generation (λ) to 20%, the mutation rate (γ) to 1% and the number of iterations (β) to 5000. These values were selected after conducting a parameter tuning campaign. These parameters gave the overall best results considering both the objective function and the execution times. For the sake of brevity, we will not discuss these experiments. We set a time limit of 7200 seconds for the MILP formulation. Table 2 presents the comparison between the two methods. Columns 2 and 3 show the costs and CPU times of the MILP formulation. The MILP formulation was able to find the optimal solution in eight out of nine instances within the time limit. The only instance for which the optimal solution was not found was Achí3. However, the optimality gap of such solution was just 0,014%. We note that the CPU time was relatively low for the first group of instances but not for the other two groups. This is because the second and third groups of instances are bigger than the first one in terms of the number of segments in which the round trips were divided, as shown in Table 1. The results of the GA were obtained with 10 different runs, each with its own random seed. Columns 4 and 5 in Table 2 present the average and best-found costs by the GA for each instance during those 10 runs, while column 6 presents the average CPU times. The CPU times also increased in the second and third groups of instances when compared to the first one, but not in the same magnitude than it did for the MILP formulation. Columns 7 and 8 show the gaps between the best-found solution of the MILP formulation (which was almost always the optimal one) and, respectively, the averages and best-found costs of the GA. Even though the GA never found the same solutions as the MILP formulation, the average and maximum gaps were just 0.58% and 0.87% when compared to the average of the 10 runs and 0.4% and 0.83% in the case of the best-found solutions. This shows that the GA is able to provide good solutions with respect to those of the MILP formulation in terms of cost. We note that most of the gaps for the average of the 10 runs are rather close to the best-found gaps, which shows that the

Table 1. Instance-dependent parameters.

Instance	T_{max} (h)	Departure hour	PVCSs*	Route length (km)	Segments
Pinillos1	4	10:00	5	110	112
Pinillos2	4.5	10:00	5	110	112
Pinillos3	5	10:00	5	110	112
Inn1	6	9:00	9	171.6	174
Inn2	6.75	9:00	9	171.6	174
Inn3	7.5	9:00	9	171.6	174
Achi1	8	7:00	11	208.8	212
Achi3	9	7:00	11	208.8	212
Achi3	10	7:00	11	208.8	212

*Photovoltaic charging stations.

Table 2. Comparison between the MILP formulation and the GA.

Instance	MILP		GA					Speedup
	Cost (USD)	CPU* time (sec)	Average cost (USD)	Best found cost (USD)	Average CPU time (sec)	Average gap (%)	Best found gap (%)	
Pinillos1	37.33	6.75	37.49	37.40	2.59	0.43	0.19	2.60
Pinillos2	32.24	25.78	32.51	32.32	2.59	0.84	0.25	9.94
Pinillos3	28.19	83.57	28.28	28.25	2.58	0.33	0.23	32.36
Inn1	83.33	5763.60	83.53	83.51	3.83	0.24	0.21	1503.17
Inn2	70.96	31.98	71.16	71.13	3.94	0.29	0.24	8.12
Inn3	63.85	195.63	64.39	64.19	3.99	0.83	0.53	49.09
Achi1	102.64	339.51	103.54	103.49	4.67	0.87	0.83	72.63
Achi2	90.10	354.59	90.60	90.52	4.80	0.56	0.47	73.93
			Achi3					
			81.44					
			7200			0.83	0.63	1376.99
			82.12					
			81.96					
			5.23					
Min		6.75			2.58	0.24	0.19	2.60
Max		7200			5.23	0.87	0.83	1503.17
Average		1555.71			3.80	0.58	0.40	347.65

Gap = 100 * (GA cost - MILP formulation cost) / MILP formulation cost. * Central processing unit.

GA gives consistent results. Column 9 shows the speedups of the GA when compared to the MILP formulation, with the average speedup being 347.65. Considering that the DEBCP requires its solution to be constantly reevaluated, having low CPU times is crucial for solving the problem. Additionally, we note that the GA had similar CPU times for instances within the same group regardless of the differences in T_{max} and time windows, while the computation times of the MILP formulation vary significantly in between instances from the same group. This variance is an issue when solving the DEBCP, as each recalculation can be seen as a separate optimization problem with different parameters, such as time windows, T_{max} and initial energy level of the EB's battery. Therefore, the CPU times would be quite inconsistent for different time slots of the rolling horizon algorithm. Considering the gaps, speedups and the consistency of the CPU times, the GA is better suited than the MILP formulation for solving the DEBCP.

6.3. Comparison between the DEBCP and the static scenario

To assess the impact of including the dynamic component in the problem, in this section we compare the DEBCP to a static scenario with no recalculations of the solution. For the static scenario, the solution to the problem is estimated prior to the beginning of the EB's operation and remains unchanged throughout such operation. Using a simulation of the EB's operation (which is detailed in the following paragraph), such solution is evaluated in terms of the feasibility and cost of the operation. These results are then compared against the ones the DEBCP which uses the rolling horizon algorithm with the dynamic recalculations of the solution while considering the same

simulation. To compare the differences between $e(s, v, p_s)$ and $\bar{e}(s, v, p_s)$, and between g_j and \bar{g}_j during the rolling algorithm, we set the number of previous data to consider for the moving averages (b_2 and b_3) to 3, the value to consider a difference as significant (r_2 and r_3) to 5% and the frequency to compare such differences (α_2 and α_3) to 10 minutes. As mentioned in Section 3.3, \bar{g}_j is estimated using a bidirectional long short-term memory artificial neural network. For the static scenario, such estimation is performed only prior to the beginning of the fluvial operation. For the DEBCP, a new estimation is performed every 10 minutes and each time slot uses the most recent available estimation. As mentioned in Section 3.1, $\bar{e}(s, v, p_s)$ is estimated using the model of Savitsky (1964). Such estimation is maintained without any kind of update during the simulation in both the static scenario and the DEBCP as our consumption model does not vary with time. However, if a consumption model with actual variations in time were to be implemented, our rolling horizon algorithm could still be used as a solution method.

We now describe the simulation with which the static scenario and the DEBCP were compared. Such simulation emulates the operation of the EB following the decisions made in either scenario. This simulation uses the real data of the energy consumption, $e(s, v, p_s)$, and the solar irradiance, g_j . For the DEBCP, the CPU times of the GA when recalculating a solution are taken into account in this simulation as we mentioned in Section 5.1. We fixed the CPU times when solving each recalculation of the solution to the CPU times of the GA reported in Table 2, even though such times would vary in real life. This was done so that the results of the rolling horizon algorithm are always the same each time it is executed. The port in Magangué opens from 6:00 to 18:00, so we define set of time intervals of a day as $T = \{6:00-6:10, 6:10-6:20, \dots, 17:50-18:00\}$. Data for g_j were taken from National Solar Radiation Database (2023) for the region near Magangué during May 11th, 2019. Such day was selected as it presented an irregular irradiance profile. Figure 4 compares the solar irradiance profile from that day against the one from January 2nd, 2019, which was a sunny day with a regular solar irradiance profile. Given that the data set is half-hourly spaced, we interpolated the missing data to have 10 minutes long intervals. Additionally, Π' was always composed by the solutions of the first and previous time slots up to the current execution of the solution method. To generate the data for $e(s, v, p_s)$, we considered an error with respect to $\bar{e}(s, v, p_s)$ according to experimental data given to us by a fellow research of the alliance ENERGETICA 2030. As the data did not fit a standard probability distribution function, we used an empiric distribution based on such data to generate our consumption. This distribution was heavily skewed towards having lower values compared to those of $\bar{e}(s, v, p_s)$, therefore making such estimation rather overestimated. However, as other consumption models could be used when solving the DEBCP, we also evaluated an scenario in which $\bar{e}(s, v, p_s)$ was underestimated. For that reason, we made a second empiric distribution with a mirrored skewness towards higher than estimated energy consumption values.

Table 3 shows the comparison of the DEBCP versus the static scenario considering the lower than estimated energy consumption. Columns 2 and 5 show the actual costs of the simulation for each scenario using the real values for the parameters. Column 8 presents the gap between these two costs. The DEBCP managed to decrease the overall cost for every single instance, with an average reduction of 2.67%. We note that the impact

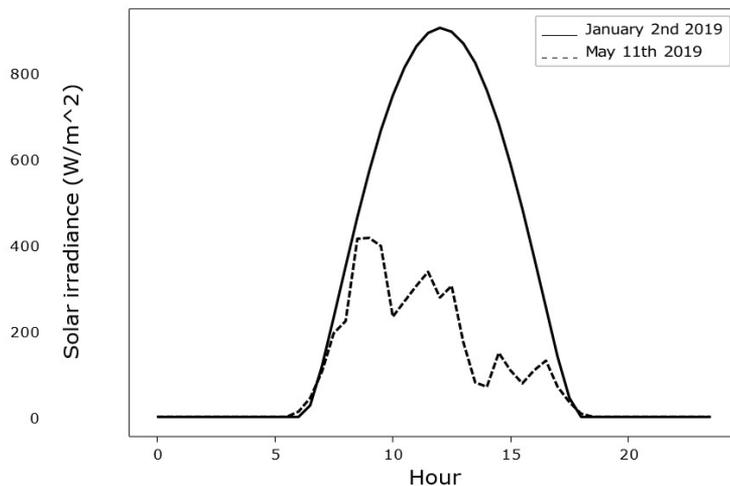


Figure 4. Solar irradiance profiles during a cloudy day (May 11th 2019) and a sunny one (January 2nd 2019) in Magangué [data taken from National Solar Radiation Database (2023)].

was higher in the groups of instances with longer round trips, as the average reduction for instances going to Pinillos was 1.63%, while the instances going to the inn had an average decrement of 2.44% and the ones going to Achi an average of 3.92%. Even though these reductions in costs may not be significant for a single execution of the EB's operation, the savings in absolute costs that would be made during multiple years' worth of operation of the EB may be substantial. Columns 3 and 6 present the total amount of energy that was drained from the battery when it was below its minimum energy level during the rolling horizon algorithm. Columns 4 and 7 show the total amount of time that the route surpassed either the time windows or T_{max} . None of the instances in either the DEBCP or the static scenario violated these constraints, which is because the operation ends up being less demanding constraint-wise than what was expected when solving the problem, as the $e(s, v, p_s)$ tended to be lower than $\bar{e}(s, v, p_s)$.

Having the same structure as Table 3, Table 4 shows the results of the comparison considering the higher than estimated energy consumption. Both the DEBCP and the static scenario violated either or both the time and energy related constraints in every single instance, although they did manage to keep a positive energy level for the EB's battery at all times. However, the DEBCP violated both constraints by a lower magnitude in every instance, which shows that it tried to compensate for the uncertainty of the operation.

To measure the extent of these reductions, we calculated the percent differences between the averages of columns 3 and 6 for the energy violations and between columns 4 and 7 for the time related ones. The results showed that the energy and time violations during the solutions of the DEBCP were on average 21.73% and 65.35% lower than those of the static scenario. The differences in energy violations were more significant in the groups of instances with longer round trips, as the average differences were 19.7%, 21.19% and 23.35% for the groups of instances going to Pinillos, the inn and Achi respectively. On the contrary, the differences in time violations were more significant in the

Table 3. Comparison of the DEBCP and the static scenario with lower than estimated energy consumption.

Instance	Static scenario			DEBCP			
	Cost (USD)	Energy violation (kWh)	Time violation (h)	Cost (USD)	Energy violation (kWh)	Time violation (h)	Gap (%)
Pinillos1	33.48	0	0	32.66	0	0	2.43
Pinillos2	28.91	0	0	28.69	0	0	0.74
Pinillos3	26.53	0	0	26.08	0	0	1.72
Inn1	72.99	0	0	71.00	0	0	2.73
Inn2	62.60	0	0	60.98	0	0	2.58
Inn3	57.07	0	0	55.92	0	0	2.02
Achi1	89.23	0	0	86.06	0	0	3.55
Achi2	80.22	0	0	76.61	0	0	4.50
Achi3	72.87	0	0	70.16	0	0	3.72
Min		0	0		0	0	0.74
Max		0	0		0	0	4.50
Average		0	0		0	0	2.67

Gap = 100 * (static scenario cost - DEBCP cost) / static scenario cost.

Table 4. Comparison of the DEBCP and the static scenario with higher than estimated energy consumption.

Instance	Static scenario			DEBCP			
	Cost (USD)	Energy violation (kWh)	Time violation (h)	Cost (USD)	Energy violation (kWh)	Time violation (h)	Gap (%)
Pinillos1	38.59	8.69	0.11	39.80	8.38	0.03	-3.14
Pinillos2	33.63	7.98	0.10	34.68	6.14	0.02	-3.14
Pinillos3	30.89	7.85	0.12	31.16	5.30	0.00	-0.88
Inn1	87.47	27.68	0.18	92.11	21.33	0.09	-5.30
Inn2	74.48	22.69	0.15	78.82	18.03	0.06	-5.82
Inn3	67.72	21.52	0.27	71.85	17.20	0.06	-6.10
Achi1	107.95	31.92	0.32	113.51	28.52	0.30	-5.14
Achi2	96.66	29.12	0.40	100.48	23.07	0.12	-3.96
Achi3	88.48	28.07	0.43	92.89	17.23	0.04	-4.99
Min		7.85	0.10		5.30	0.00	-0.88
Max		31.92	0.43		28.52	0.30	-6.10
Average		20.61	0.23		16.13	0.08	-4.28

Gap = 100 * (static scenario cost - DEBCP cost) / static scenario cost.

groups with shorter routes, with the average differences being 82.61%, 64.32% and 54.97% for the groups going to Pinillos, the inn and Achí. We concluded that this is because making more significant corrections to the energy violations makes it harder for the time violations to also be compensated at the same time. Due to these corrections, the costs of the DEBCP were always higher than those of the static scenario, with an average increment of 4.28%. The results with both types of consumption errors show that the recalculations of the DEBCP are able to provide beneficial impacts for the EB's operation, either by reducing its cost if possible, or by trying to correct it if needed.

7. Conclusions

In this research we extend the EBCP. The original problem consists of selecting the speeds for an EB to traverse a set of segments in which a fluvial route is divided as well as making the charging decisions of the EB. The objective function is to minimize the cost of the energy to purchase from the grid while charging the EB and a cost associated to the degradation of its battery. Our additions to the problem consist mainly of performing dynamic recalculations for the problem's solution as more information of the fluvial operation is known. Additionally, we added a set of time windows for the EB to depart from certain nodes of the route and the use of PVCs. The dynamic recalculations aim at correcting certain decisions that were made with estimated information. We named this variant of the problem the DEBCP.

We propose a rolling horizon genetic algorithm to solve the DEBCP. Such method consists of a rolling horizon algorithm that dynamically recalculates the solution of the problem each time a set of events are triggered. Each recalculation is solved using a GA with an embedded set of charging policies. We used a GA to have low CPU times as the DEBCP requires multiple recalculations during the operation of the EB. To assess the performance of the GA with the charging policies, we compared its solutions during 10 different runs to those of a MILP formulation. The results show that the GA is able to provide an average speedup of 347.65 and an average gap in the cost of the objective function of 0.58% when considering the average of the 10 runs and 0.4% in the case of the best-found solution for each instance. We also found that the GA's CPU times varied less than those of the MILP formulation when changing certain parameters of the problem. All of these results show that the GA is more suitable for facing the dynamic component of the DEBCP. We then assessed the impact of such recalculations by comparing the solution of the DEBCP to those of a static scenario with no recalculations. This comparison was performed considering both a data set where the simulated energy consumption was lower than the estimated and another data set with higher simulated consumption. With the former data set, the DEBCP was able to reduce the cost of the operation in all instances, with an average reduction of 2.67%. When considering the higher simulated consumption, both the DEBCP and the static scenario violated the energy and time related constraints of the problem. However, the DEBCP managed to reduce the overall violation of both types of constraints in all instances. To measure the extent of such reductions, we calculated the percent difference between the energy and time violations. Such results showed an average diminution of 21.73% for the violations of the energy constraint and 65.35% for the time related ones. These results show that the dynamic recalculations of the DEBCP are able to provide beneficial impacts for the problem by either reducing the cost of the operation or by compensating for the uncertainty in some estimated parameters.

One interesting direction for a future research consists of considering PVCs with batteries of their own (or energy storage system, as they are often called in such context). These components can help making the overall charging system more resilient against power outages or multiple continuous days with low solar irradiance. Additionally, evaluating the degradation of the PVCs' batteries could be an interesting addition to the objective function of such future research. Lastly, we wanted to talk about possible challenges related to the implementation of the dynamic recalculations. As can be seen in Table 2, the computation times for our algorithm are in the range of just a couple of seconds. Additionally, these computation times tend to get shorter as the EB advances on its route, as a shorter version of the course needs to be solved each time. Therefore, the computation times themselves do not pose a significant challenge. In our opinion, the biggest challenge for the implementation actually comes from obtaining the input data to solve the problem. This is because obtaining these data requires have a set of sensors to accurately measure in real time the energy level and consumption of the EB as well as measuring and predicting the solar irradiance with which each recalculation of the solution is calculated.

Acknowledgements

The authors would like to thank Universidad EAFIT and the alliance ENERGETICA 2030, which is a Research Program with code 58667 from the Colombia Científica initiative, funded by The World Bank through the call 778-2017 Scientific Ecosystems. The research program is managed by the Colombian Ministry of Science,

Technology and Innovation (Minciencias) with contract No. FP44842-210-2018. Additionally, the authors would like to acknowledge the supercomputing resources made available by the Centro de Computación Científica APOLO at Universidad EAFIT (www.eafit.edu.co/apolo) to conduct the research reported in this paper.

References

- Barré, A., Deguilhem, B., Grolleau, S., Gérard, M., Suard, F., & Riu, D. (2013). A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, *241*, 680-689. <http://dx.doi.org/10.1016/j.jpowsour.2013.05.040>.
- Betancur, E., Osorio-Gómez, G., & Rivera, J. C. (2017). Heuristic optimization for the energy management and race strategy of a solar car. *Sustainability*, *9*(10), 1576. <http://dx.doi.org/10.3390/su9101576>.
- Bhatti, A. R., Salam, Z., Aziz, M. J. B. A., Yee, K. P., & Ashique, R. H. (2016). Electric vehicles charging using photovoltaic: status and technological review. *Renewable & Sustainable Energy Reviews*, *54*, 34-47. <http://dx.doi.org/10.1016/j.rser.2015.09.091>.
- Bischi, A., Taccari, L., Martelli, E., Amaldi, E., Manzolini, G., Silva, P., Campanari, S., & Macchi, E. (2019). A rolling-horizon optimization algorithm for the long term operational scheduling of cogeneration systems. *Energy*, *184*, 73-90. <http://dx.doi.org/10.1016/j.energy.2017.12.022>.
- Bullard, N. (2019). *Electric car price tag shrinks along with battery cost*. New York: Bloomberg. Retrieved in September 2019, from <https://www.bloomberg.com/opinion/articles/2019-04-12/electric-vehicle-battery-shrinks-and-so-does-the-total-cost>
- Candela. (2021). *How we redefined boating*. Stockholm: Candela. Retrieved in 11 September 2023, from <https://candela.com/how-we-redefine-boating/>
- Choi, J. W., & Aurbach, D. (2016). Promise and reality of post-lithium-ion batteries with high energy densities. *Nature Reviews. Materials*, *1*(4), 16013. <http://dx.doi.org/10.1038/natrevmats.2016.13>.
- Deschênes, A., Gaudreault, J., Vignault, L. P., Bernard, F., & Quimper, C. G. (2020, October 11-14). The fixed route electric vehicle charging problem with nonlinear energy management and variable vehicle speed. In M. Elgendi, & H. Shah-Mansouri (Eds.), *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 1451-1458). New York, United States: IEEE. <http://dx.doi.org/10.1109/SMC42975.2020.9283062>.
- Empresas Públicas de Medellín. (2021). *Tarifas y costo de energía eléctrica – mercado regulado*. Medellín: EPM. Retrieved in January 2021, from https://www.epm.com.co/site/Portals/2/Documentos/tarifas/Energia_2021/Publicacion_Tarifas_Energia__16_enero_2021.pdf
- Energética 2030. (2023). *¿En qué consiste la alianza Energética 2030?*. Colombia. Retrieved in October 2023, from <https://www.energetica2030.co>.
- Forbes. (2021). *What Are The Different Levels Of Electric Vehicle Charging?* New Jersey: Forbes. Retrieved in October 2023, from <https://www.forbes.com/wheels/advice/ev-charging-levels/>.
- Giménez, J. (2017). *Así es la primera canoa solar del Amazonas*. Sharamentsa: El País. Retrieved in March 2021, from https://elpais.com/elpais/2017/06/02/planeta_futuro/1496412527_334830.html
- Gkiotsalitis, K., & Van Berkum, E. (2020). An exact method for the bus dispatching problem in rolling horizons. *Transportation Research Part C, Emerging Technologies*, *110*, 143-165. <http://dx.doi.org/10.1016/j.trc.2019.11.009>.
- Han, S., Han, S., & Aki, H. (2014). A practical battery wear model for electric vehicle charging applications. *Applied Energy*, *113*, 1100-1108. <http://dx.doi.org/10.1016/j.apenergy.2013.08.062>.
- International Energy Agency. (2020). *Global EV outlook 2020*. Paris: IEA. Retrieved in March 2021, from <https://www.iea.org/reports/global-ev-outlook-2020>
- Jaimurzina, A., Wilmmsmeier, G., & Montiel, D. (2017). *Eficiencia energética y movilidad eléctrica fluvial: soluciones sostenibles para la Amazonía*. Santiago: CEPAL.
- Krupp, C. (2010). Electrifying rural areas: extending electricity infrastructure and services in developing countries. In W. Ascher & C. Krupp (Eds.), *Physical infrastructure development: balancing the growth, equity, and environmental imperatives* (pp. 203-224). New York: Palgrave Macmillan. http://dx.doi.org/10.1057/9780230107670_8.
- Kumar, P., & Khani, A. (2021). An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access. *Transportation Research Part C, Emerging Technologies*, *122*, 102891. <http://dx.doi.org/10.1016/j.trc.2020.102891>.
- Li, D., Zouma, A., Liao, J.-T., & Yang, H. (2017). Review and behaviour for electric vehicles: review and numerical analyses of several models. *Transportation Research Part B: Methodological*, *103*, 158-187. <http://dx.doi.org/10.1016/j.trb.2017.01.020>.
- Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, *225*(1), 1-11. <http://dx.doi.org/10.1016/j.ejor.2012.08.015>.
- Rahman, I., Vasant, P. M., Singh, B. S. M., Abdullah-Al-Wadud, M., & Adnan, N. (2016). Review of recent trends in optimization techniques for plug-in hybrid, and electric vehicle charging infrastructures. *Renewable & Sustainable Energy Reviews*, *58*, 1039-1047. <http://dx.doi.org/10.1016/j.rser.2015.12.353>.
- Ramakrishna, R., & Scaglione, A. (2016, November 6-9). A compressive sensing framework for the analysis of solar photo-voltaic power. In Institute of Electrical and Electronics Engineers (Org.), *2016 50th Asilomar Conference on Signals, Systems and Computers* (pp. 308-312). New York, United States: IEEE. <http://dx.doi.org/10.1109/ACSSC.2016.7869048>.
- Saini, V., Singh, S., NV, S., & Jain, H. (2016). Genetic algorithm based gear shift optimization for electric vehicles. *SAE International Journal of Alternative Powertrains*, *5*(2), 348-356. <http://dx.doi.org/10.4271/2016-01-9141>.
- Savitsky, D. (1964). Hydrodynamic design of planing hulls. *Marine Technology and SNAME News*, *1*(4), 71-95. <http://dx.doi.org/10.5957/mt1.1964.1.4.71>.
- Seddig, K., Jochem, P., & Fichtner, W. (2017). Integrating renewable energy sources by electric vehicle fleets under uncertainty. *Energy*, *141*, 2145-2153. <http://dx.doi.org/10.1016/j.energy.2017.11.140>.
- Shepero, M., Lingfors, D., Widén, J., Bright, J. M., & Munkhammar, J. (2020). Estimating the spatiotemporal potential of self-consuming photovoltaic energy to charge electric vehicles in rural and urban nordic areas. *Journal of Renewable and Sustainable Energy*, *12*(4), 046301. <http://dx.doi.org/10.1063/5.0006893>.

- Shui, C. S., & Szeto, W. (2018). Dynamic green bike repositioning problem—a hybrid rolling horizon artificial bee colony algorithm approach. *Transportation Research Part D, Transport and Environment*, 60, 119–136. <http://dx.doi.org/10.1016/j.trd.2017.06.023>.
- Singh, P. C. (2020). *Time series forecasting*. San Francisco: GitHub. Retrieved in January 2021, from https://github.com/pcsingh/Time_Series_Forecasting
- Solartex. (2020). *Panel solar 465 Watts Jinko TIGER Mono*. Medellín: Solartex. Retrieved in July 2020, from <https://www.solartex.co/tienda/producto/panel-solar-465-watts-jinko-tiger-mono/>
- Vélez, C., & Montoya, A. (2023). The fluvial passenger transport design problem with an Electric Boat. *Case Studies on Transport Policy*, 12, 100972. <http://dx.doi.org/10.1016/j.cstp.2023.100972>.
- Vélez, C., Villa, D., & Montoya, A. (2020). Infrastructure estimation for a freight/personal transport operation with an Electric Boat on the Magdalena River. In J. C. Figueroa-García, F. S. Garay-Rairán, G. J. Hernández-Pérez & Y. Díaz-Gutierrez (Eds.), *Applied computer sciences in engineering: 7th Workshop on Engineering Applications, WEA 2020, Bogota, Colombia, October 7-9, 2020, proceedings* (pp. 329–337). Cham: Springer. http://dx.doi.org/10.1007/978-3-030-61834-6_28.
- Villa, D., & Montoya, A. (2018, April 18–20). A taxonomy of energy consumption models for electric vehicles. In Institute of Electrical and Electronics Engineers (Org.), *OVICI-MOYCOT 2018: Joint Conference for Urban Mobility in the Smart City* (pp. 1–7). New York, United States: IEEE. <http://dx.doi.org/10.1049/ic.2018.0016>.
- Villa, D., Montoya, A., & Ciro, J. M. (2019). The electric boat charging problem. *Production*, 29, e20190067. <http://dx.doi.org/10.1590/0103-6513.20190067>.
- Villa, D., Montoya, A., & Herrera, A. M. (2020). The electric riverboat charging station location problem. *Journal of Advanced Transportation*, 2020, 6527924. <http://dx.doi.org/10.1155/2020/6527924>.
- Wu, Y., Zhang, J., Ravey, A., Chrenko, D., & Miraoui, A. (2020). Real-time energy management of photovoltaic-assisted electric vehicle charging station by Markov decision process. *Journal of Power Sources*, 476, 228504. <http://dx.doi.org/10.1016/j.jpowsour.2020.228504>.
- Zhang, W., Yan, X., & Zhang, D. (2017). Charging station location optimization of electric ship based on backup coverage model. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 11(2), 323–327.