OPTIMAL OPERATIONS OF AN AUTOMATED VEHICLE PARKING LOT[1]

Sina Bahrami, University of Toronto
Matthew Roorda, University of Toronto

Introduction

Autonomous Vehicles (AVs) will bring advantages such as enhanced mobility, increased safety, and eliminated hassles of finding a parking spot. AV users can exit from their vehicles at their final destinations, and send the vehicles to find a parking spot themselves. This self-parking capability of AVs is considered as their most beneficial advantage in the World Economic Forum survey by approximately 50% of respondents (Mitchell, 2019). The self-parking capability not only provides the opportunity to park AVs farther from the final destination (Nourinejad &Amirgholy, 2018) but also can decrease the area needed for parking and revitalize valuable lands allocated to parking. Since passengers do not present in the AVs when they enter a parking lot, the required room for opening an AV's doors becomes redundant. This along with the elimination of elevators and staircases can decrease the average space per vehicle by 2 square meters (Techworld, 2016). Another essential strategy to decrease land utilization is stacking AVs behind each other in car-parks as shown in Figure 1. Although this kind of design reduces land utilization and increases parking capacity, it causes a blockage if the barricaded AV wants to leave earlier, and the parking operator must relocate blocking vehicles.

Nourinejad et al. (2018)investigate the design of such parking facility layouts. They divide a car-park into a number of islands and gaps. The islands are used to store vehicles and the gaps are used for vehicles to maneuver in and out of parking spots. Each island is made of some stacks as shown in Figure 1. The inter-island gaps are used as waiting area for blocking AVs. In this way, the blocking AVs move to the inter-island gap and wait there to make a clear path for the summoned AVto exit the facility. These AVs are assumed to return to their original stack when the recalled AV leaves. Nourinejad et al. (2018) assume that vehicle arrival and dwell time follow Poisson and exponential distributions, respectively. Nourinejad et al. (2018) find the optimal size of islands to minimize the expected number of relocations given a land dimensions and parking demand. Their model is a macroscopic model for strategic design, and only selects the island in which a newly arrived AV should be parked but does not select the exact spot on that island. In this research, we investigate how arrival and departure time information of each vehicle influences spot allocation, and propose a relocation policy to minimize the number of future movements given a car-park layout.

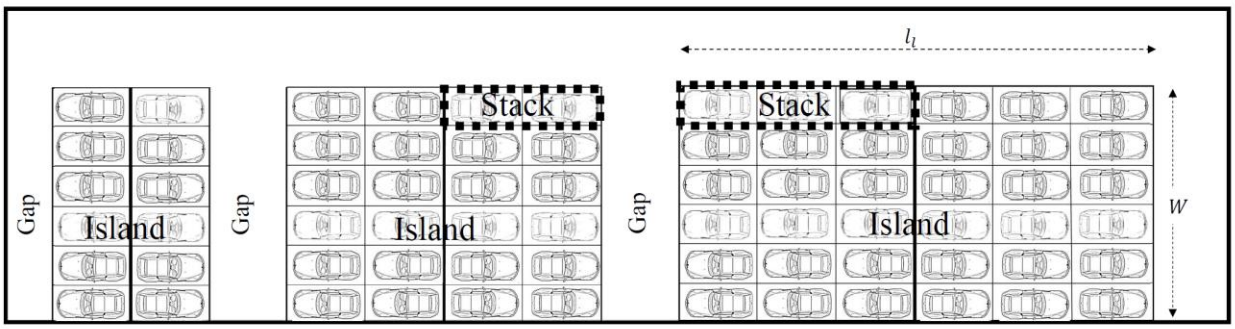
The operation of such AV parking facilities has not been investigated yet in the literature. However, the problem is similar to the container relocation problem (CRP) which concerns the storage and handling of containers in a storage area. Different versions of the CRP have been studied extensively. The CRP is categorized into static and dynamic by whether or not allowing additional container arrivals during the retrieval process (Borjian et al., 2013;HakanAkyuz& Lee, 2014). The static CRP is further categorized into full and partial information. It is assumed that the departure order of containers is known in advance in full information models (Caserta et al., 2012; Zehendner et al., 2015). In contrast, containers are classified based on departure time windows in partial information models, while the retrieval order of containers in a time window is not clear (Zhao & Goodchild, 2010; Ku&Arthanari, 2016; Galle et al., 2018). Comprehensive reviews of these problems are presented in Steenken et al. (2004), Stahlbock &Vob (2008), Lehnfeld &Knust (2014),and Bierwirth & Meisel (2015).

[1] 154th Annual Meetings of the *Canadian Transportation Research Forum*, May 26 - 29, 2019 at Vancouver, British Columbia

To the best of our knowledge, there is no research investigating the dynamic version of CRP with partial information. There are also some critical differences between our problem and the CRP. One such difference is that the size of stacks is fixed in a container yard due to the crane height restriction, while our problem can have different stack sizes in a car-park, as discussed in Nourinejad et al. (2018). Another is that containers need to be placed on top of each other due to gravity, while vehicles can be parked in any vacant spot. Vehicles can also be relocated from both ends of a stack, while the containers can be accessed only from the top.

The contribution of this paper is twofold. We first provide an integer program for a full information scenario in which we assume that the arrival and departure time of all AVs are known in advance. Second, we model the partial information scenario as a sequential stochastic optimization model where we assume that no information is known a priori about the arrival and departure times of AVs, and propose different heuristics to solve it.

Figure 1Schematic of AV parking proposed by Nourinejad et al. 2018.



Full information Model

Consider a parking layout with L islands. Each island has l_l rows and $2W$ stacks. The spots in an island are defined by its row number i and its stack number j. A set of $V = \{1, \ldots, v, \ldots, V\}$ AVs with arrival times A_v and departure times D_v park in this parking lot. We consider each arrival, departure, or relocation as one event t. Hence, the entire process of moving to an inter-island gap, waiting there, and parking in new spot is considered as one movement and one event. We add events $t = 1$ and $t = T + 1$ as the first and final events to the set of events in which the car-park is empty. The location of each AV $v \in V$ at time of each event $t \in T$ is determined by variables a_{vt}, b_{ijvt}, and d_{vt}. The binary variable a_{vt} indicates whether or not AV v has entered the parking lot until event t; $a_{vt} = 1$ if the AV has not arrived yet, and 0 otherwise. The binary variable d_{vt} similarly indicates the departure of an AV until event t; $d_{vt} = 1$ if the AV has already left the parking lot, and 0 otherwise. The binary variable b_{ijvt} states whether AV v is parked in spot (i, j) at event t or not. The spot where an AV is parked upon its arrival is determined by binary variable x_{ijvt}. Similarly, $z_{ijvt} = 1$ indicates that AV v leaves the parking lot from spot (i, j) at event t. Finally, if an AV is relocated during its dwell time, its origin and destination spots are determined by binary variable y_{ijhkvt}.

We formulate the vehicles' movement inside a car-park as the following binary integer program:

$$\min \sum_{i=1}^{W} \sum_{l=1}^{L} \sum_{j=1}^{l_l} \sum_{h=1}^{W} \sum_{k=1}^{l_l} \sum_{v=1}^{V} \sum_{t=1}^{T} y_{ijhkvt} \quad (1)$$

$$a_{vt} + \sum_{i=1}^{W} \sum_{l=1}^{L} \sum_{j=1}^{l_l} b_{ijvt} + d_{vt} = 1 \qquad \forall v, t \quad (2)$$

$$\sum_{v=1}^{V} b_{ijvt} \leq 1 \qquad \forall i, j, t \quad (3)$$

$$\sum_{i=1}^{W} \sum_{l=1}^{L} \sum_{j=1}^{l_l} \sum_{v=1}^{V} \left(x_{ijvt} + \sum_{h=1}^{W} \sum_{k=1}^{l_l} y_{ijhkvt} + z_{ijvt}\right) \leq 1 \qquad \forall t \quad (4)$$

$$\sum_{i=1}^{W} \sum_{l=1}^{L} \sum_{j=1}^{l_l} \sum_{v=1}^{V} \left(x_{ijvt} + \sum_{h=1}^{W} \sum_{k=1}^{l_l} y_{ijhkvt} + z_{ijvt}\right) \leq \sum_{i=1}^{W} \sum_{l=1}^{L} \sum_{j=1}^{l_l} \sum_{v=1}^{V} (x_{ijvt-1} + \sum_{h=1}^{W} \sum_{k=1}^{l_l} y_{ijhkvt-1} + z_{ijvt-1}) \forall t \in T \backslash \{1\} \quad (5)$$

$$a_{vt} = a_{vt-1} - \sum_{i=1}^{W} \sum_{l=1}^{L} \sum_{j=1}^{l_l} x_{ijvt-1} \quad \forall v, t \in T\backslash\{1\} \tag{6}$$
$$b_{ijvt} = b_{ijvt-1} + x_{ijvt-1} + \sum_{h=1}^{W} \sum_{l=1}^{L} \sum_{k=1}^{l_l} (y_{hkijvt-1} - y_{ijhkvt-1}) - z_{ijvt-1} \quad \forall i, j, v, t \in T\backslash\{1\} \tag{7}$$
$$d_{vt} = d_{vt-1} + \sum_{i=1}^{W} \sum_{l=1}^{L} \sum_{j=1}^{l_l} z_{ijvt-1} \quad \forall v, t \in T\backslash\{1\} \tag{8}$$
$$\sum_{v=1}^{V} \sum_{j=j+1}^{l_l} b_{ijvt} \leq V f_{ijt} \quad \forall i, j = 1, \dots, l_l - 1, t \in T \tag{9}$$
$$\sum_{v=1}^{V} \sum_{j=1}^{j-1} b_{ijvt} \leq V r_{ijt} \quad \forall i, j = 2, \dots, l_l, t \in T \tag{10}$$
$$\sum_{v=1}^{V} (b_{ijvt} + b_{ijvt-1}) \leq 2(1 - u_{ijt}) \quad \forall i, j = 2, \dots, l_l - 1, t \in T\backslash\{1\} \tag{11}$$
$$\sum_{v=1}^{V} (b_{ijvt} + b_{ijvt-1}) - f_{ijt} - r_{ijt} \geq -2u_{ijt} \quad \forall i, j = 2, \dots, l_l - 1, t \in T\backslash\{1\} \tag{12}$$
$$\sum_{t=1}^{T+1} a_{vt} + 1 \leq \sum_{t=1}^{T+1} a_{v't} \quad \forall v, v' \,|A_v < A_{v'} \tag{13}$$
$$\sum_{t=1}^{T+1} d_{vt} \geq \sum_{t=1}^{T+1} d_{v't} + 1 \quad \forall v, v' \,|D_v < D_{v'} \tag{14}$$
$$\sum_{t=1}^{T+1} a_{vt} + 1 \leq \sum_{t=1}^{T+1} (1 - d_{v't}) \quad \forall v, v' \,|A_v < D_{v'} \tag{15}$$
$$\sum_{t=1}^{T+1} (1 - d_{vt}) + 1 \leq \sum_{t=1}^{T+1} a_{v't} \quad \forall v, v' \,|D_v < A_{v'} \tag{16}$$
$$\sum_{v=1}^{V} a_{v1} = V \tag{17}$$
$$\sum_{v=1}^{V} d_{vT+1} = V \tag{18}$$
$$a_{vt}, b_{ijvt}, d_{vt}, f_{ijt}, r_{ijt}, u_{ijt}, x_{ijvt}, y_{ijhkvt}, z_{ijvt} \in \{0,1\} \forall v, t, i, j, h, k. \tag{19}$$

The objective function 1 minimizes the total number of movements inside the car-park. We only count the relocation movements and do not consider the $2V$ arrivals and departures because they are fixed and does not change with parking spot allocation. Eq. 2 ensures that each AV is present at exactly one location at any time. Eq. 3 ensures that each parking spot is at most occupied with one AV. Eq. 4 ensures that at each event t only one AV is moving, and all other AVs remain in their previous locations. Eq. 5 ensures that each relocation happens at the earliest possible time. Eq. 6 ensures that AV v must be parked upon its arrival. Eq. 7 describes the changes in the status of each parking spot based on the movement which is happened in the previous event. Eq. 8 presents the departure movements. Eq. 9 to 11 ensure that if a vehicle arrives or leaves from spot (i, j), then there is no vehicle in front or behind that spot. The left-hand side of Eq. 9 checks all the spots in front of spot (i, j), and if any vehicle is parked in those spots then f_{ijt} is 1. Eq. 10 is similar to Eq. 9 for spots behind spot (i, j). If a vehicle arrives or leaves a spot, then the left-hand side of Eq. 12 is equal to 1, and u_{ijt} should be equal to 0. Hence, the right-hand side of Eq. 11 is 0, and f_{ijt} and r_{ijt} cannot be both equal to 1. Eq. 13, 14, 15, and 16 ensure the correct sequence between arrival and departure for any AV. Eq. 17 ensures that no AV has arrived at time 1 and the parking is empty. Also, Eq. 18 ensures that all AVs left the parking at time $T + 1$. Constraint 19 is the binary variable constraint.

This problem is a binary integer program which is classified as NP-hard (Papadimitriou,1981).The problem can be solved by commercial solvers such as CPLEX for small size instances.However, the number of decision variables and constraints for a real-sized car-park is large, and the problem cannot be solved to its global optimum in a reasonable time.Moreover, it is unlikely to know exact arrival and departure times of all vehicles in advance, and there is always uncertainty for reasons such as congestion or minor changes in user schedules. Therefore, we investigate the partial information case in the following section.

Partial information model

In the partial information scenario, we assume that V AVs arrive and leave the car-park dynamically and randomly over the course of a finite and continuous time horizon $T = [0, |T|]$. AVs arrive at the parking facility randomly following a Poisson distribution with arrival rate λ which is unknown to the parking operator. Each AV declares its average dwell time, denoted by μ_v, but its exact departure time is declared later when it is summoned following an exponential distribution. We assume with no loss of generality that the time the car-park operator becomes aware of request r_v is concurrent to an AV's arrival or departure time, and there is no lag, consequently.

We now define the basic elements of the sequential stochastic optimization model using Powell (2007): At any point in time $t \geq 0$, each AV $v \in V$ has a physical location $p_v(t)$. The physical location can have one of the three values $p_v(t) = \{0, (i,j), 1\}$ which shows that AV v has not entered the parking lot yet, is parked at spot (i,j), or has left the parking lot, respectively. Then, the state of the system at time t, denoted by S_t, is defined by a tuple containing the location of all AVs $\{p_v(t) | \forall v \in V\}$. The exogenous information, denoted by W_t, states the arrival or departure of an AV with request time $r_v = t$. The decision variable $x_t = x_{vij}(t)$ denotes parking AV $v \in V$ at vacant spot (i,j) at time t. A policy $\pi \in \Pi$, defines the spot allocation decision given the state of the system, $X^{\pi}(S_t) \rightarrow x_{ij}(t)$. The goal of the sequential stochastic optimization problem is to find an optimal policy π from the set of all policies Π that minimize the following:

$$min_{\pi \in \Pi} E^{\pi} \sum_{t=0}^{T} C(S_t, X^{\pi}(S_t)), \quad (20)$$

where $C(.)$ is a cost function that counts the number of relocations; $S_{t+1} = S^M(S_t, x_t, W_{t+1})$; $S^M(.)$ denotes the transition function; $X^{\pi}(S_t)$ is policy π's decision function given state S_t; and W_{t+1} denotes the exogenous information that enters the systems between t and $t+1$. The objective function includes an expectation $E^{\pi}(.)$ because W_t and therefore $S^M(S_t, x_t, W_{t+1})$ are random.

The objective is to park and relocate AVs efficiently as they arrive and leave randomly and dynamically to minimize the number of movements. Since the arrival and departure times are stochastic and unknown a priori, this problem is a sequential stochastic optimization problem Powell (2007).

An exact solution for this problem would be challenging to find due to its large state space Powell (2007). Hence, we test and compare different policies using a simulation model of the car-park.We describe these policies in following sections. In all policies, when an AV is relocated to retrieve a blocked one, we treat it like a new arrival. Hence, we find a new spot for the relocated AVs instead of returning them to their previous stack.

Arrival policies

This section describes how each arrival policy allocates a parking spot to a newly arrived AV or a relocating one. We categorize these policies into three groups: arrival time which only considers the arrival time, clustering which clusters vehicles to short-term and long-term based on their dwell time, and blockage probability which calculates the probability that a blocked AV leaves earlier.

Arrival time policies

The first two policies do not consider the dwell time of AVs. The first policy, denoted by $A1$, starts parking from the smallest island and fills it from the central rows regardless of vehicle departure time. When the smallest island becomes full, vehicles are parked in the second smallest available island. The process continues until the car-park becomes full. The second policy, denoted by $A2$, starts from central rows of the smallest island, and continues by parking vehicles in central rows of bigger islands. When all such rows are full, the second policy starts to block parked AVs with new arrivals. This policy tries to minimize the number of AVs that is blocked by a new arrival, however, it does not consider the dwell time of vehicles.

Clustering policies

By knowing the average dwell time of vehicles, policies three to six first cluster AVs based on their dwell time into short-term and long-term, and then park them in different ways. The third policy, denoted by $C1$, starts from the smallest island, but short-term parking vehicles are parked on one side of the island and vehicles that dwell for more extended periods are parked on the other half of the island. Again, when the smallest island becomes full, new arrivals are parked in the second smallest island, and so on. If there is no available spot on the side that coincides with vehicle's dwell time, but there is an available spot on the other half, the spot is allocated disregarding the dwell time category.

The fourth policy, denoted by $C2$, again stores short-term parking vehicles on one side of the islands, and long-term vehicles on the other half. However, it starts from the smallest island for the short-term parking, and from the largest island for long-term ones.

The fifth arrival policy, denoted by $C3$, starts to park short-term parking from the smallest island, and long-term parking from the biggest island. The difference between this policy and $C2$ policy is that the whole island is assigned to vehicles.

The last clustering policy, denoted by $C4$, starts parking from central rows of all islands similar to$A2$ policy. However, if there are long-term vehicles in the central rows, and a new short-term parking vehicle arrives, this policy parks it in front of long-term vehicle to save the central rows space for future long-term parking vehicles.

Blockage probability policies
To further decrease the number of relocations, we can calculate the probability that a vehicle is blocked by another one knowing their average dwell times. Policies $B1$, $B2$, and $B3$ consider blockage probability. Policies $B1$ and $B2$ are similar to$A1$ and $A2$ in terms of choosing the optimal island, respectively. However, policies $B1$ and $B2$ consider the blockage probability, and allocate the spot with the lowest blockage probability to a new arrival. Let AV_1 and AV_2 be two independent AVs. AV_2 has just arrived and might be parked in front of AV_1 which arrived t units of time earlier. Let D_1 and D_2 denote their dwell times that follow an exponential distribution with average dwell times of μ_1 and μ_2, respectively. Then, the probability that AV_2 has not left the parking lot yet, when AV_1 is summoned, can be calculated as follow:

$$\begin{aligned} &P(D_2 \geq D_1 - t) = 1 - P(D_1 - t > D_2) \\ &= 1 - \int_t^{\infty} \int_0^{D_1 - t} f_{D_1,D_2}(D_1, D_2) dD_1 dD_2 \\ &= 1 - \int_t^{\infty} \int_0^{D_1 - t} \frac{1}{\mu_1 \mu_2} e^{-\frac{D_1}{\mu_1}} e^{-\frac{D_2}{\mu_2}} dD_1 dD_2 \\ &= 1 - \frac{\mu_1}{\mu_1 + \mu_2} e^{-\frac{t}{\mu_1}}, \end{aligned} \tag{21}$$

where $f_{D_1,D_2}(D_1, D_2)$ is the joint probability density function of D_1 and D_2. Also, if there is more than one vehicle in the stack, the minimum of their dwell times also follows an exponential distribution, and Eq. 21 can be used:

$$D_i \sim Exp(\frac{1}{\mu_i}) \rightarrow min\{D_i\} \sim Exp(\sum_i \frac{1}{\mu_i}). \tag{22}$$

The last policy based on blockage probability, denoted by $B3$, consider all available spots in the whole parking lot, and assigns a new arrival to the spot with the lowest blockage probability.

Relocation policies
We now investigate the impact of retrieving a blocked AV from rear side. All retrieval policies move the blocking AVs in front of a summoned AV to let it leave the car-park. However, there is possibility that fewer vehicles parked behind a summoned AV, and it can be retrieved from the rear side by fewer movements. Vehicles parked on the other side of the island would move to let the summoned AV leave the parking lot.

Numerical example

All proposed policies are simulated in Matlab and tested to compare their performance. Since there is no benchmark layout, we use some of the layouts that are presented in [14]. For each instance, the number of generated vehicles is twice the parking size, and we calculate the number of relocations needed to retrieve all vehicles. If the parking lot is full when a vehicle arrives, it is rejected and is not considered again. Due to the stochasticity of arrival and departure time of vehicles, the simulation of each instance is run 100 times.Different factors such as parking layout, number of stacks in each island, average inter-arrival gap, short-term dwell time definition, and proportion of short-term versus long-term vehicles have an impact on policy performance. We first fix the number of stacks at $W = 10$, average inter-arrival time at $\frac{1}{\lambda} = 1[min]$. We consider vehicles that are parking less than 4 hours as short-term and more than 4 hours as long-term, and there is 50% of each type. Table 2 shows the average number of relocation movements for different layouts. Results show that policy $B3$ has the lowest number of relocation movements when all the islands are sizeable, and there is no two-row or four-row island. Table 2 also shows that policies $B1$ and $B2$ do not necessarily outperform $A1$ and $A2$, although they consider the blockage probability. The logic behind it is that $A1$ and $A2$ policies consider future arrivals by parking vehicles in the deepest position, while policies $B1$ and $B2$ only find the spot based on the blockage probability.

Table 1 Average number of relocations for different parking layouts. Key: (4,10x3) means there is one island with 4 rows and 3 islands with 10 rows.

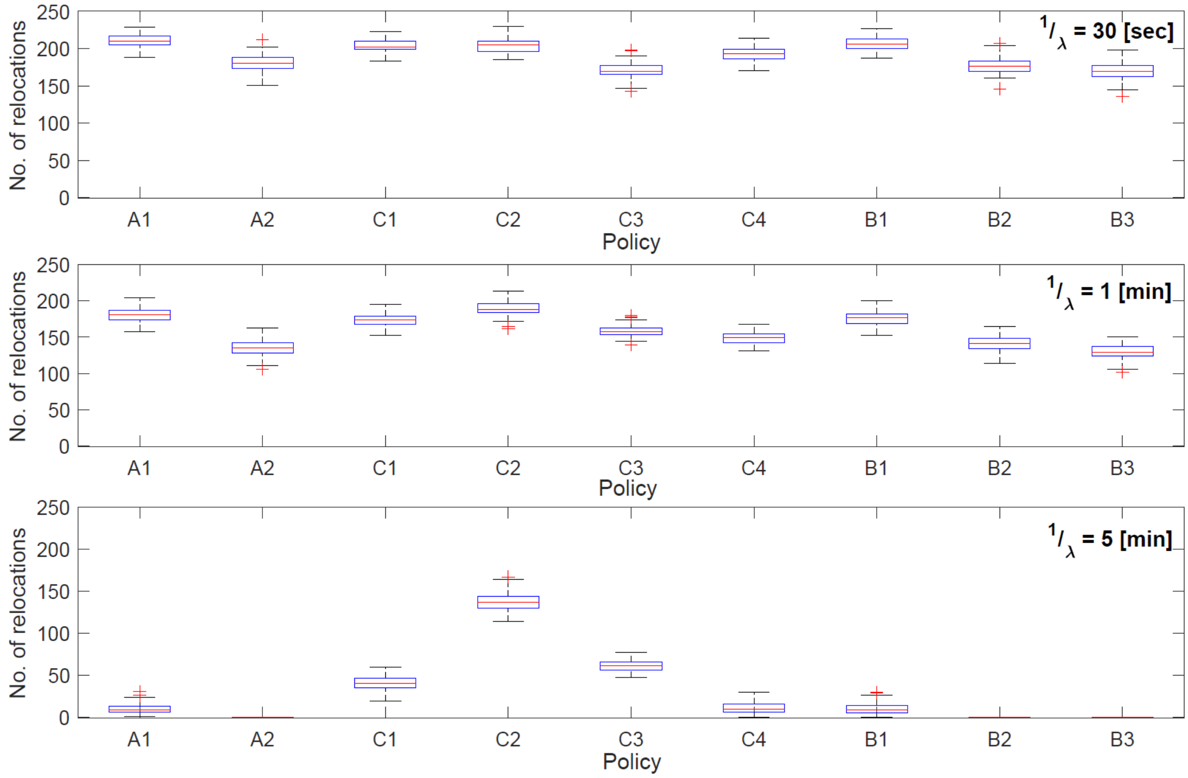
Policy / Layout	A1	A2	C1	C2	C3	C4	B1	B2	B3	Best Policy
2x2, 6x2, 8	181	135	174	189	159	149	176	141	129	B3
2, 24	405	405	398	422	363	413	408	408	408	C3
4x2, 6x2	150	111	151	153	141	120	146	110	101	B3
4, 10x3	415	241	406	427	330	247	404	286	247	A2
6x4	221	145	232	230	199	156	213	154	131	B3
8x2, 10	307	212	321	330	279	216	296	225	202	B3
12	155	155	159	159	155	155	146	146	146	B1, B2, B3
14x2	459	334	455	455	340	333	446	368	328	B3

Then, we vary the average inter-arrival gap between vehicles to see how it changes each policy performance. We consider three values of $\frac{1}{\lambda} = 30[sec]$, $1[min]$, and $5[min]$. Figures 2 shows the results. We can see that as the inter-arrival gap increases the number of relocations decreases. It happens because more short-term vehicles leave before the arrival of new cars, and the parking lot usage, defined as the number of parked vehicles to the parking capacity, decreases, consequently. Also, results show that policy $B3$ outperforms other ones regardless of inter-arrival gap value. However, other policies such as $A2$ and $B2$ have same results when the average inter-arrival gap increases to 5 minutes.

Conclusion

This paper investigates different parking and relocation policies inside an AV car-park where vehicles are stacked behind each other to increase land utilization. If all AV arrival and departure times are known in advance, the problem can be modelled as a binary integer program. Not only it is unlikely to have such information a priori, but also the integer program cannot be solved in reasonable time due to its numerous binary variables. Therefore, we also consider random arrival and departure of AVs and model it as a sequential stochastic optimization problem. This problem is solved by simulating different policies. The simulation results show that assigning the vehicles to the spot with the lowest blocking probability is the best scenario when all the islands are sizeable, and there is no two-row or four-row island. This policy outperforms others greatly when the arrival rate is high. We find that considering blockage probability is not necessarily better than parking in the deepest available spot because it does not consider the future arrivals. Also, results show that considering the possibility of retrieving vehicles from the rear side does not reduce the number of relocation movements in a parking lot.

Figure 2: Changes in number of relocation movements relative to average inter-arrival gap for 2x2, 6x2, 8 layout.



References

Christian Bierwirth and Frank Meisel. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3):675–689, 2015.
SetarehBorjian, Vahideh H Manshadi, Cynthia Barnhart, and Patrick Jaillet. Dynamic stochastic optimization of relocations in container terminals. *Accessed*, 9:2017, 2013.
Marco Caserta, Silvia Schwarze, and Stefan Voß. A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal of Operational Research*, 219(1):96–104, 2012.
Virgile Galle, VH Manshadi, S BorjianBoroujeni, Cynthia Barnhart, and Patrick Jaillet. The stochastic container relocation problem. *Transportation Science*, 2018.
M HakanAkyüz and Chung-Yee Lee. A mathematical formulation and efficient heuristics for the dynamic container relocation problem. *Naval Research Logistics (NRL)*, 61(2):101–118, 2014.
Dusan Ku and Tiru S Arthanari. Container relocation problem with time windows for container departure. *European Journal of Operational Research*, 252(3):1031–1039, 2016.
Jana Lehnfeld and Sigrid Knust. Loading, unloading and premarshalling of stacks in storage areas: Survey and classification. *European Journal of Operational Research*, 239(2):297–312, 2014.
Alex Mitchell. Are we ready for self-driving cars? *World Economic Forum*, 2015.

Mehdi Nourinejad and Mahyar Amirgholy. Parking pricing and design in the morning commute problem with regular and autonomous vehicles. 2018.

Mehdi Nourinejad, Sina Bahrami, and Matthew J Roorda. Designing parking facilities for autonomous vehicles. *Transportation Research Part B: Methodological*, 109:110–127, 2018.

Christos H Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981.

Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

Robert Stahlbock and Stefan Voß. Operations research at container terminals: a literature update. *OR spectrum*, 30(1):1–52, 2008.

Dirk Steenken, Stefan Voß, and Robert Stahlbock. Container terminal operation and operations research-a classification and literature review. *OR spectrum*, 26(1):3–49, 2004.

TechWorld. The huge impact driverless cars will have on parking and urban landscapes, Jul 2016.

Elisabeth Zehendner, Marco Caserta, Dominique Feillet, Silvia Schwarze, and Stefan Voß. An improved mathematical formulation for the blocks relocation problem. *European Journal of Operational Research*, 245(2):415–422, 2015.

Wenjuan Zhao and Anne V Goodchild. The impact of truck arrival information on container terminal rehandling. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):327–343, 2010.