GENERALIZING THE COMBINATORIAL LANE MERGING PROBLEM

AURORA HIVELEY

Abstract. Picture a sequence of cars approaching a road with two lanes. Each car has a preference for which lane to join: either the rightmost lane, or the shortest of the two lanes. We can represent each car's preference with a digit, 1 or 2, and calculate the expected length of each lane once the sequence has been resolved. Now, consider a generalization of this problem to three lanes, or k total lanes. We give recurrences for the number of arrival sequences which end with x_1 cars in the right lane, x_2 cars in the second lane, and so on. We also provide a mapping to standard Young tableaux, an equivalent counting for voting sequences, and a closed formula for the number of arrival sequences on $[k]^n$ which end with lane lengths $(x_1, x_2, \ldots, x_k)$ given from right to left.

1. Introduction

Consider a road with two lanes of traffic. The road approaches a stoplight, and shortly after the stoplight the left lane will merge into the right lane. An individual driving a car on this road must make a decision as they approach the stoplight: which lane will they occupy? The driver may prefer to stay in the right lane so that they need not worry about merging, or they may prefer to occupy the lane with fewer cars so that they can make it through the light as quickly as possible. This problem is referred to as a combinatorial lane merging problem, and it has been studied previously by [1] and [2].

In this problem, we will represent each car's preference with a single digit: a 1 if the car prefers the right lane, or a 2 if the car prefers the shorter of the two lanes. In the case of a tie, a car denoted by a 2 will opt for the right lane, so the right lane is always at least as long as the left lane as a result. Observe, then, that at any point during the resolution of the arrival sequence, the lane lengths are weakly decreasing from right to left. A car can only join the left lane if the right lane is strictly longer, so the left lane can never be longer than the right lane.

Definition 1. A 2-lane *arrival sequence* is a binary sequence of 1's and 2's representing the preferences of all cars in a lane merging problem.

Example 2. Let A be the arrival sequence $A = 11222121$. Car 1's preference is the right lane, so Car 1 moves to the right lane. Car 2's preference is also the right lane, so it also joins the right lane. Car 3 prefers the shorter of the two lanes, and since no cars are in the left lane, Car 3 joins the left lane. The same occurs for Car 4. Then Car 5 prefers the shortest lane, but both lanes have two cars in them, so Car 5 opts for the right lane to resolve the tie. Next, Car 6 moves to the right lane, and Car 7 will go to the left lane since there are only two cars there compared to four cars in the right lane. Lastly, Car 8 goes to the right lane. Then the final configuration is $[1, 2, 5, 6, 8]$ in the right lane and $[3, 4, 7]$ in the left.

Bardenova et al. [1] is primarily concerned with computing the expected length of the right lane for an arrival sequence of length n (of course, the length of the left then follows immediately). In their work, they represent a lane merging problem with a 2-dimensional lattice path, where the vertical component is incremented when a car joins the right lane and the horizontal component is incremented when a car joins the left lane. Then an arrival sequence with y cars in the right lane and x cars in the left lane will end at the point (x, y). By counting these lattice paths, the authors were able to determine a recurrence and eventually a closed formula for the number of cars y in the right lane after the sequence has fully resolved. From there, computing the expectation and studying the asymptotics is straightforward. Additional analysis of the lane merging problem with limits on the capacity of each lane was also discussed in [2].

Bardenova et al. [1] also propose a generalization of their work which considers three lanes of traffic. They suggest extending the definition of an arrival sequence as follows:

- A 1 denotes a car whose preference is the right lane.

- A 2 denotes a car whose preference is the shorter of the two rightmost lanes.
- A 3 denotes a car whose preference is the shortest of all three lanes.

This generalization will be the focus of our work. We begin by developing a recurrence, which is then generalized to 3 lanes, and eventually an arbitrary number k total lanes. These recurrences eventually give rise to a generalized closed formula, which is finally used to analyze the lane lengths from a statistical perspective.

2. Three Lane Problem

Previously, Bardenova et al. mapped two-lane arrival sequences to lattice paths in two dimensions. In these paths, the vertical components was incremented whenever a car joined the right lane, and the horizontal component was incremented whenever a car joined the left lane. In this sense, the lattice path never crosses below the diagonal since the left lane can never be longer than the right lane. As we generalize this work to three or more lanes, we note the difficulties visualizing and drawing lattice paths in three or more dimensions. So, we instead pivot to a different visual representation: a standard Young tableau.

2.1. **Standard Young Tableau.**

A Young diagram, sometimes called a Ferrers diagram when represented with dots instead of boxes, is a left-justified arrangement of boxes in rows such that no row is longer than any row above it. A standard Young tableau, henceforth called an SYT, is a Young diagram whose boxes have been populated with the integers $\{1, 2, \ldots, n\}$ where n is the number of cells in the diagram, each row is increasing from left to right, and each column is increasing from top to bottom [5]. Figure 2.1 depicts a Ferrers diagram and one possible SYT that can be obtained from its population with numbers. The shape, or underlying Ferrers diagram, of a SYT can be written $L = (l_1, l_2, \ldots, l_k)$ where l_i is the length of the i-th row of the SYT and $l_i \geq l_j$ for all $i < j$. The number of SYT of a given shape is well understood and has been extensively studied in [4] and [5], to name just two sources.

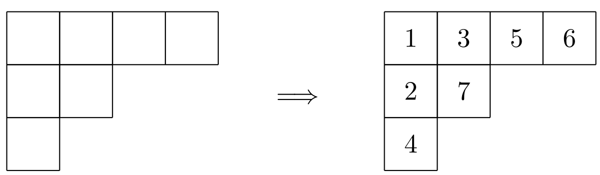

Figure 1. An example Young diagram on $n = 7$ (left) and the same Young diagram filled in to form a standard Young tableau (right) of shape $L = (4, 2, 1)$.

Since the lanes in our arrival sequences must all be weakly decreasing in length from right to left, the rows of a SYT are ideal for representing the final configuration of lanes in an arrival sequence. The cars arrive at the intersection in numerical order, so of course each row will be increasing. Furthermore, a car can only join a lane further to the left if there are at least that many cars in all lanes to the right, so the columns will also be increasing. We also note that using a SYT to represent an arrival sequence offers greater ease of visualization since an additional lane of traffic is modeled with a new row of the tableau rather than a new dimension for the lattice path to traverse.

2.2. **Tableau Enumeration.**

Originally, Bardenova et al. [1] observed that the map from two-lane arrival sequences to lattice paths was not one-to-one. For example, regardless of what the first car's preference is, it will always join the right lane. Then there are at least two sequences which map to the same lattice path: the sequence $1 + S$ and the sequence $2 + S$, where S is the rest of the sequence. The precise enumeration of how many sequences map to a lattice path is detailed in [1] in an effort to enumerate arrival sequences' ending configurations.

Our new mapping to SYT is no different: there are still several arrival sequences which map to a single SYT. Crucially, the multiplicity of each visualization is enumerated by considering each time that a tie break occurs. In Bardenova et al. [1], the authors define a "bounce" as a place where the lattice path makes contact with the diagonal, but a tie break forces the lattice path to move upwards, effectively "bouncing" off of the main diagonal.

Definition 3. A *bounce* is a location in the arrival sequence where a car with preference $i > 1$ opts for a lane further to the right of i because of a tie between lane lengths.

In the language of SYT, we see that a "bounce" occurs when two or more rows have the same length, and the n-th car is added to the highest row with a tied length. Consider the resolution of the arrival sequence 12131233 illustrated in Figure 2.2. While Car 8 is willing to join whichever lane is the shortest, since lanes 2 and 3 are tied in length with two cars each, Car 8 "bounces" and lands in lane 2, like so:

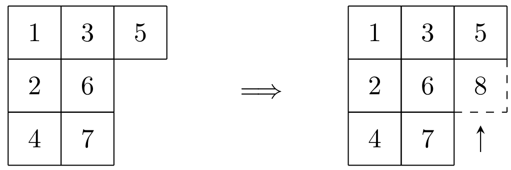

FIGURE 2. Resolution of Car 8 in the arrival sequence 12131233, which "bounces" into lane 2.

Then to count the number of arrival sequences which resolve to a single SYT, we must keep track of the number of bounces which occur during its resolution, and the number of rows which are tied at each bounce. This formula is somewhat ill-defined, but keep in mind that the number of ties rows will give rise to the coefficients in the recurrences in the next section.

Proposition 4. *The number of k-lane arrival sequences which resolve to a SYT T with k rows is given by:*

$$\#A(T) = \prod_{c \in T} \beta(c)$$

where $c = [i, j]$ is a cell in the tableau T and

$$\beta(c) := \begin{cases} \#\{c' = [i', j-1] \mid i' > i, T[c'] < T[c]\} & j > 1 \\ k - i & j = 1 \end{cases}$$

Example 5. Consider the SYT to the left. We calculate $\beta(c)$ for each of the five cells as follows:

$$\begin{aligned} \beta([1,1]) &= 3 \\ \beta([1,2]) &= 2 \\ \beta([1,3]) &= 1 \\ \beta([2,1]) &= 2 \\ \beta([3,1]) &= 1 \end{aligned} \qquad \begin{array}{|c|c|c|} \hline 1 & 3 & 5 \\ \hline 2 \\ \cline{1-1} 4 \\ \cline{1-1} \end{array}$$

Then $\#A(T) = 3 \cdot 2 \cdot 1 \cdot 2 \cdot 1 = 12$. The twelve 3-lane arrival sequences which resolve to this SYT are:

12131	13131	12231	13231
22131	23131	22231	23231
32131	33131	32231	33231

Proof. We will count the number of arrival sequences which can be reverse engineered from a tableau T. Start first with the final cell added to the tableau, or the final car to choose a lane: n. Say that Car n joins lane i for $1 \leq i \leq k$. How many possible preferences could Car n have had which cause it to join lane i?

Certainly if Car n's preference is any digit smaller than i, then lane i would never be considered by the car, so we can narrow our search space to $\{i, i+1, \ldots, k\}$. However, a preference of $i+1$ will only cause Car n to join lane i if the lanes i and $i+1$ have tied lane lengths. In the tableau T, this means that there are already digits occupying rows i and $i+1$ in the column to the left of the element n. Extrapolating, we see that a preference of $i+r$ will only cause Car n to join lane i if *all* lanes $\{i, i+1, \ldots, i+r\}$ have tied lengths, so all rows $\{i, i+1, \ldots, i+r\}$ of T have entries in the column to the left of n. Then the number of possible preferences that Car n could have which would cause it to join lane i is equal to $\beta(c)$ where $c = [i, j]$ is the location of Car n. In this sense, $\beta(c)$ counts the number of dimensions that the merging path would "bounce" in, or the number of possible digits that Car $n = T[c]$ could have that would still cause it to enter lane i and thus fill location $[i, j]$ in T. Once we have determined the number of possible digits in position n of an arrival sequence which resolves to T, we can "peel off" $n = T[c]$ from T, and consider all possible preferences that Car $n-1$ can have in the same fashion.

Note, however, that any entry in the first column of T has no column to the left to compare to. But the first car to join any lane can prefer any open lane to the left, and it will still join lane i. For example, as noted before, Car 1 can prefer any of the k lanes, and it will always join the first lane. If Car 2 joins the second lane, then it could have preferred any of the $k-1$ open lanes to the left, and it would still join lane 2. In general, a car in the first column and the i-th row could have $k-i$ preferences corresponding to the empty lanes to the left. Hence, $\beta(c)$ is defined differently for $c = [i, j]$ when $j = 1$.

Repeating this process of counting possible preferences for all n cars, or counting $\beta(c)$ for all $c \in T$, we have that the number of possible arrival sequences is obtained from the product of the number of possible preferences for each car, which is $\prod_{c\in T} \beta(c)$. $\square$

Remark 6. The product $\prod_{a\in T} \beta(a)$ where $a = [i, 1]$ for some $1 \le i \le k$ is:

$$\prod_{a\in T} \beta(a) = k \cdot (k-1) \cdot \dots \cdot (k-m+1) = k^{\underline{m}}$$

where m is the number of rows filled in T. This means that $\#A(T)$ will always be divisible by k.

Note that there may be empty lanes in an arrival sequence, which translate to empty rows in a SYT. For example if A is a 3-lane arrival sequence which contains only 1's and 2's, then cars will only merge into one of the two rightmost lanes, and thus the resulting SYT will only have numbers filled in to the top two rows. In this example, $m = 2$.

2.3. **Recurrences.** In an attempt to develop a closed formula for the number of arrival sequences that end at a given configuration, we will first concern ourselves with constructing recurrences. The recurrence for two lane arrival sequences has previously been studied in [1], but we include it here as well for the sake of completeness, and to show how the calculation extrapolates to additional lanes.

Proposition 7. *The recurrence for a 2-lane merging sequence with x cars in the right lane and y cars in the left lane is:*

$$F(x,y) = \begin{cases} F(x,y-1) + F(x-1,y) & x-y>1 \\ F(x,y-1) + 2F(x-1,y) & x-y=1 \\ F(x,y-1) & x=y \end{cases}$$

Noting that $F(x, y) = 0$ if $y > x$, and $F(0, 0) = 1$.

This proposition is equivalent to Lemma 5 of [1], and the author refers the reader to the proof in the previous paper. Our work begins by adapting this recurrence to fit arrival sequences with three lanes.

Proposition 8. *The recurrence for a 3-lane merging sequence with x cars in the right lane, y cars in the middle lane, and z cars in the left lane is:*

$$F(x,y,z) = \begin{cases} F(x-1,y,z) + F(x,y-1,z) + F(x,y,z-1) & x-y>1 \text{ and } y-z>1 \\ 2F(x-1,y,z) + F(x,y-1,z) + F(x,y,z-1) & x-y\le 1 \text{ and } y-z>1 \\ F(x-1,y,z) + 2F(x,y-1,z) + F(x,y,z-1) & x-y>1 \text{ and } y-z\le 1 \\ 2F(x-1,y,z) + 2F(x,y-1,z) + F(x,y,z-1) & x-y=1 \text{ and } y-z=1 \\ 3F(x-1,y,z) + 2F(x,y-1,z) + F(x,y,z-1) & o.w. \end{cases}$$

Noting that $F(x, y, z) = 0$ if $z > y$ or $y > x$, and $F(0, 0, 0) = 1$.

We will defer our proof of this proposition and instead direct the reader towards the proof of the more general recurrence in the next section. In the meantime, we offer the following examples:

Example 9. If the ending configuration of our three lane arrival sequence is $(4, 2, 1)$, then there are three possible lanes which the seventh (last) car could have joined:

(1) If Car 7 joined the right lane, then the previous configuration would have been $(3, 2, 1)$. This outcome occurs if Car 7's preference is a 1, as any other preference would cause the car to pick one of the shorter lanes.
(2) If Car 7 joined the middle lane, then the previous configuration would have been $(4, 1, 1)$. This outcome occurs if the car's preference was either a 2 or a 3. If the car's preference was a 2, then the car chooses the shorter of the right and middle lanes, which is, of course, the middle lane. If the car's preference was a 3, then it picks the shortest of all lanes, opting for the rightmost lane in case of a tie. Then the middle lane is selected instead of the left, even though their lengths are the same.
(3) If Car 7 joined the left lane, then the previous configuration would have been $(4, 2, 0)$. The only preference which produces this outcome is a 3.

Then the overall recurrence is $F(4, 2, 1) = F(3, 2, 1) + 2F(4, 1, 1) + F(4, 2, 0)$.

Example 10. If the ending configuration of our three lane arrival sequence is $(3, 3, 2)$, then the final car can only have joined two of the possible lanes: the middle or the left. If the final car joined the right lane, that would mean that the previous configuration was $(2, 3, 2)$, which is not possible since the lane lengths must be weakly decreasing from right to left. Then we only have two cases:

(1) If the final car joins the middle lane, then the previous configuration was $(3, 2, 2)$. Then Car 8's preference could have been either a 2 or a 3, and in either case the car would have chosen the middle lane, either by breaking a tie or by selecting the shorter of two lanes.
(2) If the final car joins the left lane, then the previous configuration was $(3, 2, 1)$. This means that Car 8's preference must have been a 3.

This means that $F(3, 3, 2) = 2F(3, 2, 2) + F(3, 2, 1)$.

3. Further Generalizations

The framework established by Bardenova et al. [1] can be generalized even further to arrival sequences with k lanes of traffic, and we note that such a sequence can be represented with a standard Young tableau consisting of k rows. We introduce the following revised definition of an arrival sequence for an arbitrary number of lanes, k:

Definition 11. An *arrival sequence* is a sequence in $\{1, \ldots, k\}^n$ where each digit represents a car's preference. A 1 represents a car which will only enter the right lane, a 2 will enter the shortest of the two rightmost lanes (opting for the right in case of a tie), and generally an i will enter the shortest of the i rightmost lanes, opting for the rightmost tied lane in the case of a tie.

Theorem 12. *The general recurrence for a k-lane arrival sequence with lanes $x_1, x_2, \ldots, x_k$ (from right to left) is:*

$$F(x_1, \ldots, x_k) = \sum_{i=1}^{k} c_i \cdot F(x_1, \ldots, x_i - 1, \ldots, x_k)$$

where c_i is defined:

$$c_i := 1 + |\{j > i \mid x_j = x_i - 1\}|$$

and $F(0, \ldots, 0) = 1$, and $F(x_1, \ldots, x_k) = 0$ if $x_i < x_j$ for any $1 \le i < j \le k$.

Example 13. $F(4, 3, 3, 2) = 3F(3, 3, 3, 2) + 2F(4, 3, 2, 2) + F(4, 3, 3, 1)$

(1) First, $c_1 = 1 + 2 = 3$ since x_2 and x_3 are equal to $x_1 - 1 = 3$.
(2) Secondly, note that $F(3, 2, 3, 2) = 0$ since $x_2 < x_3$. So this term does not appear in the recurrence.
(3) Thirdly, $c_3 = 1 + 1$ since $x_4 = x_3 - 1 = 2$. So $j = 4$ is the only entry in the set counted by c_3.
(4) Lastly, $c_4 = 1 + 0 = 1$ since there are no $j > 4$ to be considered in the set $\{j > i \mid x_j - x_i - 1\}$.

Proof. of **Theorem 12**.
To produce an arrival sequence of length n from an existing arrival sequence of length $n - 1$, the n-th car must join one of the existing k lanes. Let $X = (x_1, x_2, \ldots, x_k)$ be the final configuration of the arrival sequence with n cars. Say that we remove the n-th car, which was in the first lane. Then the configuration for the $n - 1$ car arrival sequence is $(x_1 - 1, x_2, \ldots, x_k)$. If the n-th car was in the second lane, then the

configuration for the first $n-1$ cars was $(x_1, x_2 - 1, x_3, \ldots, x_k)$. Note, of course, that any sequence such that $x_i < x_{i+1}$ is not possible, so $F(x_1, \ldots, x_k) = 0$ for such a configuration.

For the possible configurations, it then suffices to determine how many possible preferences in the set $\{1, \ldots, k\}$ would allow the n-th car to join lane i for some fixed lane $i \in \{1, \ldots, k\}$. Say that the n-th car joins lane i. Then the smallest integer that can possibly represent its preference is i, since any smaller number would not even consider lane i as an option. For instance, a car with preference 2 considers only the first two lanes, so it would never go to the third lane, or any lane further to the left.

However, a car with a preference greater than i may also join lane i. If the car's preference was $i+1$, then it will still join lane i if lanes i and $i+1$ have the same length, since each car defaults for the rightmost lane in case of a tie. Similarly, a car with preference $i+2$ could also end up in lane i if lanes i and $i+2$ had the same length (which also implies that lane $i+1$ has this length, since the lane lengths must be weakly decreasing from right to left). Then the number of possible preferences that a car can have is equal to the number of lanes to the left of lane i which have the same length as i, plus 1 (when its preference is i).

Thus, c_i as defined above precisely counts the number of possible preferences that a car joining lane i can have: either the preference i (1) or the preference for any integer larger than i where the lane lengths are tied ($\{j > i \mid x_j - x_i - 1\}$). If the n-th car joins lane i, there are $F(x_1, \ldots, x_{i-1}, \ldots, x_k)$ ways for the previous $n-1$ cars to have lined up so that the final configuration is $F(x_1, \ldots, x_k)$. Then, summing over all possible lanes k, we have a recurrence for the number of arrival sequences of length n which resolve to the ending configuration $(x_1, x_2, \ldots, x_k)$. □

Definition 14. A *majority voting sequence* $V \in \{1, \ldots, k\}^n$ is a sequence such that each digit represents a vote cast for one of k total candidates. A *state* of the voting sequence $(p_1, p_2, \ldots, p_k)$ is defined such that after $\sum_{i=1}^{k} p_i$ votes have been cast, there are p_1 votes for the most popular candidate, p_2 votes for the second most popular, and so on.

Theorem 15. *Majority voting sequences satisfy the same recurrence as Theorem 12.*

Proof. Say that $n-1$ votes have been cast so far, and our ending configuration after n votes will be $(p_1, p_2, \ldots, p_k)$. If the n-th voter votes for the most popular candidate, meaning that the voting configuration after $n-1$ voters was $(p_1 - 1, p_2, \ldots, p_k)$, then there are two possibilities. If the most popular candidate was already the most popular, meaning that $p_1 - 1 > p_j$ for all $j > 1$, then there is only one possible vote that voter n could have made: voting for the most popular candidate and furthering their lead. Otherwise, say that the top two candidates were tied, so $p_1 - 1 = p_2$. Then if voter n votes for *either* of the tied candidates, the new "most popular" candidate will have p_1 votes at the end of the voting day, and will clinch the lead from the tie. In general, if there are m tied candidates, then voter n could have voted for any of these m candidates, and the final voting configuration would still be $(p_1, p_2, \ldots, p_k)$. Then if we know that the first $n-1$ voters produced the voting result $(p_1, \ldots, p_i - 1, \ldots, p_k)$, the number of ways that voter n could have voted is equal to $1 + |\{j > i \mid p_i - 1 = p_j\}|$, as this is the number of candidates with $p_i - 1$ votes prior to voter n's ballot being cast. Summing over all possible candidates $i \in \{1, \ldots, k\}$ that voter n could have voted for, and noting that a voting configuration must be weakly decreasing so $p_i < p_j$ for $i < j$ means that $F(p_1, \ldots, p_i, \ldots, p_j, \ldots, p_n) = 0$, we have proven that majority voting sequences satisfy the same recurrence as in Theorem 12. □

Theorem 16. *Majority voting sequences are counted by the closed form*

$$F(p_1, \ldots, p_k) = \binom{n}{p_1, \ldots, p_k} \cdot \binom{k}{q_1, \ldots, q_r}$$

where $q_1, q_2, \ldots, q_r$ is the partition of k defined by $(p_1, \ldots, p_k)$ in frequency notation.

Example 17. Consider the majority voting sequence ending at $(4, 3, 3, 2)$. Then there are $n = 4+3+3+2 = 12$ votes that are cast for $k = 4$ candidates. Using frequency notation, we partition 4 into $q_1 = 1$, $q_2 = 2$, and $q_3 = 1$ since $(4, 3, 3, 2)$ in frequency notation is $(1, 2, 0, 1, 0)$. Then we can calculate:

$$F(4, 3, 3, 2) = \binom{12}{4, 3, 3, 2} \cdot \binom{4}{1, 2, 1}$$

Proof. of **Theorem 16**.

Say that a total of $n = p_1 + p_2 + \cdots + p_k$ votes are cast for k candidates such that the most popular candidate received p_1 votes, the second most popular receives p_2 votes, and so on.

There are certainly $\binom{n}{p_1, p_2, \ldots, p_k}$ ways that the voters can cast their votes for k candidates so that the ending vote count is precisely $(p_1, p_2, \ldots, p_k)$. However, we must also account for the permutations of the candidates by popularity. This multinomial alone assumes that Candidate 1 is the most popular, and Candidate k is the least popular. But we cannot simply permute the candidates and conclude that there are $k!$ possible final rankings, because ties are allowed! Then we convert $(p_1, p_2, \ldots, p_k)$ to frequency notation $q_1, q_2, \ldots, q_r$. Of these k candidates, we choose q_1 of them to have the most votes, or to be tied for the most votes if $q_1 > 1$. Then, of the remaining $k - q_1$, we choose q_2 to receive (or be tied for) the second most votes, and so on. This product of binomials is precisely equal to the multinomial $\binom{k}{q_1, q_2, \ldots, q_r}$. Then majority sequences with the ending result $(p_1, \ldots, p_k)$ can be counted by the desired closed form. □

Corollary 18. *Arrival sequences are counted by the same closed form as in Theorem 16*

Proof. Since both arrival sequences and majority walks satisfy the same recurrence with the same initial conditions, both objects are counted the same way. Then the closed form for majority walks also functions as a closed form for arrival sequences. □

Observe that this closed form is equivalent to the closed form given in Theorem 6 of [1]. Their closed form for 2 lanes with lengths x and y where $x > y$ and $x + y = n$ is given as follows:

$$F'(x,y) = \begin{cases} 2\dbinom{x+y}{x} & x > y \\ \dbinom{2x}{x} & x = y \end{cases}$$

In our case, when $x > y$ we have that $k = 2$ and $p_1 = 1$, $p_2 = 1$ since each lane has a unique length. Then $F(x,y) = \binom{x+y}{x,y} \cdot \binom{2}{1,1} = \binom{x+y}{x} \cdot 2$, which matches $F'(x,y)$. Similarly, if $x = y$ in our case, then $p_1 = 2$ when $k = 2$ since each lane has the same length, and $n = 2x$ since $x = y$. So, we have that $F(x,y) = \binom{x+y}{x,y} \cdot \binom{2}{2} = \binom{2x}{x}$. Once again, this matches $F'(x,y)$, so our generalized closed form matches the previous closed form for 2-lane arrival sequences.

3.1. **Mapping Arrival Sequences and Voting Sequences.** In Section 3 of [1], the authors provide a bijection between lattice paths and coin flipping sequences. The authors map a lattice path obtained from a two-lane arrival sequence of length n with x cars in the right lane and $y = m - x$ in the left to a sequence of n coin flips where the maximum number of heads or tails is x. Their bijection is constructed as follows:

Definition 19. The bijection $\phi : \{0,1\}^n \to \{0,1\}^n$ maps an arrival sequence A to $\phi(A)$ as follows:

(1) Define the parity vector P of A as follows:
   (i) $P[1] = 0$
   (ii) For $2 \le i \le n$, if there is a bounce for Car $i-1$ of A, then $A[i] = A[i-1] + 1 \mod 2$. If there is no bounce for Car $i-1$ of A, then $A[i] = A[i-1]$.
(2) Let $C := \phi(A) = A + P \mod 2$.

Note that by adding the parity vector P to the existing binary string A, whenever a bounce occurs we effectively flip which column of the SYT a car is added to. Consider the following example.

1	3	5	7
2	4	6	8

FIGURE 3. SYT obtained by resolving A=01110111 or C = 01101001

Example 20. Let $B = 12221222$. Then the bounces occur for cars numbered 3 and 7, so the parity vector flips at locations 4 and 8, resulting in $P = 11122221$. Adding $P + A$, we obtain $C = 12212112$. Figure 3.1 depicts the SYT obtained by resolving the arrival sequence A. Note that the length of each row is 4, and there are 4 heads and 4 tails in the coin flip sequence A.

Bardenova et al. [1] suggest that this mapping may be extended to a bijection between lattice paths formed by k-lane arrival sequences and sequences of k-sided die rolls (or, equivalently, k-candidate majority voting sequences). Since we have used the language of voting sequences thus far, we will continue with this terminology, but note that it *is* equivalent to the framework suggested by [1]. We will also consider the final configuration of an arrival sequence to be a SYT rather than a merging path, to maintain consistency. So, in an effort to generalize the existing bijection, we actually seek a map between SYT with n elements and k rows and majority voting sequences with n voters for k candidates.

Of course, the results of Theorems 12 and 15 indicate that, for a fixed k and n, the ending configurations of arrival sequences and of voting sequences have the same distribution. This indicates that the set of SYT that can be obtained from arrival sequences is identical to the set of SYT obtained from voting sequences, even up to multiplicity. This suggests a natural bijection where an arrival sequence which resolves to a tableau T maps to a voting sequence which resolves to the same tableau. However, the difficulty comes from the fact that a majority sequence does not immediately resolve to a SYT. Consider the following example:

Example 21. Let $V = 111223333$. If we construct an SYT by letting all of the votes for candidate 1 fall into row 1, and the votes for candidate 2 into row 2, and so on, then we obtain the SYT on the left of Figure 3.1. Of course, this *not* a SYT as the row lengths are not weakly decreasing. Even if we sort the rows by length so that they are weakly decreasing, the columns will fail to be increasing.

1	2	3	
4	5		
6	7	8	9

6	7	8	9
1	2	3	
4	5		

Figure 4. Tableaux obtained by naively resolving the voting sequence 111223333. Observe that neither tableau is a SYT.

To produce a SYT from a voting sequence, we must also permute the rows according to which candidate has the majority number of votes at each stage. We illustrate this concept more simply by revisiting the SYT obtained by resolving the coin-flipping sequence C in Example 20. When resolving the coin flipping sequence, the number n is added to the top row if the n-th coin flip is the most frequent result thus far, or the bottom row if it is the least frequent result thus far. This resolution actually produces the same SYT as in Figure 3.1. So, we conclude that our bijection between arrival sequences and voting sequences should preserve each's SYT, and it does this by tracking which candidate has the most votes at each stage, and modifying the placement of vote n according to the current majority holder. Experimental data using procedures `kLane` and `kVote` in the Maple package linked in the Appendix 5 also supports the notion that the sets of SYT obtained from both arrival sequences and voting sequences have the same distribution, and thus a bijection using this property should exist.

For two rows, this permutation of rows is simple, as it boils down to a swap of the two rows. For three or more rows, this becomes more complicated, as we must consider *each* possible adjacent pairwise swap, and the case of a candidate being tied with more than one other candidate and thus “jumping” more than one row up the tableau. In fact, we also need to track this information over time as all votes are cast so that we can determine which candidate(s) positions must be swapped. So, while for two candidates this boiled down to the problem of parity swapping for a binary vector, the problem becomes more complicated as we must consider many permutations of k total elements.

4. Lane Length Statistics

Previously, Bardenova et al. [1] computed the expected length of the right lane for a two lane arrival sequence with n cars. More interestingly, the numerator of the expected value produces the following integer sequence, which can be found in the OEIS as A230137.

$$0, 2, 6, 18, 44, 110, 252, 588, 1304, 2934, 6380, 14036, 30120, 65260, 138712, \ldots$$

Repeating this calculation for three or more lanes, we obtain Table 1. Each of these sequences links back to A265080, which lists the entries of a table which arises from public key cryptography. Henry Bottomley's 2021 comment on this entry also indicates that these table entries count the number of balls in the fullest box if one were to randomly throw n balls into k boxes across all possible outcomes. This hearkens back to our discussion of majority voting, as the balls in boxes problem mimics the same process.

k	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	OEIS entry
2	2	6	18	44	110	252	588	A230137
3	3	12	51	192	675	2358	8043	A265083
4	4	20	108	544	2540	11544	52192	A265084
5	5	30	195	1220	7145	40230	224175	A265085

Table 1. Sum of the length of the right lane in a k-lane arrival sequence of length n

The previous authors did not study the expected length of the left lane, likely because it follows immediately from the length of the right lane. However, for three or more lanes, this calculation is less trivial, and can mean two different things: the length of the second lane from the right, or the length of the leftmost lane. In the sense of our majority voting framework, we may be concerned with the number of votes received by the second most popular candidate, or perhaps the number of votes received by the least popular candidate.

Table 2 tabulates the summed lengths of the second lane from the right across all arrival sequences of length n for k lanes, and Table 3 does the same for the leftmost lane. None of the rows in either Table 2 have entries in the OEIS [3]. Of course, the first k numbers in each row of Table 3 match up with the first k entries of the same row in A019538, but the sequences diverge once $n > k$. This makes sense, as A019538 counts ordered set partitions of n into k parts, and as long as $n \leq k$, these partitions are equivalent to the end states (SYT) of our arrival sequences.

k	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$
2	0	2	6	20	50	132	308
3	0	6	24	96	390	1386	4830
4	0	12	60	288	1500	7392	33432
5	0	20	120	680	4220	26220	151340

Table 2. Sum of the length of the second lane in a k-lane arrival sequence of length n

k	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
2	0	2	6	20	50	132	308	744
3	0	0	6	36	150	630	2436	8736
4	0	0	0	24	240	1560	8400	43344
5	0	0	0	0	120	1800	16800	126000

Table 3. Sum of the length of the leftmost lane in a k-lane arrival sequence of length n

5. Conclusion

In summary, in this paper we considered generalized arrival sequences with k lanes. We introduced a mapping from arrival sequences to standard Young tableaux, and we counted the number of sequences which map to each tableau after noting that the map is not one-to-one. We also derived recurrences for our k-lane arrival sequences, and proved that these recurrences also hold for majority voting sequences. Then we produced a closed formula for the number of majority sequences with a given final vote count, and proved that the same closed formula holds for arrival sequences by virtue of their equivalent recurrences and initial conditions. Finally, we provided tables counting lane lengths for k lane arrival sequences of length n with references to the OEIS [3], when applicable.

Of course, the mapping between k-lane arrival sequences and majority voting sequences (or, equivalently, sequences of k-sided dice rolls) remains open, and may be tractable with methods not discussed in this paper. Previous authors in [1] and [2] suggest that modifications to the lane merging problem can be made by limiting the capacity of one or more lane(s), fixing the number of 1's in the arrival sequence, and by constructing color-blind equivalence classes of sequences. Each of these modifications may also be imposed on k-lane arrival sequences to yield further insight into the problem. Additionally, in [1] the authors mapped merging paths onto domino snakes, or longest paths in complete graphs with no loops. They then suggest that there may be an analogous mapping utilizing hypergraphs for a k-lane case. This remains an open problem.

Appendix

The findings in this paper are supported by a Maple package linked at https://aurorahiveley.github.io/merging.txt. Any bugs should be reported to the author at aurora.hiveley@rutgers.edu.

Acknowledgements

The author thanks her advisor Dr. Doron Zeilberger for the introduction to the problem and feedback on earlier drafts.

References

[1] V. Bardenova, E. Insko, K. Johnson, and S. Sullivan, A combinatorial model for lane merging, *Journal of Integer Sequences* **26** (2023) Article #23.5.3

[2] E. Insko, K. Johnson, and S. Sullivan, A combinatorial model for lane merging: Limited capacity lanes, *ECA* **6:2** (2026) Article #S2R9.

[3] N. Sloane and The OEIS Foundation Inc., *The on-line encyclopedia of integer sequences*, `https://oeis.org`, 2025.

[4] R. P. Stanley, *Enumerative combinatorics. Volume 2*, 2nd ed., Cambridge Studies in Advanced Mathematics, Volume 208, Cambridge University Press, Cambridge, 2013.

[5] D. Zeilberger, Enumerative and algebraic combinatorics, *The Princeton Companion to Mathematics*, Princeton University Press, pp. 550-561, 2008.

(A. Hiveley) Department of Mathematics, Rutgers University, Piscataway, NJ 08854
Email address: aurora.hiveley@rutgers.edu