CIRCULAR LOGIC

Contents

CONTROL NUMBER: 4095

1. Introduction

Traffic circles or roundabouts have been in use in Europe and other places around the world where they often replaced old public squares with the introduction of the car. Given the irregular nature of some of these old public spaces and and the streets that entered them, allowing motorist in the circle to proceed unimpeded while requiring motorist entering the circle to yield may have seemed like the only way to deal with the situation. Over time, roundabouts have been built as flexible, efficient intersections, often where the incoming road approach irregularly, or where their traffic volume are asymmetrical [6]. More recently, in the United States. traffic circles have been more commonly utilized as efficient replacement for controlled intersections, indeed their use has been growing exponentially there [5].

Despite their advantages, there are problems. Motorists in the United State are largely unfamiliar with them and either misuse them or are unnecessarily cautious. The most recent large Transportation Research Board study of traffic circles [5] found evidence that clear lane markings and inclusion of additional entry lanes improved traffic circle safety and performance. Moreover, they found that in multi-lane traffic circle those with the highest crash frequencies were those with lane overlaps in either entry or exiting. Given this, one must be fairly sophisticated

Date: 02/09/09 (In Theory).

about what one means by controlling 'right of way' when it comes to traffic circles; that is, simply postulating that entering vehicles must yield to those already in the circle (or visa versa), is not very helpful since what matters is the arrangements of entry and exit lanes and whether or not they cross. This paper develops a means for evaluating the efficiency of a traffic circle taking into account these issue of lane arrangement.

Our approach is straight forwardly evaluates the full range of possible *sane* geometries all with sensible and relatively safe geometries with few or no lane crossings for efficiency as determined by minimum average time spent in the traffic circle.

2. Assumptions

We will assume that the following data is given to us by the engineer.

- A function indicating the volume of vehicles per step time, Δt, for each time step of the day.
- The critical headway, $t_{critical}$, which is the measured timed between cars needed for a typical driver to merge into crossing traffic (Usually about 5 seconds–varies by region).
- The follow-up headway, $t_{followup}$, which is the additional time that a car in the queue, following the merging car, needs to merge (2 -3 seconds typically).
- The maximum density, k_{max} of vehicles at which traffic comes nearly to a halt (1.5 - 2.5 car lengths).
- The geometry of the traffic circle, the geometry of the roads entering the traffic circle, as well as the geometry of the roads exiting the traffic circle. This includes the average distance from the center midlines of the circle's lane, the lengths of various sections of the circle. Note that the engineer need not specify the number of lane associated with each incoming and out going roadway, the algorithm will determine the ideal arrangement.
- The volume $V_i(t)$ at input i per time step , Δt, for each time step.
- The probability g_{ij} of a driver going from entrance i to exit j.

From this information, we will compute the average delay, in seconds per car, of a car traveling through the roundabout. The traffic control method that produces the smallest average delay time we will declare to be the optimal traffic control method.

3. Axiomatic Traffic Theory

3.1. **Introduction.** Our first step in choosing the optimal control scheme for a traffic circle of given dimensions is to divorce superfluous physical properties of the circle from what, in essence, determines how traffic flows through it. That is, we wish to focus on how the underlying structure of the circle facilitates traffic

movement, and what choices we have in altering this structure when we alter the circle's control parameters.

Our first reduction amounts to regarding the traffic circle as a directed network. Both the utility and reasonableness of this reduction are illustrated throughout the remainder of the paper. The edges of the network's graph 'represent', in a sense made precise in §4, the lanes of the roads which pass through the circle. The nodes of the graph serve primarily to aid us in 'positioning' the edges in a manner which corresponds to how lanes are positioned on a road. On the network, edge-weights (some aspects of which are time-dependent, others not) represent localized quantities such as traffic density, the physical 'length' of a given edge, and other values which play a substantial role [3] in determining how traffic flows. (Of particular interest is the effect these quantities have on merging, the centrality of which is discussed in the traffic simulation section below.) While representing the roundabout in this way does cost us some geometric information about the circle, studies show that these finer geometric properties play little role in determining traffic flow [5].

Underlying every network is a graph, and it is this object which we consider in the present section. One might be given to believe that an unmanageably large number of graphs may correspond to possible lane coordinations in a traffic circle of even somewhat diminutive size. If one places no restriction on how the edges of the graph 'interact' with each other, there is some truth in this statement. However, in a real traffic circle, one does not allow arbitrary lane crossings, mergings, and enterings and exitings of the circle; there are (for reasons of both safety and sanity) restrictions on what can occur.

We now put forth a exemplary set of so-called 'axioms' which govern how lanes near a roundabout (i.e., edges on our network) may interact. We model these on observed lane behavior in actual roundabouts. The purpose of this process is to give rise to an algorithm which will compute *all* (and only those) edge configurations which satisfy some minimal 'sanity' requirements. The purpose of the axioms is to make this set of graphs reasonably (that is, realistically, but *not* artificially) small, so that, given a traffic simulation program (see §4) and basic input data (such as vehicles observed entering the vicinity of the roundabout from a given direction, and leaving in another direction as a rough function of the time of day) which determine edge weights, one may have a computer calculate which graph (i.e., control paradigm) optimizes traffic flow in a given situation, by computing a quantity (such as the average time a car spends in the 'traffic circle system') for each graph, and then selecting the graph which optimizes this quantity.

In following this program, we will see that what actually determines the essential aspects of a roundabout is the edge behavior we allow near entries to the circle and exits to the circle. Thus, the axioms we establish below are really as they were claimed above: exemplary. A differing set of axioms may be taken, and in order to apply the approach described here, one would merely need to recompute

the allowable structures at entries and exits, and feed this information along with the axioms to the graph-computing algorithm. (If one were to throw out even the most basic assumptions, one might need to consider how the traffic circle changed between entrances and exits; many of the resulting graphs would likely be pathological, but the approach suggested here would work to a good extent, if one considered nodes between entrances and exits in addition to the entrances and exits themselves.)

3.2. **Axioms.** Now, we set out our choice of axioms, and derive from them *all* allowable entry/exit structures. We then define an algorithm which, given this data, computes all graphs satisfying the axioms, and show by means of examples that this number is not unreasonably large, even for substantial numbers of inputs/outputs. For concreteness, and ease of producing pictures, we consider primarily the case when there are at most 2 lanes within the traffic circle at any given point; we observe that this, just as with our particular choice of axioms, is not a terrible hindrance to our method as even for considerably larger numbers of lanes, the essential information is contained in what the entry/exit structures of the traffic circle are, and how whatever choice of axioms we may make restricts how these structures can interact.

Our first axiom is the following:

1. **Traffic circles are circular**.

That is to say, the lanes orbiting the middle of the roundabout, as a whole, go 'all the way around'. (We do not assume that one particular lane goes all the way around.) This is really no restriction; if there were some case in which a non-circular traffic circle was desired, and was sufficient to connect all entries to all exits, one would hardly lose any performance value by allowing connections where, strictly speaking, there need be none. Lifting this axiom is conceivable, and would simply result in a larger number of entry/exit structures, but since the behavior prohibited by this assumption seems somewhat pathological, we choose to keep the axiom and restrict the possible structures.

Next, we have:

2. **Every entrance should have access to every exit**.

This is simply a statement of the fact that a traffic circle is seemingly most useful when one can come from any of its entrances and leave by any one of its exits. It is hard to imagine that a traffic circle which only allows traffic from a proper subset of its entrances to leave in a given direction would be more efficient than one which allows traffic from any entrance to leave by any exit, as a traffic circle not satisfying this axiom may require a given vehicle to enter and exit several times to get where it wants to go.

3. **Traffic in the circle does not yield**.

We wish to exclude this situation as the consequences of allowing yielding could be quite negative. Indeed, aside from the probable increase in in-circle accidents, allowing yielding within the circle would almost surely slow traffic flow through the circle. Practically speaking, this axiom restricts the possible lane arrangements, so that, for example, one lane of the roundabout may not merge into another lane which may contain traffic already in the circle.

4. **Lanes within the circle end only at exits**.

This is, to some extent, a corollary of axiom 3. If lanes were ended at points within the circle itself, this would necessarily lead to an issue of who is to proceed into the restricted set of lanes first, thus requiring yielding on the part of some (or all) cars involved. Moreover, it makes little sense to cut off a lane at any point except where traffic flow is leaving the circle; if a portion of the circle has two lanes, and these two and forced into one lane prior to an exit, then since the flow through this one lane must be at least as large as the flow through the previous two lanes (as there is no exit in between), either (i) the single lane will be 'congested', if two lanes are required to carry the full flow of traffic there, or (ii) the single lane will not be congested, but one of the two previous lanes will be superfluous (as a single lane suffices) and hence may as well have been cut off at a previous exit.

5. **Lanes within the circle are created only at entrances**.

For the most part, this too follows from 3, for if a lane is created away from an entrance and access is given to more than one existing lane, there is an issue as to who is licensed to enter this lane when. Moreover, as entrances are the only points where additional traffic flow enters the circle, it seems somewhat useless to introduce additional lanes away from an entrance; if traffic was flowing well with 1 lane at a given point on the circle, and we choose to split this into two lanes before any additional flow has entered the stream of traffic, what good will the additional lane do? If instead an additional lane is needed, it may as well have been created at the previous entrance, which is where the increased flow enters the circle.

6. **Distinct lanes approaching an entrance to the circle necessarily enter distinct lanes of the circle immediately after the entrance**.

Roughly speaking, we do not permit situations which cause traffic entering the circle to have to yield to other traffic entering the circle at the same entrance. We do, however, allow a single lane approaching the circle to enter multiple lanes within the circle (provided no other approaching lane may enter any of these lanes).

7. **Every lane approaching an entrance to the circle may enter at least one lane of the circle**.

This seems to be a reasonable way to avoid the possibility of 'useless' lanes.

8. **At an entrance, every lane in the circle immediately after the entrance is accessible to at least one of the entering lanes**.

This is not much of a restriction; in the event one lane of the circle is highly congested with thru-traffic, it seems likely that entering motorists will simply opt for a less congested lane.

9. **At an exit, lanes within the circle which have access to the exit are joined to at most one lane leaving the circle**.

We assume this because it simplifies the structure of exits, and does not materially effect how traffic flows. Indeed, if one lane within the circle were allowed to exit to multiple lanes, the additional bandwidth would be useless, as at most 'one lane's worth' of cars can be directed to the exit by the lane within the circle.

3.3. **Derivation of Entrances and Exits.** Now, from these axioms, we will derive all acceptable entrance and exit structures for a roundabout with at most two lanes. (The ideas presented below in computing entrances and exits apply equally well when more than two lanes are allowed, there are simply more cases to work out). It is important to note that not all axioms will be used in these derivations; some axioms govern how entrances and exits interact, while others govern how they may be formed. The former group of axioms will be applied in the algorithm described below, when we compute, given minimal data, all graphs of a 'given type' satisfying all the axioms.

Consider the general structure of an exit.

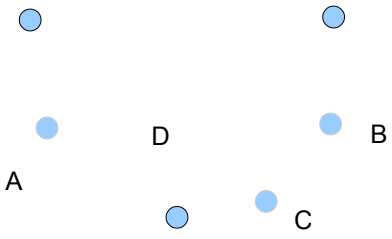

FIGURE 1. Exit.

Here, the top two solid nodes represent points on the lane which we know exists at the exit. The top two lighter circles represent possible points on an outer lane which may or may not exist. The solid node on the bottom represents the exiting

lane we know must exist, and the lighter node at the bottom represents a second exit lane which may or may not exist. We are licensed to assume that there is at least one lane both before and after the exit by axiom 1, and we may assume there is at least one lane leaving the circle, as this is what we mean by 'exit'.

We take the symbols A, B, C, D to be either true in false, where A is true iff there is a second lane in the circle approaching the exit, B is true iff there is a second lane in the circle departing from the exit, C is true iff there is a second lane at the exit leaving the circle (assuming that all lanes exiting the circle are accessible to a lane within the circle, as we lose no generality by ignoring the lanes which exit the circle but are, in essence, useless), and D is true iff there is a second lane in the circle which ends at this exit. (In fact, fewer variables will suffice, but considerations seem more intuitive with four.)

Suppose that B (that is, suppose that B is true). Then A (that is, A is true), for we do not create lanes at exits (axiom 5). Moreover, not D (that is, D is false) for if the second lane ends at the exit, it will not be present immediately afterwards (which contradicts the truth of B). Moreover, not C, as the outer lane has access to at most one exiting lane (axiom 9), and the inner lane cannot exit here, as it would necessarily cross the outer lane in doing so (creating a yielding situation, in contradiction of axiom 3). The exit structure described by this interpretation of the symbols A, B, C, and D is shown below.

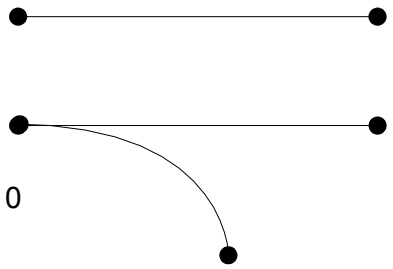

FIGURE 2

If not B, it turns out that there are three different cases (resulting in a total of four different exit structures). If C, then we necessarily have A and not D. For, the existence of a second exiting lane implies that both the inner and outer lanes of the roundabout have access to an exiting lane, as a single lane of the roundabout uses at most one exiting lane (axiom 9). Thus, both the inner and outer lanes may exit here, and so the outer lane must surely exist (whence A), and may not persist

after the exit, as we have not B (whence D). The sketch of this configuration is shown below.

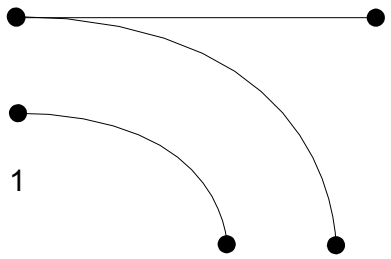

FIGURE 3

If not B and not C, then we clearly have A iff D, and so there are two more cases here (again, shown below).

By inspecting the graphs above, these four situations are unique, and as we have considered all interpretations of A,B,C, and D which are consistent with our axioms, these are the only four exit structures. (There is nothing more to consider about how various lanes of traffic may proceed, due primarily to axiom 3.)

Now, we consider the situation of entrances.

As with exits, the top solid nodes represent points on a necessarily existent lane in the roundabout. The top two lighter nodes represent points on a possible second lane. The bottom solid node represents an entering lane which must exist (by the definition of 'entrance'), and the bottom lighter node represents a possible second entering lane.

The situation here is slightly more complicated than that of exits. Here A is true iff there is a second lane in the circle approaching the entrance, B is true iff there is a second lane in the circle departing the entrance, C is true iff there is a second lane of traffic approaching the entrance to the circle, D is true iff the inner lane is allowed to proceed in the inner lane past the entrance, and E is true iff the inner lane is allowed to proceed to the outer lane past the entrance.

Suppose not B. We then have not A, as lanes are ended only at exits. Moreover, we have not C, for if C were true, we would have two oncoming lanes entering the single lane after the entrance (in contradiction of axiom 6). We also have not E, as there is no outer lane for traffic in the inner lane to move into. Therefore, we have D, as otherwise our traffic circle would not be 'circular', in the sense of axiom 1. This situation is depicted below.

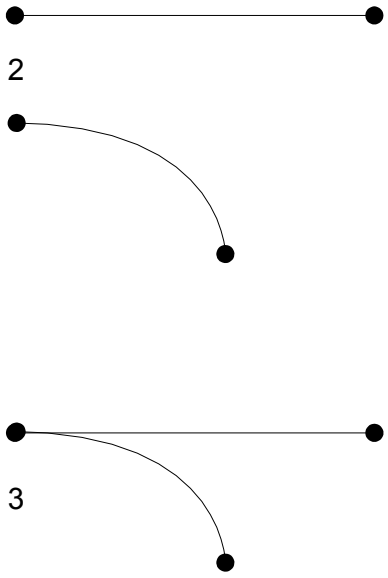

FIGURE 4

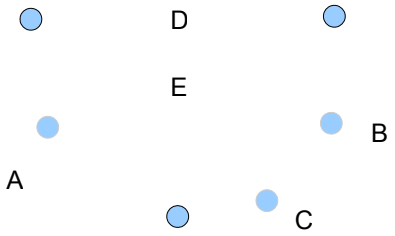

FIGURE 5. Entry.

If instead B, suppose A. Then not E by axiom 3, whence D by axiom 4. It is conceivable that either C or not C. See the figures below.

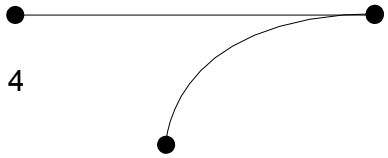

FIGURE 6

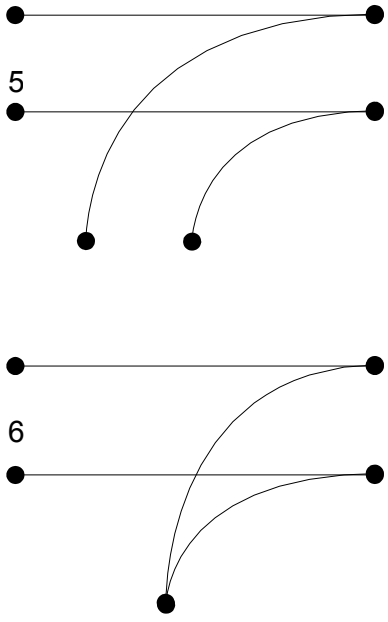

FIGURE 7

If B and not A, suppose not E. Then D. Again, either C or not C is consistent.

If B, not A, and E, then D and C are arbitrary. Thus, there are four more cases.

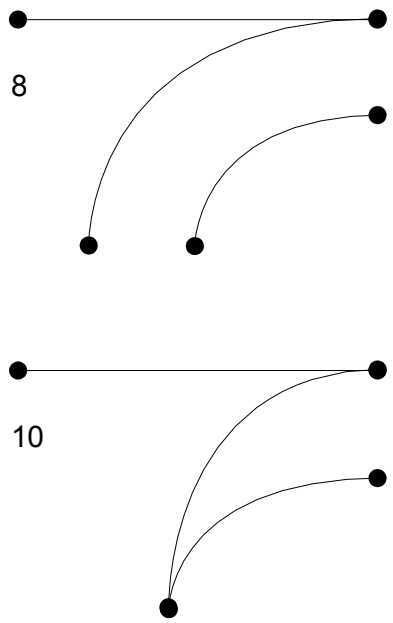

FIGURE 8

Thus, we have considered all cases, and there are 9 distinct entrances. In the following section, we will refer to the structures by their numbers, indicated in the above figures.

3.4. **Algorithm.** Limiting the possibilities as we have, we note that, given the relative order of inputs and outputs of a roundabout (starting at a given input/output and proceeding, for concreteness, counterclockwise when viewed from above), we can determine all graphs which satisfy our axioms and have entrances and exits in the sequence given. That is, if we are given as input a list $[i, o, o, i, i, i, o, o, ...]$ of the order in which 'ins' (i.e., i's) and 'outs' (i.e., o's) occur in a roundabout, we may determine every possible control scheme within the circle which satisfies the axioms above and has entrances and exits in the proper order. The outline of the algorithm follows, and as examples we compute the possibilities for various inputs. The idea is that each of these possibilities could be run through a traffic simulator given basic information about the dimensions of the location for the traffic circle and how many cars are entering the roundabout area from each possible direction and how many cars are leaving in each possible direction, and the graph (or control scheme) producing the minimal average wait time for drivers (for the given data) could then be selected as a candidate for installation. Although the number

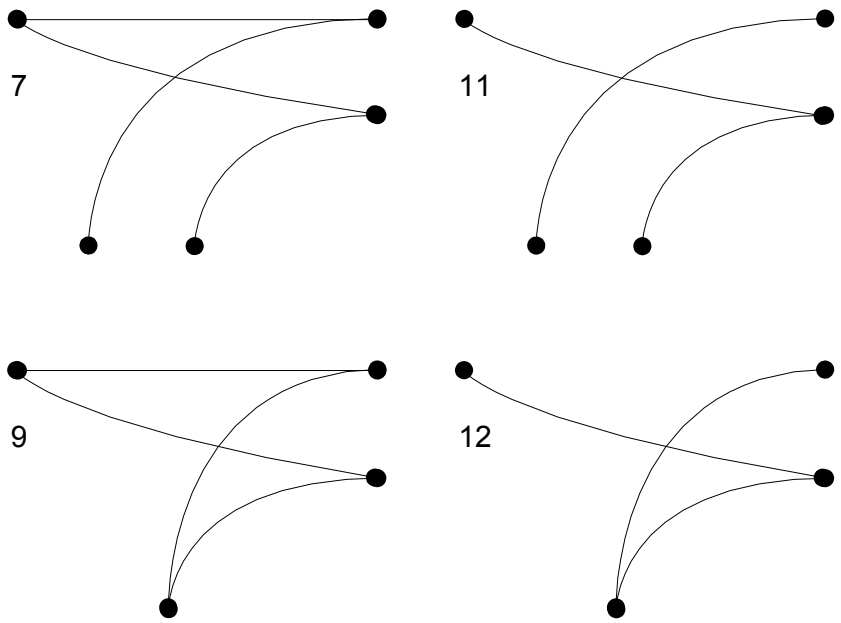

FIGURE 9

of possible combinations of entries and exits under no hypotheses whatsoever is $4^{n_o}9^{n_i}$, where n_o is the number of 'outs' and n_i is the number of 'ins', the number of axiomatically reasonable possibilities is *much* smaller, as we will see below.

We illustrate the algorithm with an example. Let $Z = [i, o, i, o, o, i, i, o]$. We wish to compute all reasonable control schemes for this type of roundabout.

We begin with the first index of Z, that is, an i. Let $X_1 = [[4], [5], [6], [7], [8], [9], [10], [11], [12]]$ be the list consisting of nine lists, each consisting of one of the nine possible entrances. The idea is to iteratively build up complete lists (that is, of length equal to the length of Z) in a way to avoid lists we know to be inconsistent prior to completing them. We then check that the last element of each list is consistent with the first element; if this is the case, then the list represents a 'consistent' control scheme (although there is more checking to be done, as discussed below). Otherwise, it does not, and we discard it.

Thus, we wish to build up the lists in X_1. Notice that the entrance 4 has only one lane in the circle to the right of the entrance; thus, the next structure must have only one lane on the left. As the next structure must be an exit (as the value at the second index of Z is an o), the only possible structures we may append to [4] to obtain a conceivably consistent 'partial' list are those exits with one 'incoming

lane', i.e., the exit 3. Similarly, each of the entrances 5-12 has two 'outgoing' lanes, and so the next structure in each case must have two 'incoming' lanes. Therefore, to these lists we may append any of the following exit numbers: 0,1,2. Thus, we have a new set $X_2 = [[4,3],[5,0],[5,1],[5,2],[6,0],[6,1],[6,2],...,[12,0],[12,1],[12,2]]$ of larger partial lists which have some hope of being consistent.

The idea applied before generalizes immediately to the case where X_j consists of lists of length j, where j is less than the length of Z. We first observe how many outgoing lanes the structure at the end of a given list in X_j has, and then construct a set of new lists by appending to the given list each of the different possible structures (which depend on the number of outgoing lanes and whether the element at the j^{th} index of Z (assuming 0-based indexing) is an i or an o).

Once the above process has been iterated until the lists under construction have length equal to the length of Z, there are two more steps we must take to obtain precisely those lists corresponding to control schemes which are consistent. First, if the collection of these lists is called X, we must check each element of X to determine if the last entry in the given element can be consistently followed by the first entry in that same element. If the elements may follow each other in that way, then we place this element of X into a new set, say, Y, otherwise we discard it.

Finally, there is a question of guaranteeing axiom 2. To this end, we note that each entrance and exit (with index j) determines a map f_j defined on the set $\{a,b\}$, where a represents the 'inner lane' of the circle (or the only lane of the circle at those locations where there is only one lane) and b represents the 'outer lane' (where it exists) such that $f_j(a)$ is the set of lanes of the roundabout (i.e., a subset of $\{a,b\}$) which traffic entering the structure j from the inner lane may leave the structure j in, and $f_j(b)$ is the set of lanes of the roundabout which traffic entering the structure from the outer lane may leave the structure in (where this set is empty if the second lane does not enter the structure j). For example, the map defined by the structure 0 is $f_0(a) = \{a\}$ and $f_0(b) = \{b\}$, while the map defined by structure 11 is $f_{11}(a) = \{b\}$ and $f_{11}(b)$ is empty. Moreover, each entrance determines a set of lanes which traffic entering at that entrance has access to. For example, the set corresponding to entrance 4 is $\{a\}$, and the set corresponding to entrance 5 is $\{a,b\}$. Of course, each exit also determines a set of lanes which has access to that exit. For example, the set corresponding to exit 0 is $\{b\}$, while the set corresponding to exit 1 in $\{a,b\}$.

To determine if every entrance has access to every exit for a given control scheme y in Y, we proceed as follows. If j is in index such that the j^{th} element of Z is i, let S_0 be the set of lanes which traffic entering at the j^{th} element of y has access to. If j_1 is the least non-negative congruence of $j+1$ modulo the length of Z, we then apply the map corresponding to the j_1^{th} element of y to S_0 to obtain the set S_1; this is the set of lanes which traffic entering at the entrance corresponding to the j^{th} element of y have access to after passing the structure corresponding to

the j_1^{th} element of y. If the j_1^{th} element of Z is o, we also apply to map determined by the exit structure at the j_1^{th} index of y to the set S_0, to obtain a set E_1 (if the j_1^{th} element of Z is not an o, define E_1 to be any nonempty set). Two things can go somewhat awry here. First, the set E_1 could be empty. If it is, then we know that there is an exit which traffic entering at the entrance corresponding to the j^{th} element of y cannot reach, so that axiom 2 is contradicted. Secondly, S_1 could be empty. If this is the case, we cannot immediately conclude that there is an exit which traffic flowing in at the j^{th} element of y cannot reach; in fact, such an exit will exist iff there is an o at the j_k^{th} index of Z (where j_k is the least non-negative congruence of $j+k$ modulo the length of Z) for some integer k greater than 1 and less than the length of Z. So, in the case that S_1 is empty, we check to see if the j_k^{th} element of Z is every o as k ranges over $\{2, 3, ..., |Z|\}$, where $|Z|$ is the length of Z. If it ever is, axiom 2 is not satisfied. If it is not, then traffic entering at the entrance j can indeed reach any exit in the roundabout.

Should neither issue above arise, we proceed in the manner which the above suggests, applying the map corresponding to the j_2^{th} element of y to S_1 to obtain S_2, and obtaining E_2 as above, and more generally applying the j_k^{th} element of y to S_{k-1} to obtain S_k, and obtaining E_k as above, until either (i) the set E_k is empty, or (ii) the set S_k is empty, or (iii) k equals the length of Z. In case (iii), we may immediately conclude that all traffic entering at the entrance corresponding to the j^{th} element of Z may access every exit. In case (i), we may immediately conclude that the traffic entering at j cannot reach some exit. In case (ii) we check the j_l^{th} element of Z (as l ranges over $\{k+1, k+2, ...|Z|\}$) to see if it is an o; traffic coming in at j cannot reach some exit iff the j_l^{th} element of Z (for some l as above) is o.

We repeat this process for every index of Z at which an i occurs. If for each such index, we determine that traffic entering at that index in the control scheme defined by y may access every exit, then we add y to some other set, say P. Otherwise, we move on to the next element of Y.

Every element of this set P now satisfies all the axioms, and hence defines a control scheme worth performing a simulation with.

Translating the above outline into computer code, we obtain an algorithm which will compute all the elements of P. With Z as above, we find that, before carrying out the last step of the algorithm above, there are 6825 elements (these would be elements of Y). The number of elements of P is then determined to be 2903. For example, one element of P is $[11, 0, 6, 0, 2, 7, 5, 2]$, and an element of Y which is not in P is $[7, 1, 10, 0, 2, 4, 7, 2]$. One sees immediately by drawing out these figures that $[7, 1, 10, 0, 2, 4, 7, 2]$ prohibits, for example, cars entering at the first instance of 7 from leaving at the exit 0, while $[11, 0, 6, 0, 2, 7, 5, 2]$ does indeed allow all possible combinations of entry and exit.

4. Traffic Modeling in a Traffic Circle

In most implementations, traffic circles are governed by the rule that entering traffic yields to circulating traffic, as such, they are a simple low maintenance way to deal with complex intersections. This simple rule, though it endows traffic circles with their flexibility, may cause significant delays for entering vehicles during congested periods. Thus, in modeling them we have chosen to focus on two essential issues: the delay of incoming traffic when it looks for an opportunity to enter the circle and circumstances that cause congestion in the circle.

The yielding issue is commonly modeled using the critical headway, $t_{critical}$, which is the spacing in time between successive cars traveling in the circle that is just large enough to encourage an average driver to enter the circle sliding in between them, and the follow up headway, $t_{followup}$, which is the additional spacing in time between successive cars just large enough to encourage an average driver to sneak in behind the first [8]. These values are empirical measurements that vary from region to region. We expect the engineer will provide local values for these. The issue of congestion we address using a car following model, found in [3] which looks at expected responses of driver to varying spaces between vehicles, to deduce a relation of traffic throughput to density. One instantiation of this relation that modeled traffic in Chicago well is

$$q = k\alpha^5 \left((k_{max})^{1.8} - k^{1.8}\right)^5$$

where q is the vehicles past a point per unit time, k is the traffic density, and k_{max} is the traffic density at which everything comes to a halt (1.5-2.5 car lengths) [3]. This result readily implies that the speed

$$v = \alpha^5 \left((k_{max})^{1.8} - k^{1.8}\right)^5,$$

where

$$\alpha = \frac{(v_{free})^{1/5}}{(k_{max})^{9/5}},$$

and v_{free} is the velocity of a vehicle unconstrained by other traffic. Using these two ideas, we attempt to model the average driver time through the traffic circle during the course of a day as a measure of the efficiency of the traffic circle design, and, thus, compare the effectiveness of various lane configurations by our circle generator described above.

We model the geometry of the circle using a directed graph. This is a reasonable approach since previous studies have noted that the finer detail of circle geometry is not significant in how well it works[3]. Moreover, graph vertices model roadway merging, and edges model roadway lengths well while simplifying things substantially.

As noted above, in addition to the circle geometry and relative position of circle ins and outs, we ask that the engineer provide, volume $V_i(t)$ at *in* i per time step, Δt , which we assume to be equal in size to the critical headway, $t_{critical}$, for each

time step and the probability g_{ij} of a driver going from *in* i to out j. Our model generates an average time per driver spent in the circle given the input volumes. We abort our model when a queue at any in exceeds one hundred vehicles.

We build our model on a generated graph by first assigning to each edge, e, a radius and length, L_e determined by the real world section of traffic circle roadway it models, and, using known empirical relationships between unconstrained vehicle velocity and radius of traffic circles [1] [5], we determine the free velocity on that edge. Then, we assign the following variables describing the traffic on each edge, e: n_{ej} the expected number of vehicles on that edge heading to circle out i; $\bar{t}_{ej}$ the expected, average time spent by the e: n_{ej} vehicles heading towards out i; and the expected, average velocity of the vehicles, $\overline{v}_e$ as calculated from the formula in the previous paragraph using the vehicle density in the edge.

We assume that the volumes $V_i(t)$ build from low 'midnight' level, and the initially empty traffic circle fills with vehicles gradually. The process of introducing vehicles to the circle is a bit roundabout. We look at each of the graph's connected component separately. If that component has an edge that feeds only into an out, we alway begin the process there. If not, we survey the edge velocities, $\overline{v}_e$ and the number of vehicles on each edge, $N_e = \sum_i n_{ei}\Delta t$. The expected number of vehicles that an edge will try to dump into the edge ahead of it is $N_e v_e$ — assuming velocity does not change and there are no outs nearby. Then, we begin at the edge where $N_e v_e$ is minimum in that connected component. This guarantees that there is 'room' for that many vehicles in the edge ahead of it, since the latter edge itself will be dumping at least that many vehicles into the edge it front of it. In either case, to calculate the traffic distribution after one additional time step, Δt, we proceed to update each edge one at a time going backwards against the flow of traffic. There are five kinds of edges when considered in relation to the edges they feed into or feed into them.

4.1. **Algorithm Description.** The following are algorithm snippets for updating each type of edge. Let e be the edge we are updating, f be the edge in front of it, $TimeStep$ is Δt, and $TotalTime$ be the total time spent by all vehicles in the circle.

Updating for an edge that feeds only into an out.
Let o be the out e dumps into.

```
# e is the edge
# o is the out
# n[e, o] is the expected average number of vehicles in e going to out o
# t[e, o] is the expected average time vehicles in e going to out o
#             have spent in the circle
#
#Dump Out vehicles going to out o
#
```

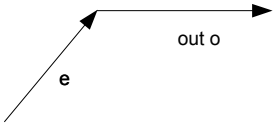

FIGURE 10. An edge that feeds only into an out.

```
Movement := min{ v[e]*TimeStep/L[e], 1}
TotalTime := TotalTime + Movement*(n[e, o]*t[e, o]),
n[e, o] := n[e,o] - Movement * n[e,o]
t[e, o] := t[e, o] + TimeStep
```

Note that since edge e only empties into out o, $n_{e,k} = 0$ for $k \neq 0$.

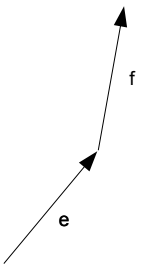

FIGURE 11. An edge that feeds only into one edge.

Updating for an edge that feeds only into one edge.

Let N_e and N_f be the number of vehicles on the edges e and f, k_{max} be the maximum density, and $\alpha = \frac{(v_{free})^{1/5}}{(k_{max})^{9/5}}$.

```
# alpha, Kmax, N[e], and N[f] are as above
# L[e] is the length of edge e
# v[f] is the average velocity on edge f
#
#Dump Forward vehicles into edge f
   #if all vehicles can't fit, scale the movement
   #
Movement := min{ v[e]*TimeStep/L[e], 1}
TotalAdd := Movement*N[e]
SpaceAvailable := (0.9)*L[f]/Kmax - N[f]
Proportion :=  min{ SpaceAvailable/TotalAdd , 1 }
for each out in  Circle:
     MovedForward :=  Proportion*Movement * n[e, out]
     t[e, out] := t[e, out] + TimeStep
     t[f, out] := ( t[e, out]*MovedForward + t[f, out]*n[f, out] )
                       / (MovedForward + n[f, out])
     n[f, out] := n[f, out] + MovedForward
     n[e, out] := n[e, out] - MovedForward

v[f, out] := alpha^5 *(Kmax^{1.8} - (N[f]/L[f])^{1.8} )^5
```

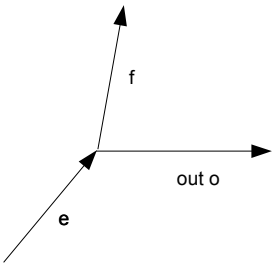

FIGURE 12. An edge that feeds into one edge and also feeds into an out.

Updating for an edge that feeds into one edge and also feeds into an out.

```
#Dump Forward all but those vehicles going to out o
  #if all vehicles can't fit, scale the movement
```

```
  #
Movement := min{ v[e]*TimeStep/L[e], 1}
TotalAdd := Movement*N[e]
SpaceAvailable := (0.9)*L[f]/Kmax - N[f]
Proportion :=  min{ SpaceAvailable/TotalAdd , 1 }
for each out in  Circle except for out o:
     MovedForward :=  Proportion*Movement * n[e, out]
     t[e, out] := t[e, out] + TimeStep
     t[f, out] := ( t[e, out]*MovedForward + t[f, out]*n[f, out] )
                        / (MovedForward + n[f, out])
     n[f, out] := n[f, out] + MovedForward
     n[e, out] := n[e, out] - MovedForward

v[f, out] := alpha^5 *(Kmax^{1.8} - (N[f]/L[f])^{1.8} )^5

#Dump Out Vehicles going to out o
Movement := min{ v[e]*TimeStep/L[e], 1}
TotalTime := TotalTime + Movement*(n[e, o]*t[e, o]),
n[e, o] := n[e,o] - Movement * n[e,o]
```

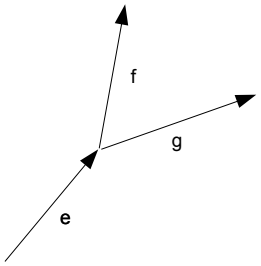

FIGURE 13. An edge that feeds into two edges.

Updating for an edge that feeds into two edges.

Typically, one edge, f, leads to some of the outs, say $\{1, 2, 3\}$ and the other, g, leads to $\{3, 4\}$. In this case, we build a probability table, $P_{g,o}$, to help determine which vehicles go where. The probability of going of a vehicle with destination out 1 going to edge f is 1, (and the probability of it going to g is zero). However, he probability of going of a vehicle with destination out 3 going to edge f is 0.5.

```
# P[g, o] is probability for a vehicle on the edge before g and destined
#              for out o of choosing edge g to move to
#
#Dump Forward vehicles into edge f or edge g depending
   #if all vehicles can't fit, scale the movement
   #
Movement := min{ v[e]*TimeStep/L[e], 1}

SpaceAvailablef := (0.9)*L[f]/Kmax - N[f]
TotalAddf := Sum over outs {Movement*n[e, out]*P[f, out] }
Proportion :=  min{ SpaceAvailable/TotalAddf , 1 }
for each out in  Circle:
     MovedForward :=  Proportion*Movement * n[e, out]
     t[e, out] := t[e, out] + TimeStep
     t[f, out] := ( t[e, out]*MovedForward + t[f, out]*n[f, out] )
                       / (MovedForward + n[f, out])
     n[f, out] := n[f, out] + MovedForward
     n[e, out] := n[e, out] - MovedForward

v[f, out] := alpha^5 *(Kmax^{1.8} - (N[f]/L[f])^{1.8} )^5

Repeat this as above for g
```

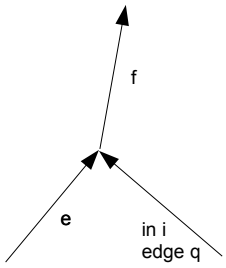

FIGURE 14. An edge that passes an *in*as it feeds into the next edge.

Updating for an edge that passes an *in* as it feeds into the next edge Note that we, for simplicity sake, present the algorithm addressing the situation where the *in* feeds only one lane. An in can, however, feed into both the inner

and outer ring of the circle. In that case, we add a probability table similar to the above case to help decide how many vehicles feed into each edge, and we insert an additional conditional to determine if the time spacing between cars in the inner ring is sufficient to allow merging.

```
# Edge q is the edge holding the queue for the in i
# TimeSpace is the time gap between circulating cars
# Tcritical is the Timespace needed to enter from a stop (=TimeStep)
# Tfollowup is the Timespace needed to enter behind another vehicle
# Flag  indicates if the in is in stopped or followup state (intially Stop)
# TimeUsed is the portion of TimeSpace filled by merging vehicles at present
# g[i, o] is the probability of a vehicle entering at in i, and leaving at out o
#
Dump Forward all vehicles going from edge e to f (see above)
TimeSpace := L[f]/ (v[f]*N[f])
If Flag = Stop then:
     If TimeSpace >= Tcritical then:
          Flag := Go
          # Merge one representative vehicle into edge f
          #
          for each out in  Circle:
               t[f, out] := (t[q, out]*n[q, i] + t[f, out]*n[f, out]*g[i, o])
                       / (n[q, i]  + n[f, out])
               t[q, out] := t[q, out] + TimeStep
               n[f, out] := n[f, out] + g[i, o]
               n[q, out] := n[q, out] - g[i, o]

          v[f, out] := alpha^5 *(Kmax^{1.8} - (N[f]/L[f])^{1.8} )^5
          TimeUsed := TimeStep

If Flag = Go then:
     TimeAvailable := L[f]/(v[f]*N[f]) - TimeUsed
     FollowupCars := min{ TimeAvailable, TimeStep }
     for each out in  Circle:
          t[f, out] := (t[q, out]*n[q, i] + t[f, out]*n[f, out]*g[i, o])
                       / (n[q, i]  + n[f, out])
          t[q, out] := t[q, out] + TimeStep
          n[f, out] := n[f, out] + g[i, o]
          n[q, out] := n[q, out] - g[i, o]

      TimeUsed := TimeUsed + TimeStep
      if TimeUsed >= L[f]/(v[f]*N[f]) then Flag := Stop
      if Flag = Stop then TimeUsed := 0
```

After updating each edge, we advance the time by one time step, δt, and add the appropriate number of vehicles to each entering queue based on the volume function the engineer provided. We, then, check to see if any of the queues are of excessive length, if so, we abort the simulation. If all is good, we begin updating the edges again.

This traffic simulation algorithm works in conjunction with the circle generating algorithm just as one might imagine. After the engineer provides the parameters of the traffic circle, the circle generating algorithm generates all possible lane arrangements for the circle, each of which is fed into the traffic simulation for one days worth of volume. Admittedly, some circle geometries can generate 1000 to 15000 different possibilities combined with 10000 time steps each yield up to 10^8 iterations . This is an upper bound, even so, it could be done in 10^6 seconds, 12 days, using SAGE on a public server, making this very lengthy, but practical algorithm for serious design problems.

5. Technical Summary

Traffic Control Algorithm. This algorithm minimizes the average time of traveling through a multi-lane traffic circle. Given appropriate input, this algorithm will indicate how to restrict entrances, exits and paths through the network to minimize travel time.

Input

- The critical headway, which is the amount of time gap needed between cars for a average driver to merge into traffic.
- The follow-up headway, which is the additional time that a car in the queue, following the merging car needs to merge.
- The distance from the center of the circle to the outside of the island, the lengths of the lane, as well as the position of lanes entering and exiting the circle.
- The rate of cars flowing into the circle at entrance roads, at every given time.
- For every pair of entrances into, and exits out of, the circle, the percentage of drivers entering that entrance that leave at that exit.

Output

From this data, we simulate the traffic in a given network at any given time keeping track of total wait time of all cars through the network, and computing the average wait time when the program terminates. We do this for all logical choices of restrictions on entrances, exits and paths throughout the network, and output the choice of restrictions which produces the smallest average time of traveling through a multi-lane traffic circle. This is done by

- Simulating traffic movement that occurs in a change of time equal to the critical headway. We consider the circle as being made up of discrete pieces, and consider the number of cars in each piece of the traffic circle. Using known formulas for velocity and traffic flow in terms of capacity and density, as well as the number of cars going from a specific entrance to a specific exit, and the rate of cars flowing into the entrance lanes, we calculate how many cars will enter the traffic circle, how many will exit, and how many will merge into specific lanes.
- When traffic moves to another lane, enters the circle, or exits the circle, we then update the velocity and average wait time at each piece of the traffic circle, as well as update the total wait time of all cars through the network. We also update the number of cars in each piece of the traffic circle, as well as the number of cars waiting to enter and exit the traffic circle.
- We increment the time forward an amount equal to the critical headway. We then update the critical headway, if necessary, accounting for how many cars are in the traffic circle.
- We repeat the above steps until an entire day is simulated.
- Having done this for all logic choices of traffic connections, we output the choice of connections and lane arrangements that give the minimal wait time through the network.

References

[1] Rahmi Akcelik, *Estimating negotiation radius, distance and speed for vehicles using roundabouts*, Akcelik & Associates Pty Ltd, 2004

[2] Sandra Almeida, Luis M. Correia, Jose Queijo, *Spatial and Temporal Traffic Distribution Models for GSM*

[3] Wilhelm Leutzbach, *Introduction to the Theory of Traffic Follow*, Springer-Verlag, 1988

[4] Metro Count Traffic Data Analysis, *Gap Analysis*, www.metrocount.com/downloads/fiels/flyers/Gap analysis.pdf

[5] *National Cooperative Highway Research Program, NCHRP Report 572: Roundabouts in the United States,* http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rpt_572.pdf, 2006

[6] *National Cooperative Highway Research Program, NCHRP Report 264: Modern Roundabout Practice in the United States,* http://onlinepubs.trb.org/Onlinepubs/nchrp/nchrp_syn_264.pdf, 1998.

[7] *Ourston Roundabout Engineering*, http://www.ourston.com/04a_Interchanges.htm

[8] Zong Z. Tian, and Feng Xu, *Driver Behavior and Gap-Characterstics at Roundabouts In California*, http://trb.metapress.com/content/02135v08071106066/fulltext.pdf, 2008

[9] U.S. Department of Transportation, *Roundabouts: An Informational Guide*, Publication No. FHWA-RD-00-67, 2000