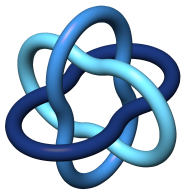
Summary

On your mark, get set, go! Triathlons are globally held competitions that put the fittest of the fit to the test. A local town wants to hold a triathlon and is expecting 2000 athletes – both amateurs and professionals – to participate. The sponsor of the triathlon, Super Tread Racing Company, would like to reduce congestion on the course, and the Town Mayor does not want the public roads to be closed (during the running and cycling segments) for more than 5.5 hours. In order fulfill both requests, we created divisions that will be used in the race as well as a schedule of wave start times. Furthermore, to determine the advantages that might be achieved by varying the distances of each segment of the triathlon, we developed an agent-based model capable of organizing each segment.

Our agent-based model was created in NetLogo, using data taken from a previous triathlon to determine the apportionment of athlete sex and segment times. We placed the athletes into eight divisions (four per each sex), where each division represents a respective quartile of the course completion times. Congestion was defined as being linearly related to the number of people athletes had to pass and logarithmically related to the number of people around each athlete. We developed the course of the triathlon, in which we demarcated the path into three lanes – fast, main, and slow – where each lane is based on segment times for each athlete. The three-lane system decreases congestion by 72% relative to a base model with no lanes. Separately closing the cycling roads and the running roads allowed us to minimize congestion by 29%. We then used our agent-based model to generate a schedule of wave start times that will fit the criteria set by the CEO and the mayor. We constructed "wave-packets" to further minimize the congestion and road closure time. Each wave-packet contains 55 athletes with a distribution of: 25% fast athletes, 25% slow athletes, and 50% medium-speed athletes. By releasing the wave-packets in order of decreasing average speed, we ensure that wave-packets do not interfere with each other, decreasing congestion by 27%. By testing our model with various values for the separation time between wave-packets, we found that separating wave-packets by 4 minutes would minimize congestion while keeping the road closure time under 5.5 hours.

Using our model, we determined that the 2000 athletes should be placed into groups of 55, with 14 fast athletes, 27 medium athletes and 14 slow athletes in each group, and the groups should be deployed every four minutes. Our model is in line with the Olympic triathlon standards for segment lengths and satisfies the CEO's desire to attract high-skilled racers with professional standards. The measures we take to decrease congestion (lanes, wave-packets, and bike/run road splitting) all significantly decrease congestion. By changing segment distances, we saw that decreasing the running length from 10 km to 5 km would significantly decrease congestion by 22% but would compromise our adherence to Olympic standards. Furthermore, our solution does not exceed the 5.5 hour limit for road closure time. Our numerous, skill-based divisions allow for a fair yet competitive environment.



MathℒetΣs
Mathematical Modeling Team **John Smith, PhD**

John Smith
MathℒetΣs
123 Broadway
Berkeley CA 12345
Phone: (555) 555-5555
E-mail: johns@mathletes.com
URL: http://www.mathletes.com

November 14, 2016

Mayor Tali North
City Council of Bradtown
Bradtown, California 12823

Dear Mayor North:

Thank you for working with us to design a schedule for our town's triathlon! We are truly very excited to help our town and community with this project. As we discussed in prior meetings, the City Council shared your concerns about the traffic and complicated logistics of closing roads for an extended period of time. Furthermore, the Super Tread Race Company, the main sponsor of the event, wanted to reduce the congestion during the race to cater to professional and premier athletes.

Considering the complexity of scheduling waves for the triathlon and catering to both amateurs and professionals, we devised agent-based models that suggest that the athletes should be divided up into 8 divisions - first by gender and then by skill. The divisions are: Female Black Diamond, Male Black Diamond, Female Diamond, Male Diamond, Female Gold, Male Gold, Female Silver, and Male Silver. After analyzing past triathlon data, we were able to divide athletes into four quartiles, from which we approximated the cutoffs for each division. For the all-male divisions, the record time cutoffs are 2.50, 3.00, and 3.50 hours. For the all-female divisions, cutoffs will be at 2.75, 3.25, and 3.75 hours. To assess the past record of the athletes that attend the triathlon, we will conduct a survey during registration. Although we considered making divisions along the categories of the athletes (such as professional, premier, and open), we decided to allow experienced and amateur athletes to compete against athletes of similar skill levels. By evening the playing field, everyone has at least some chance to receive commendation. Furthermore, a balanced, competitive atmosphere between the professionals and premier athletes will increase cooperation and future triathlon ventures.

Given your request that the roads cannot be closed for more than 5.5 hours, we must send a large amount of people into the triathlon at a time. This could potentially cause large amounts of congestion within the triathlon. In order to properly address this, we decided to model the congestion. We found that splitting the course into a multi-lane system with a fast-lane and a slow-lane reduced congestion by 72%. We can reduce congestion further by 29% if we individually close the biking segment and the

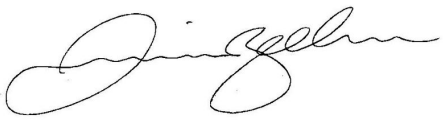
running segment. Of course, for this to work, we would need the running and biking segments to be on two separate road systems. Thus, we would be much obliged if you reserved space for our triathlon such that these two adjustments are possible. It would also benefit us if you could reserve either long or wide stretches of roads, so we can minimize congestion due to lapping.

Logistically, our triathlon sends out "wave-packets" of 55 individuals at a time. These wave-packets are sent out in order of decreasing average speed. This special group ordering was shown to decrease the congestion by 27%.

Using our model, we have also come up with a schedule for the triathlon program. Since the optimal difference between two wave-packets is 4 minutes, we have adjusted our schedule accordingly. Under such conditions, the triathlon will last for a maximum of 7 hours. Furthermore, we have scheduled fundraiser events after the triathlon for the benefit of the community!

Start Time	End Time	Event
7:00 AM	7:30 AM	Setup - Youth volunteers help set up stations
7:30 AM	8:00 AM	Registration - Athletes register and receive wave numbers
8:00 AM	-	First Wave-Packet - First 55 Athletes Depart
8:04 AM	-	Second Wave-Packet - 55 Athletes Depart
8:08 AM	-	Third Wave-Packet - 55 Athletes Depart
⋮	⋮	⋮
10:24 AM	-	Thirty-Seventh Wave-Packet - Final 20 Athletes Depart
8:00 AM	3:00 PM	Triathlon
3:00 PM	3:30 PM	Lunch for Participants
3:30 PM	4:00 PM	Keynote Speaker from Sponsor for Fundraiser
4:00 PM	5:00 PM	Meet-and-Greet with Professionals

Sincerely,



John Smith, PhD

Introduction

The Super Tread Race Company is sponsoring a triathlon in a local town, and they want us to make it a world-class event. A triathlon is a multiple-stage athletic endurance competition of three continuous, sequential segments of a 1500 m swim, a 40 km bike ride and a 10 km run [1]. Participants compete with each other to achieve the lowest course completion time, which includes the time it takes to transition between segments [1]. In anticipation of the triathlon, the town Mayor wants the race to be enjoyable for the 2000 athletes that are expected to partake in this race. We are provided with an Excel spreadsheet of results from a recent triathlon to help us with organizing the event. The spreadsheet contains data on the age, sex, category (professional, premier, and opening), time for each segment, and course completion time for over 3000 participants. Unfortunately, issues such as multiple athletes congesting part of a course, as well as the limited duration of road closure time, are major challenges in preparing for the race. In this paper, we provide an agent-based model that considers these factors and places the participants into divisions as well as develops a schedule of wave start times to entice athlete enrollment in future triathlons. Let the race begin!

Restatement of Problem

In order to attract professional racers and serious amateur athletes to future triathlons, the CEO of Super Tread tasks us with minimizing the amount of congestion that the athletes experience throughout the course. It is important that the path of the athlete is not significantly hindered by the presence of other surrounding athletes. Meanwhile, the mayor wants us to minimize the time that the roads are closed during the cycling and running portions of the triathlon. Specifically, the mayor does not want the road closure time to be more than 5.5 hours. Catering to both requests, we must use the spreadsheet provided to partition the athletes into divisions based on age, gender and category, as well as generate a schedule of start times for each respective wave of triathlon athletes. Placing the athletes in different divisions is for athlete satisfaction, while the wave schedule will have the dual criteria of minimizing congestion and the road closure time. We will also explore the effects of changing the segment distances on minimizing the congestion and road closure time.

We propose creating a congestion metric and a temporal metric to address the needs of the CEO of Super Tread Race Company and the Mayor of the town respectively. The congestion metric can be used to measure the amount of congestion athletes face throughout the whole event, and the temporal metric can be used to measure the amount of time roads will be closed for the event. To implement these metrics with the data from previous triathlons in a dynamic environment, we use an agent-based NetLogo model [2].

Definitions, Assumptions, and Measured Values

Definitions

Athlete: Individual participant in the race.

Course: The terrain (water or road) that the athletes traverse during the entirety of the race.

Cutoff Time: The time (hh:mm:ss) that is the dividing boundary of two divisions/groups/categories.

Division: A division is any group of athletes in the triathlon that will compete against each other for a division award. In our model, an athlete is categorized into one of eight divisions.

Passing: The act of a faster athlete overtaking a slower athlete.

Segment: Refers to the swimming, cycling, or running portion of the triathlon.

Transition Area: The area designated to proceed from one segment to another. **T1** refers to transition from swimming segment to the cycling segment, and **T2** refers to transition from the cycling segment to the running segment.

Wave: During a triathlon, a set of athletes is released periodically so as to ease congestion. This set of athletes is called a wave.

Assumptions

- Weather and road conditions, as well as the structure and topography of the terrain, are negligible for each segment in the triathlon.
  - Environmental factors are stochastic and have unpredictable effects on the race. Additionally, we assume that the Mayor would like to decide on the configuration and date of the course in a way that would reduce the effect of weather or other environmental factors. We will use this generalization in recreating the triathlon course in our computational simulation.
- The data from previous triathlons can be reasonably extrapolated to the athletes that attend the triathlon.
  - The spreadsheet provided to us has data for 3217 athletes from previous triathlons. We assume that selecting 2000 data points from the spreadsheet can accurately represent the distribution of athletes attending the triathlon. Adjusting for variances between our triathlon and their triathlon is impossible to quantify.
- The roads will be closed during the cycling and running portions of the triathlon only.

  - The swimming part of the triathlon will not require any opening in the road, as it will take place in a body of water.

- The athletes will roughly know their mean times for each segment in the triathlon and will be able to report this time to us during registration.

  - In our model, we utilize this registration information to determine the division and the wave number of the athlete. Therefore, it is imperative that all athletes have their records from past triathlons. This expectation can be advertised beforehand and allows for a controlled and innovative solution to resolve the congestion in the city.

- The age of the athlete does not impact his/her performance.

  - Age is not necessarily indicative of the physical condition of the athlete and thus does not influence his/her allocation into a division. This assumption allows us to extrapolate information from the spreadsheet to our triathlon.

- The distance covered at the transition areas is not significant.

  - Although the time spent in the transition area is significant, the distance covered during that time is not considered in a standard triathlon. The transition time will be constant even if the length of the segments are modified.

- The race course is long and/or wide enough so that multiple laps do not lead to athletes earlier in the race lapping athletes later in the race.

  - Athletes passing other athletes in a lap-based circuit or a self-intersecting course is hard to quantify in a model. Thus we assume that no such interactions occur.

- The roads for the running portion and biking portion are different and can thus be closed off at different times.

  - We assume that the roads for the biking and running portions are distinct enough that each can be closed off separately. This arrangement minimizes the time that a road is closed. This can be easily arranged by the mayor if requested.

- Athletes have constant speed within a segment and only take breaks at transition points.

  - As we cannot discretely determine each athlete's acceleration functions over time, we assume a zero acceleration. The accelerations and decelerations will average out over time, so this is a valid assumption. We use this assumption when calculating the speed of each athlete.

Measured Values

Congestion: It represents the total number of passings and athlete density during the cycling and running portions of the race. The metric associated with congestion is denoted C.

Road Closure Time: The total, continuous time that the roads are closed during the race.

Analysis of Problem

Before diving into the problem, we first analyze the properties of an ideal triathlon so that these properties can be incorporated into the model.

Our triathlon is being sponsored directly by the Super Tread Race Company. If we wish to keep this valuable sponsorship in the future, we must tailor our race to meet their goals. Our sponsor has specifically stated that they want "to ensure [that] the race is a world-class event that will attract professional racers and serious amateur athletes each year to further promote the company's brand and sales." Therefore, an ideal triathlon will promote the interests of professionals and talented amateurs in a fair, competitive fashion.

Furthermore, the Mayor has taken an interest in the triathlon from a citizens' perspective. He wishes that the triathlon is an "enjoyable event for the many amateur participants" but requests that we keep the road closed for no more than 5.5 hours, to ensure that the triathlon does not terribly inconvenience other citizens of the town. An ideal triathlon will stick to the 5.5 hour limit (with some time buffer for the time it takes to close/open the roads) while not inconveniencing some of the slower amateurs in the triathlon.

Model Design

Developing the Agent-Based Model

In order to accurately model the complex system of individual athletes affecting the aggregate congestion and road closure time, we ran multiple computer simulations using an agent-based model. We developed a model that generated 2000 athletes, each with varying speeds, and recreated the entire course of the triathlon, including transition areas.

Configuring the Athletes

The first part of our agent-based model is configuring every athlete, or agent, in the triathlon. In this part of our computational model, we will assign properties to every athlete that we deem pertinent to calculating congestion and road time closure. This exercise will only serve to organize our model and, more importantly, classify each athlete into a division.

We first scaled down the given data set of 3217 athletes to the 2000 athletes that will participate in the town's triathlon. For example, there are 7 male professional athletes in

the data set of 3217 athletes. Therefore, the number of male professional athletes in our model was:

$$\frac{2000 \cdot 7}{3217} \text{ male professional athletes} = 4.33 \approx 4 \text{ male professional athletes.}$$

Thus we randomly selected 4 data points from the 7 in the original data set, ensuring that the ratios of each category were conserved.

Using the generated data set, we assign athletes a swim speed, T1 (transition area 1) speed, bike speed, T2 (transition area 2) speed, and run speed. These individual speeds are calculated by dividing the length of the track by the time it takes to complete the segment.

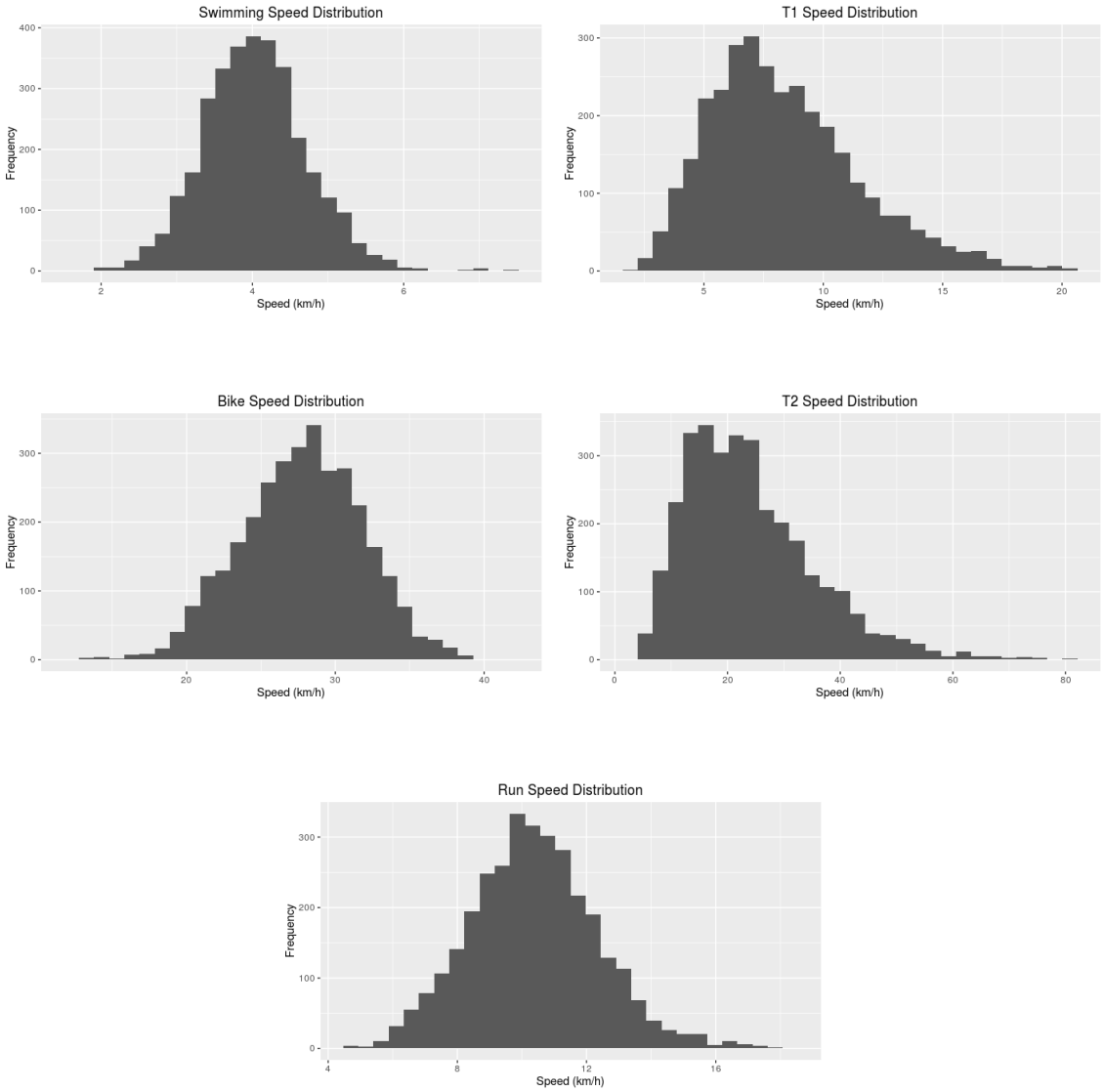


Figure 1: Distribution of athlete speeds for each segment

The figures above depict the distributions of the speeds that is representative of the 2000 participants for each segment. After scaling this distribution down to 2000 participants, we assign each speed randomly to an athlete, which determine his or her course completion time.

We use the course completion time to make divisions for the athletes, as the athletes would want to be able to compete with other athletes of similar skill. Divisions are based on sex and projected course completion time, which are both gathered from the results of the registration survey (see Appendix). We have four divisions for males and females each, assigned the encouraging names of Black Diamond, Diamond, Gold, and Silver. Each division represents a quartile of the distribution of course completion times. If we display the course completion times (of the corresponding sex) in a box-and-whisker plot, we can determine the cutoff times for each division. The following table describes the total times required to be placed in each division.

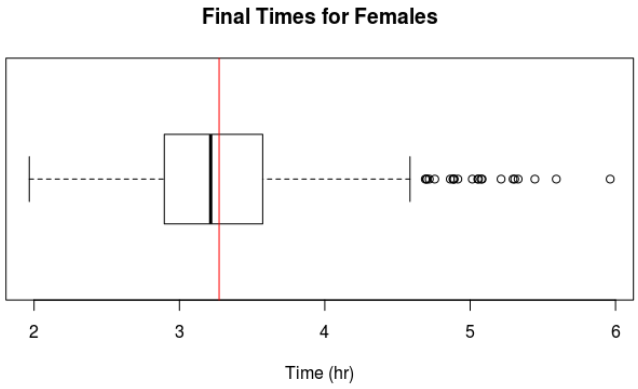


Figure 2: A box-and-whisker plot of the final times from our data for females. Red line represents the mean.

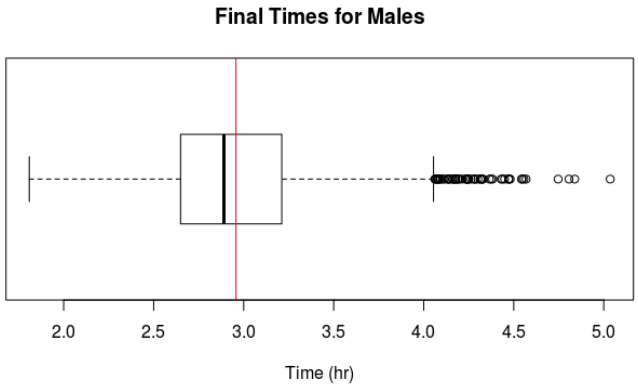


Figure 3: A box-and-whisker plot of the final times from our data for males. Red line represents the mean.

Loosely, the Black Diamond division corresponds to participants who rank in the top 25% of course completion time, Diamond represents the 25% - 50% tier, Gold the 50% - 75% tier, and Silver the bottom 25%. The cutoff times were rounded to a convenient multiple of 15 minutes. These clean, rounded values have a practical purpose: participants will not list their times on the survey as "2:34:23"; they will report it as "2:30:00". It is also very simple for us to divide the participants with the cutoffs of 2:30:00 and 3:00:00 rather than that of 2:22:43 and 2:56:45.

Hence, we will be able to assign each agent a speed as well as division in our agent-based model.

Designing the Triathlon Course

It is essential that our computer-simulated course is as close to a real-life triathlon as possible. We begin by dividing the entire course into five "tracks", or portions of the course, one for each segment and transition area. We then assign "patches", which are units of area (in this case 1 patch = 1 km^2), to each track. Thus, as each wave is called, athletes start at the swimming segment, proceed through the transition area where they pick up their bike, bike through the biking patches, drop off their bike in the second transition area, and run through the running patches before crossing the finish line at the right-most patch.

As an auxiliary note, the width of the course becomes a slight consideration, specifically in regard to the cycling and running segments. Since the terrestrial parts of the race occur on town roads, we will assume that the roads have a width of twelve athletes. In other words, twelve athletes will be able to compete side by side at the same time. As for the swimming portion of the triathlon, the track takes place in a body of water, and there will be ample space to compete. As a result, we assumed that twenty-four athletes will be able to compete side by side at the same time.

In order to provide a pleasant experience to all participants, the triathlon should aim to reduce any steric hindrances athletes experience on the course. Moreover, all divisions should be able to proceed with similar congestion, i.e. lower congestion for one group should not come at the cost of the other group. To accomplish this, we propose to split the segments into three lanes:

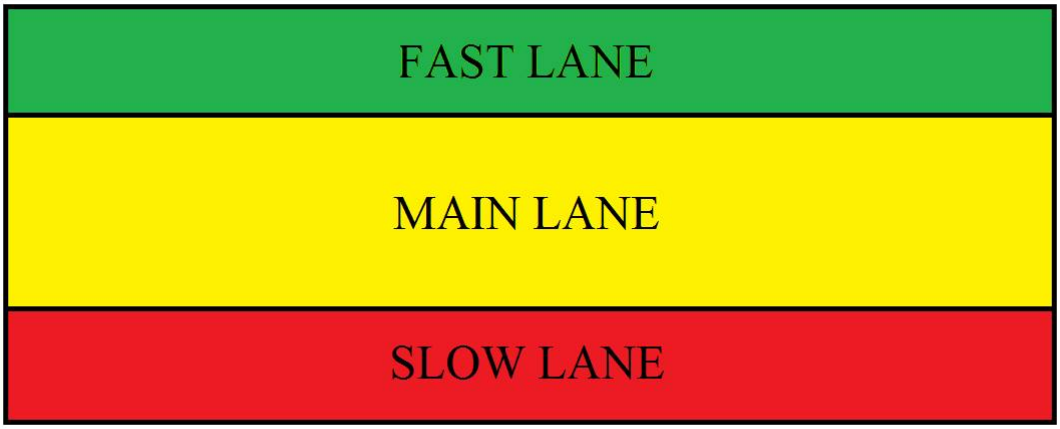


Figure 4: Pictorial representation of the three different lanes. Notice the 1:2:1 ratio.

Thus, individuals who do exceedingly well at biking, for example, will use the fast lane to bypass the athletes using the main lane. On the other hand, individuals who do poorly will use the slow lane to prevent themselves from slowing down the main lane. By having the cycling and running tracks split into three lanes, an athlete only has to consider those in their lane, so it heavily cuts down on the congestion of the overall triathlon for all participants. The slow lane also has some practical utility: if one's bike breaks down or one has a severe cramp, one can just use the slow lane to fix their bike or catch their breath without slowing down the main group.

We implemented the lanes in our agent-based model. We devised a two-dimensional coordinate system for patches, where the horizontal patches indicate the length of the course, and the vertical patches indicate the different lanes.

In regards to the allocation of athletes into their respective lane, we decided that the lanes will be based on the quartiles of the track time distribution due to the 1:2:1 nature of the lanes. Thus, the athletes in the first quartile will use the the fast lane, and the athletes in the fourth quartile will use the slow lane, while everyone in between will use the main lane. Since this was done on the three segments of the race (swim, bike, run), it is theoretically possible for an individual to be in the fast lane for swimming, the main lane for running, and the slow lane for biking (Note that we use the quartiles of the course completion time distribution for determining divisions, but for the lanes we are using the quartile of individual track time).

Attaining the Optimum Schedule for Wave Start Times

The optimum schedule for wave start times is one that minimizes the congestion experienced by the athletes and minimizes the road closure time. We can model congestion and road closure time and seek to reduce those quantities in our agent-based model.

Defining Congestion

It can be seen time and again that participants naturally form "chains" or groups that move together [3]. Once an athlete finds his or her way into a group, he or she may find it difficult to deviate from the overall group speed. Thus, if an athlete wants to move faster than the group that he or she is in, he/she will have to pass other athletes in the group, which can become a severe hindrance. In fact, we find that the congestion is logarithmic with respect to the number of people around the athlete.

Let $C(N)$ be the congestion that an athlete experiences when there are N people around him or her. We reasoned that at first thought, the congestion should be linear with N, where the change in congestion is constant and is independent of the number of people. However, if we considered the limiting case, it becomes clear that when $N \to \infty$, $\frac{\partial C}{\partial N} \to 0$. In other words, the change in the perception of congestion is rather negligible when the number of surrounding people is large, i.e. having twenty people surrounding the athlete versus ten would not make the athlete feel twice as congested. Hence, mathematically:

$$\frac{\partial C}{\partial N} \propto \frac{1}{N}$$

Thus, we have that:

$$C(N) \propto \int \frac{1}{N}\, dN = \ln(N)$$

*We use partial derivatives here because C can be a function of multiple variables.

Furthermore, the congestion becomes worse if the athlete has to make multiple passings. It is easy to see that the congestion should be linear with respect to the number of passes, i.e. doing 2 passes is twice as hard as doing 1 pass. Hence, we can define our congestion metric in terms of the number of people around an individual and the number of people that an individual needs to pass.

Definition 1: Congestion Metric

Let P be the set of all passes that occur in the simulation.

$$C = \frac{1}{t} \sum_{p \in P} \ln(N) \ ,$$

where C is the congestion metric, N is the number of people surrounding an athlete, and t represents simulation time, which normalizes C with respect to time.

*When applying the congestion metric in our agent-based model, we also included the individual in N.

After we defined congestion, we implemented it in our agent-based model by having the athletes calculate the number of people around them as well as the number of people they passed.

Defining Road Closure Time

Road closure time is the time that the roads are closed in order to accomodate the triathlon. We assumed that the roads will not be closed during the swimming segment of the race since it will take place in water. Thus the road closure time is the time it takes to complete the cycling and running segments. We accounted for this in our agent-based model by measuring the time it took to complete the cycling and running portion of the event. However, since different roads will be closed for the cycling and running segments, we will make sure neither segment times exceed more than 5.5 hours.

Reducing Congestion and Road Closure Time: Constructing the "Wave Packet"

In the 2016 Rio Olympics, the game-coordinator held two triathlons over a period of three days (one for male and female), with 55 athletes competing in each triathlon [4]. In the local town, we are expecting 2000 athletes to compete in the upcoming triathlon, which will be held for only one day. We must therefore consider how to deploy all of the athletes while not closing the roads for more than 5.5 hours. Hence, the road closure time is dependent on the deployment time of the waves. Since we cannot deploy more than 55 athletes per wave (because we are abiding by Olympic triathlon standards), we must define the road closure time with respect to the time spent between the deployment of two consecutive waves. First, however, we must consider the type of athletes that will be in each wave.

Recall the three lane model, where each track represents one or two quartiles of the distribution of their respective track times. If we consider all of the track times together, in other words the course completion time, we can generalize the lanes that each athlete will follow. We can then split all of the athletes to the respective lanes that they will tread. This is different than simply splitting the athletes into divisions because now we will have three groups instead of four (one for each lane). We define "group 1" to be the group of the fastest athletes, who will generally use the fast lane; "group 2" to be the group of the average athletes, who will generally use the middle lane; and "group 3" to be the group of the slowest athletes, who will generally use the slow lane.

Group 1 = Athletes in Fast Lane
Group 2 = Athletes in Main Lane
Group 3 = Athletes in Slow Lane

Each group is then ordered into ranks. The athletes that in each rank are determined by 15 minute intervals, starting from the cutoff time for each group and based on reported total time (i.e. the time that they provided on the survey). For example, if group 1 are those with a reported total time better than 2:45:00, the group would be split into ranks of 2:30:00 to 2:45:00, 2:15:00 to 2:30:00, ... Splitting the groups into intervals of less than 15 minutes would be highly impractical as people do not report their times as 2:36:45. We note that

these groupings do not depend on sex. This is because the faster ranks should always be deployed before the slower ranks otherwise faster individuals will be trapped behind slower individuals (which would add to our congestion).

The deploying mechanism for the groups is as follows: It has 55 individuals that need to be deployed in every wave (we call this a wave-packet), with some set amount of time between each wave-packet. At the start of the triathlon, each athlete is given a rank to inform them of their wave-packet. Due to the widths of each lane, one-fourth of the 55 individuals of each wave-packet should come from group 1, one-half should come from group 2, and one-fourth should come from group 3. When selecting from a group, the individuals should come from the highest ranks, then the next highest ranks, and so on to prevent faster individuals from being trapped behind slower individuals. Ideally, the last person from each group should finish the triathlon at roughly the same time so that we are not keeping the roads closed for just a handful of people in one group. Since group 1 is naturally fast and group 3 is naturally slow, in order to make group 1 finish at the same time as group 3, there should be fewer individuals in group 3 and more individuals in group 1. Upon testing this in our agent-based model, we found that defining group 1 to be those athletes with a time better than 2:45:00 and defining group 3 to be those athletes with a time worse than 3:30:00 causes all 3 groups to finish at the same time.

Definition 2: Wave-Packet

A **Wave-Packet** is a group of 55 athletes that are deployed at the same time. 25% of the athletes come from the highest ranks of Group 1 that have not yet been deployed, 50% of the athletes come from the highest ranks of Group 2 that have not yet been deployed, and 25% of the athletes come from the highest ranks of Group 3 that have not yet been deployed.

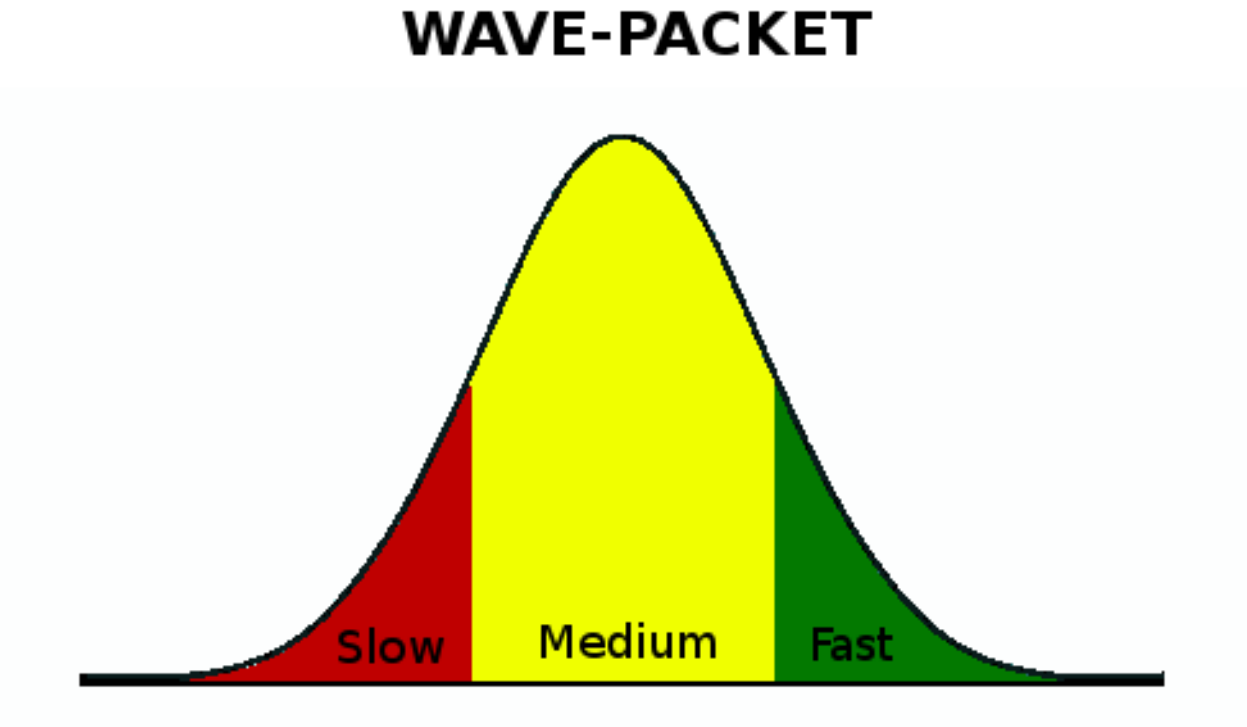


Figure 5: Wave-Packet Structure (Note: Areas not to scale)

The grouping/ranking system minimizes the intersections of different wave-packets over time. If the slowest athletes go first, they will take the fast lane and will interfere with athletes in the next wave-packet much like actual waves.

With our deploying mechanism in place, we just need to adjust the time between wave-packets. A higher time between wave-packets leads to less interaction between wave-packets, which leads to a lower congestion value. Therefore, a large interval of time between wave-packets is ideal. However, we are held back by the fact that we only have 5.5 hours to run our triathlon. We can improve on this number by realizing that if the biking track is in use but the running track is not, we can potentially have the roads for the running track open and vice versa. This of course would have to be specially arranged. With this optimization, we can simply increase the time between waves until the road closure time approaches 5.5 hours.

In summary, we simulate the triathlon using the following algorithms:

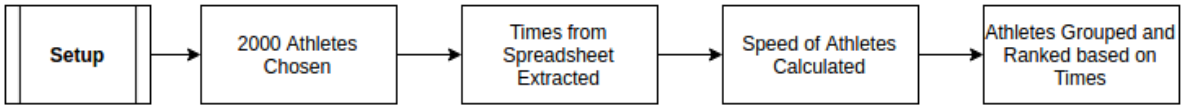


Figure 6: Setup Algorithm

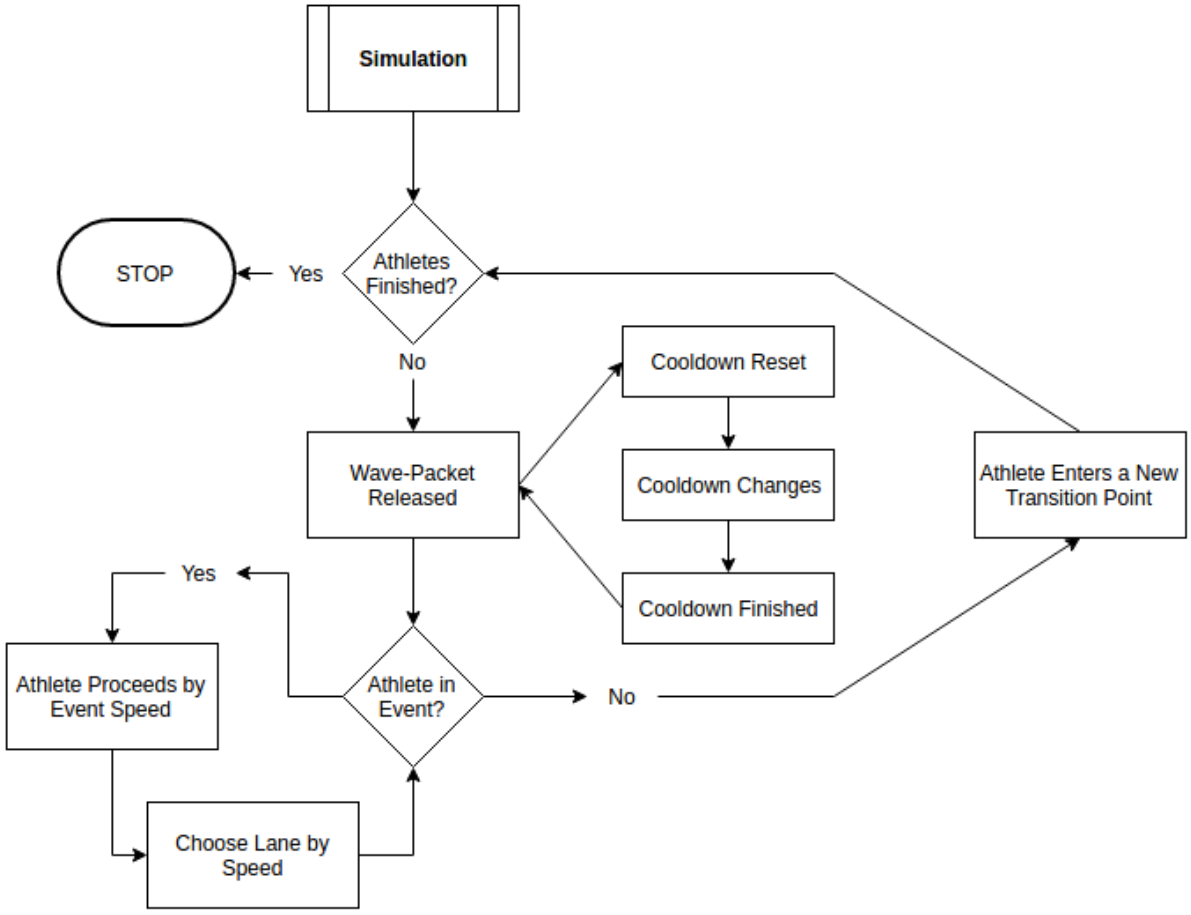


Figure 7: Simulation Algorithm

The complete NetLogo code can be found in the Appendix.

Results

Exploring Congestion and Road Closure

Divisons

The first task of the problem is to devise divisions to promote a healthy competition between the athlete. We created the following divisions for the athletes.

Division	Male Cutoff Time	Female Cutoff Time
Black Diamond	2:30:00	2:45:00
Diamond	3:00:00	3:15:00
Gold	3:30:00	3:45:00
Silver	All Other Times	All Other Times

Figure 8: Table of the eight divisions along with cutoff times.

Each category (Black Diamond, Diamond, etc) represents a certain skill level which was quantified by finding the quartiles of the course completion times of the previous triathlon. By placing the athletes into their respective division by using the registration survey, we conduct a competitive environment where athletes in the same division have a similar completion course time.

Schedule of Wave Start Times

In regards to generating a schedule of wave start times, we created wave-packets of 55 athletes. The wave-packets were then divided into groups, where the groups approximately reference the lane that the athlete will remain on (fast, main, slow). The groups are further divided into ranks to determine the order of deployment. The following describes the wave-packet structure of the triathlon.

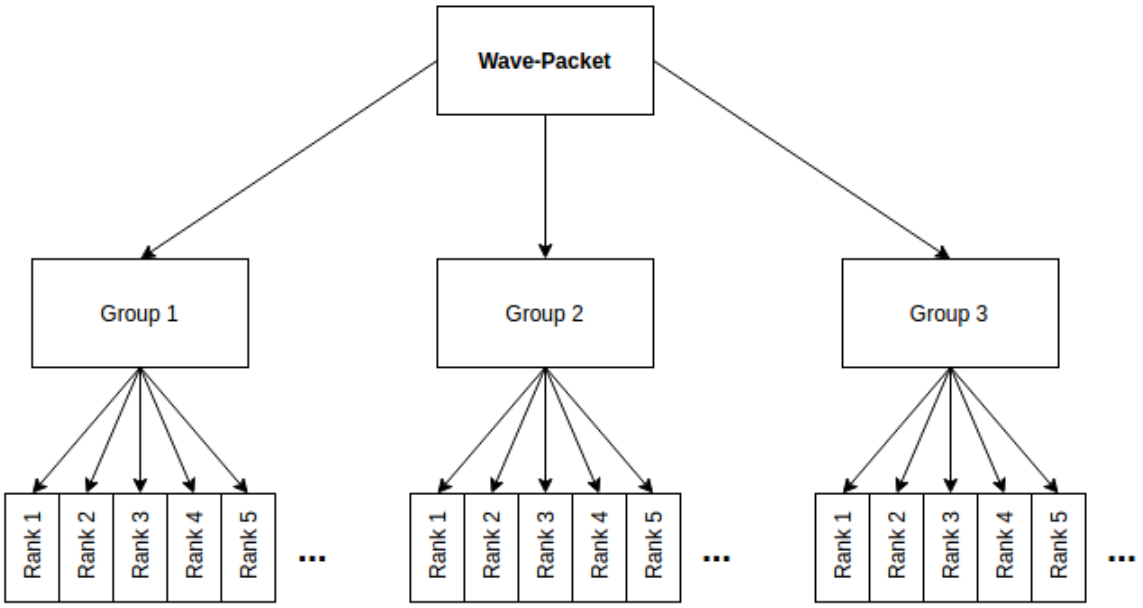


Figure 9: Wave-Packet Partitioning

Time Between Wave-Packets (s)	CM	BTM (hr)	RTM (hr)
200	100.03	4.49	4.44
240	71.65	5.34	5.22
260	59.48	5.74	5.62

From this table, we can see that increasing the time between wave-packets results in decreased congestion but increased road closure time. If we allow road closure time to be as close to 5.5 hrs as possible, we find from the graph below that a wave-packet separation of 240 seconds, or 4 minutes, gives a relatively low congestion value of 71.65.

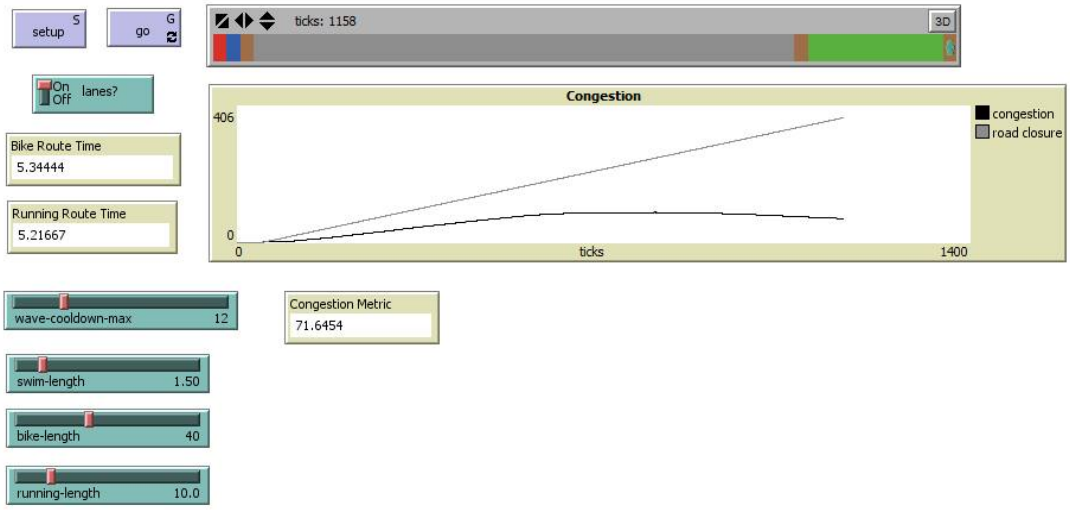


Figure 10: Total Congestion was 71.6

It can be seen from the data that a separation of 4 minutes between waves minimizes congestion while also keeping road closure time under 5.5 hrs. Therefore the optimal time between waves was 4 min, creating the following schedule.

Time	Wave-Packet
8:00 AM	Wave-Packet 1
8:04 AM	Wave-Packet 2
8:08 AM	Wave-Packet 3
⋮	⋮
10:16 AM	Wave-Packet 35
10:20 AM	Wave-Packet 36
10:24 AM	Wave-Packet 37

This table is the generated schedule if the triathlon began at 8:00 AM. Once the first wave-packet is deployed, we wait four minutes until the next packet can be deployed. For the full table, see the Appendix.

Changing Segment Lengths (Sensitivity Analysis)

To test the sensitivity of our model to the various parameters, we changed the time between wave-packets' release as well as the lengths of the three segments of the race (Note: time between wave-packets is set to 200 s, see Appendix for full data).

Swim Length (km)	CM	BTM (hr)	RTM (hr)
0.50	90.32	4.65	4.58
1.50	100.03	4.49	4.44
2.50	94.99	4.85	4.81
3.50	96.13	5.03	4.85

Bike Length (km)	CM	BTM (hr)	RTM (hr)
20	85.93	4.07	4.42
30	85.14	4.46	4.57
40	100.03	4.49	4.44
50	95.55	5.30	4.96

Running Length (km)	CM	BTM (hr)	RTM (hr)
5	77.84	4.79	4.25
10	100.03	4.49	4.44
15	93.68	4.79	5.67
20	101.06	4.73	6.79

Given these results, we can say that increasing or decreasing the swimming and biking lengths from its original values leads to a general decrease in the congestion of the triathlon. On the other hand, for running length, the correlation is less clear as increasing it to 20 led to a larger congestion whereas increasing it to 15 led to a smaller congestion. However, as the model is probabilistic in its data selection, we cannot be definitive about the congestion values without running more trials. The road closure times, on the other hand, provide a more concrete measure of the impact of the length parameters. For example, changing the bike length from 40 km to 50 km does not push the road closure time above the 5.5 hr limit, but increasing the running length to even 15 km means that the 5.5 hr criterion will no longer be met.

One plausible explanation for the positive correlation between length and congestion at low lengths is the "catch-up factor". Suppose that there are two athletes biking, one of whom is slightly faster than the other. However, suppose that the faster athlete is behind the slower athlete. With a longer biking length, the faster athlete has a bigger window to

catch-up. By catching-up, the faster athlete adds to the congestion. On the other hand, with a longer biking length, the distribution of athletes is more spread out (differences in speeds matter more), decreasing the congestion.

If we want to minimize congestion in our model, one realistic way is by decreasing the size of the running section to only 5 km as it decreased congestion from 100.03 to 77.84 in our model, larger than any other change. However, this would deviate from the Olympic standards and thus make the event less professional.

Justification

Our model sought to satisfy the needs of both professionals and amateurs while sticking to the constraint of a road closure of 5.5 hours. There are multiple ways in which our model tailors to the needs of professionals and talented amateurs: Given our group 1, group 2, and group 3 system, all of our talented male and female athletes will find themselves at or near the highest rank of group 1. This allows them to start the triathlon with little to no waiting time. In addition, they generally experience less congestion since they are almost always in the fast lane with very few other waves in front of them. Most importantly, starting near the beginning implies that the first individuals to finish are these talented individuals. At the finish line, this adds to the reward from an athlete point-of-view and to the excitement from a spectator point-of-view. The triathlon can be also seen as legitimate (thereby carrying weight by the talented athletes) as it uses an Olympic-style format.

On the other hand, the triathlon also tailors to the more amateur side of the triathlon. The triathlon track includes a slow lane that allows athletes that are falling behind the rest of the group to slow down and rest without impacting everyone behind them. Even though the more talented athletes experience a bit less congestion then the rest of the participants, the burden does not fall heavily on one particular group. In addition, the divisions like Silver and Gold allow lesser skilled individuals to compete amongst themselves, giving everyone a chance of earning a high placing.

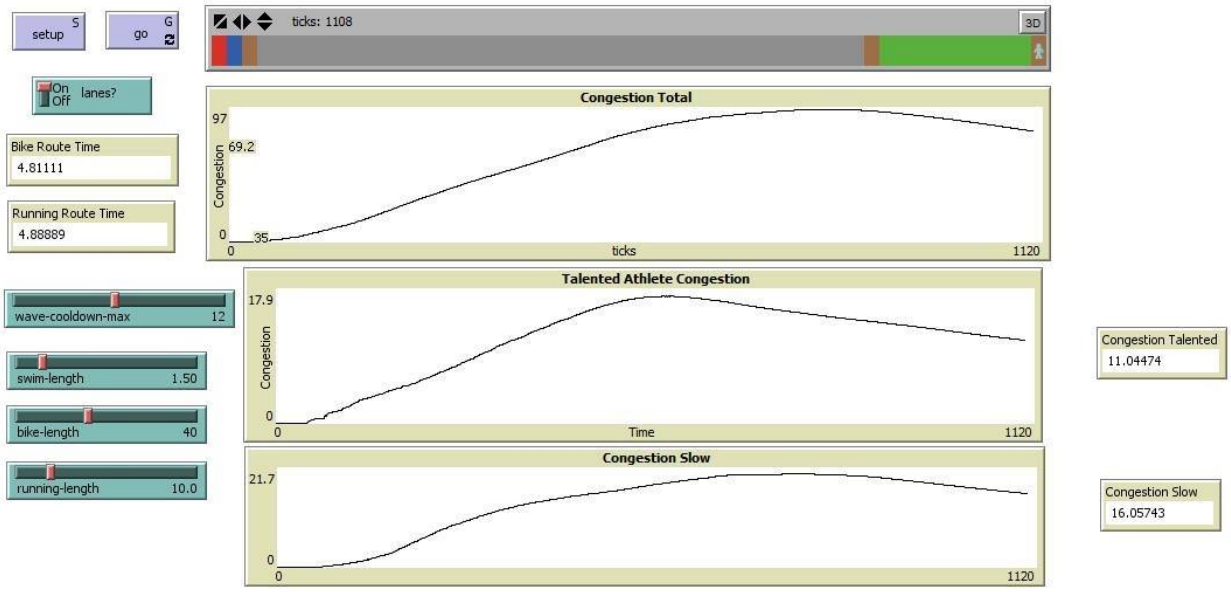
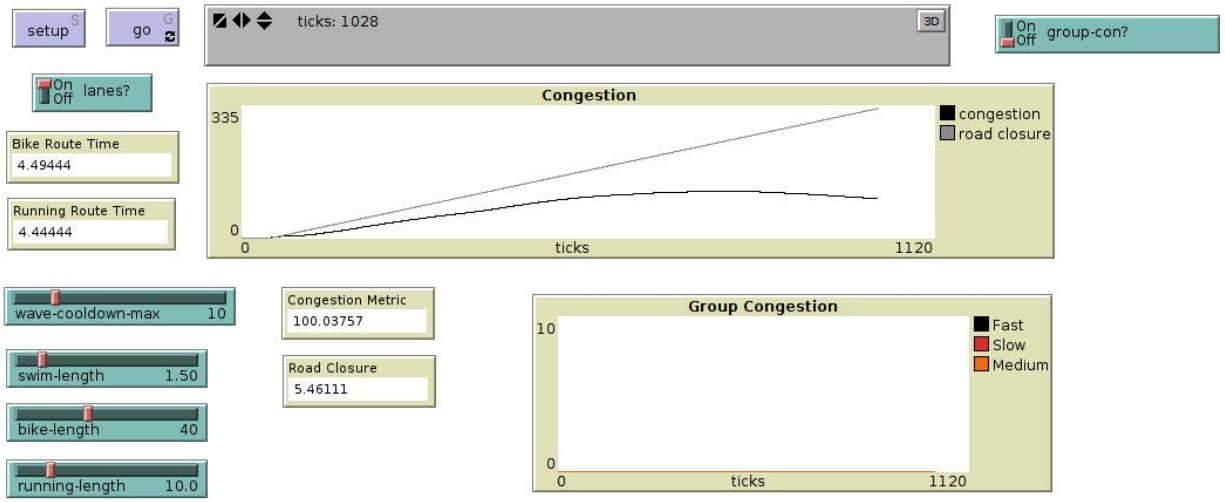


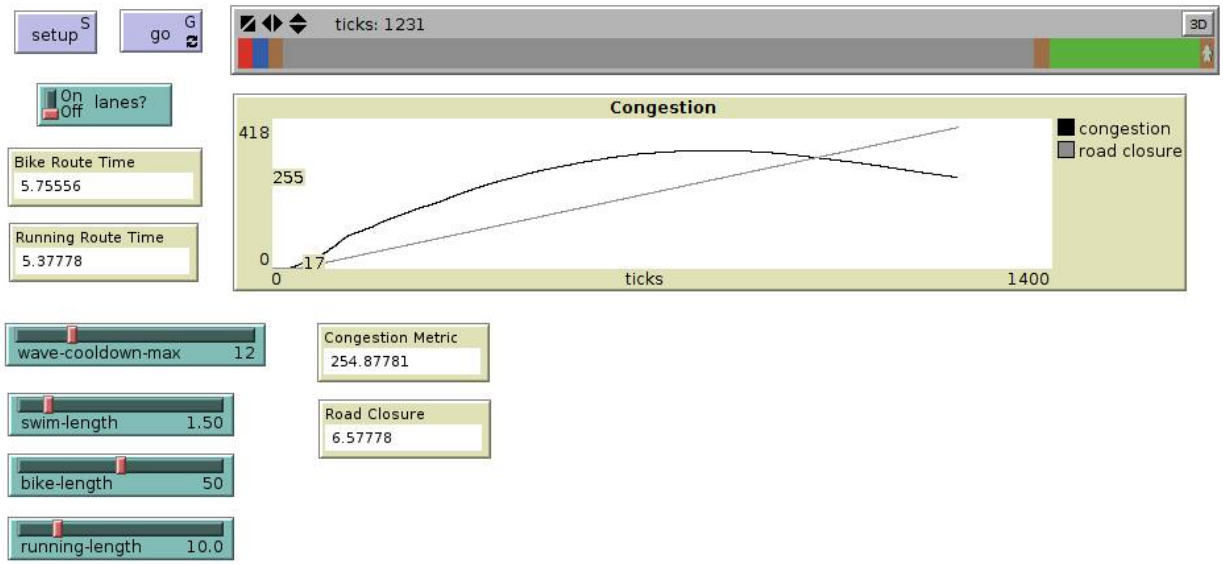
Figure 11: Even though the congestion for Black Diamond is low, the burden does not fall heavily on the other groups

Our model assumed that we used separate road systems for the bike and running sections. If we consider a singular road system, we obtain the following results:



We can easily see that road closure time has risen to 5.46 hours, which may not leave enough time for the road to transition between closed and open. On top of this, the time between wave-packets has decreased to 200 seconds. The congestion metric itself has risen to almost 100.03, a sharp 40% increase (29% decrease) over the 71.6 metric given by the separate system model. Therefore having two separate closures provides a strong benefit to any triathlon.

Our model used a three lane system to prevent faster and slower athletes from congesting one another. If we consider a one lane system instead, we obtain the following results:



This is a bigger disaster than the singular road system model: the congestion value has more than tripled to 254.9 (256% increase). Just by adding in two more lanes, we were able to decrease congestion by 72%. Most likely, using a single lane system would lead to far more complications and general congestion to the point that the triathlon would no longer be a fair, competitive contest of skill.

Our model characterized athletes into three separate groups in order to effectively use the three lanes. If we instead group randomly, we obtain the following results:

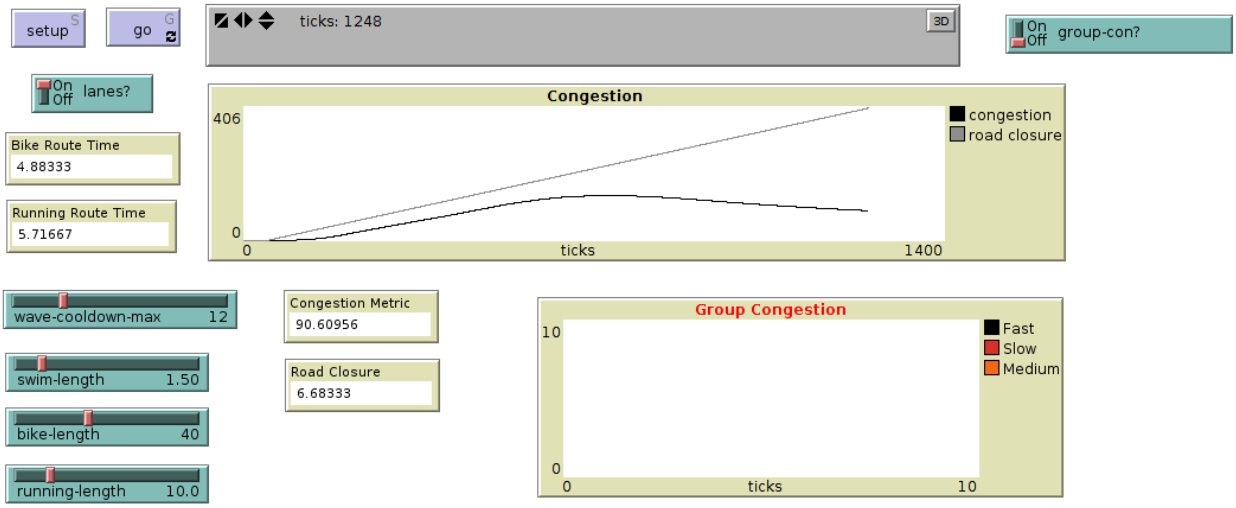


Figure 12: Wave-packaging is done arbitrarily

The congestion value here has risen to 90.6, a 27% increase (21% decrease) over the congestion value of 71.6. The increase can be attributed to greater interference among the wave-packets. This is also an indication that the remaining 70% or so of the congestion is primarily "intraference" that occurs naturally within a wave-packet.

Strengths and Weaknesses

Strengths

- Our agent-based model accounts for the complex interactions between the individual athletes and calculates a congestion metric based on the number of people athletes pass as well as the number of people around each athlete. This provides us with an easy way to judge the efficacy of a wave-packet schedule. Our model also calculates the amount of time the roads need to be closed, which tells us if a wave schedule will fit within the required road closure time of 5.5 hours.

- We divide the race course into three different lanes based on the speed class of the athletes. This allows us to minimize congestion as the faster athletes will not have to continuously zig-zag around slower athletes and vice versa. By using this 3-lane system, we decrease the congestion by 72%.

- Our model uses wave-packets composed of a distribution of athletes. Even though some athletes are faster, they will take the faster lane, and will not run into as many of the slower athletes from the previous wave-packet. Similarly, the slower athletes will not be forced aside by faster athletes in a successive wave. Furthermore, since the athletes in each wave-packet are chosen by their group (based loosely on speed quartiles) and rank within a group (based on 15-minute time intervals), we know that the fastest people will be sent out first, which will motivate them to come back for the next triathlon, strengthening our sponsorship from Super Tread Race Company.

- Our model incorporates many aspects of the Olympic Triathlon, including the number of athletes per wave (55) and the lengths of the segments of the race (1.5 km swim, 40 km bike, 10 km run). This makes the triathlon more alluring to professionals and will thus attract more pro racers and serious amateurs, as per the Super Tread Racing Company CEO's discretion [4].

- We incorporate four divisions for each sex into our triathlon model. This motivates more people to join as they have a higher chance to win in their category. We also ensure that almost all Premier and Professional racers will be in the same division, which increases the competitiveness of the top division.

Weaknesses

- Our race course is linear and thus does not account for race courses in which there are multiple short laps. We could not account for the road infrastructure of the town in

which the triathlon is held, so we generalized it to a linear case. Wide roads (such as highways) can be subdivided to form loops.

- Our model heavily relies on the data received from the participants. For example, if the participants were to underestimate by 5%, our model will not group athletes properly, which will compound to large inefficiencies in terms of congestion and road closure time.

- As we do not have an explicit formula for calculating the time between wave-packet releases or the group distributions, we have to compute the optimal time for every setting. This means that it will take large amounts of computational power to perfect the values.

- As some parts of the model (mainly the data set generation) are probabilistic, we have to run multiple trials of the model at each setting in order to obtain metric values for that setting. However, since the model is computationally intensive, this is difficult to do with our limited computing power.

Conclusion and Future Work

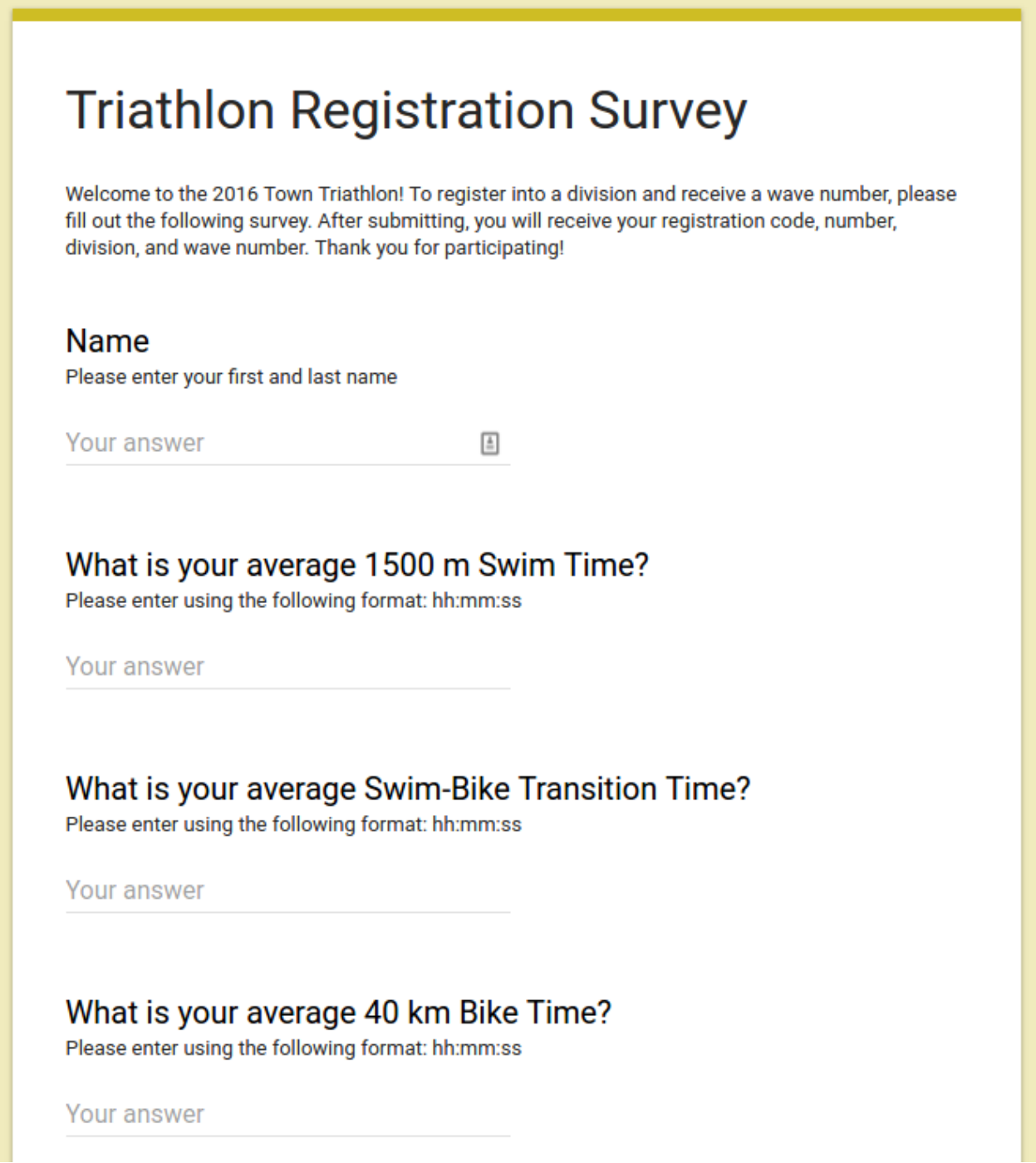
We were tasked with properly modeling and organizing a schedule for a local youth organization's triathlon. The Mayor of the town permitted 5.5 hours of time during which roads could be closed. Furthermore, the main sponsor of the triathlon requested that professional athletes be allowed to attend so as to increase the future prospects of another triathlon. For this to occur, the sponsor asked us to minimize the congestion experienced by athletes during the event. We began by determining the method by which registered athletes would be divided to foster friendly competition and yet allow all contenders to have a decent chance of winning awards. We then chose an optimal layout for our track and used a wave-packet approach to ease congestion. Our simulations were run in an agent-based model developed in NetLogo.

We determined that athletes should be divided into eight divisions: Female Black Diamond, Male Black Diamond, Female Diamond, Male Diamond, Female Gold, Male Gold, Female Silver, and Male Silver. This method of separation removes the necessity of categorical qualifiers, such as being a premier or professional athlete, to compete with other members in one's skill level. Furthermore, this allows all athletes to compete relatively fairly for commendations. The time interval between the deployment of wave models was optimized to a value of 240 seconds, or 4 minutes. Therefore, athletes leave in waves of 55 athletes spaced out in 4-minute intervals.

Since we were constrained by computational power, we were unable to iterate through all the possible variations of parameters. In the future, we hope to use faster computational methods to achieve this goal. If we receive information on the topography and circuit of the triathlon from the Mayor, we hope to also take into account any lapping that occurs in our wave model so as to optimize the time between wave-packets even further.

Appendix

Registration Survey



Triathlon Registration Survey

Welcome to the 2016 Town Triathlon! To register into a division and receive a wave number, please fill out the following survey. After submitting, you will receive your registration code, number, division, and wave number. Thank you for participating!

Name

Please enter your first and last name

Your answer

What is your average 1500 m Swim Time?

Please enter using the following format: hh:mm:ss

Your answer

What is your average Swim-Bike Transition Time?

Please enter using the following format: hh:mm:ss

Your answer

What is your average 40 km Bike Time?

Please enter using the following format: hh:mm:ss

Your answer

What is your average Bike-Run Transition Time?

Please enter using the following format: hh:mm:ss

Your answer

What is your average 10 km Run Time?

Please enter using the following format: hh:mm:ss

Your answer

Please enter your weight in pounds

Your answer

SUBMIT

Never submit passwords through Google Forms.

Selected Simulation Results

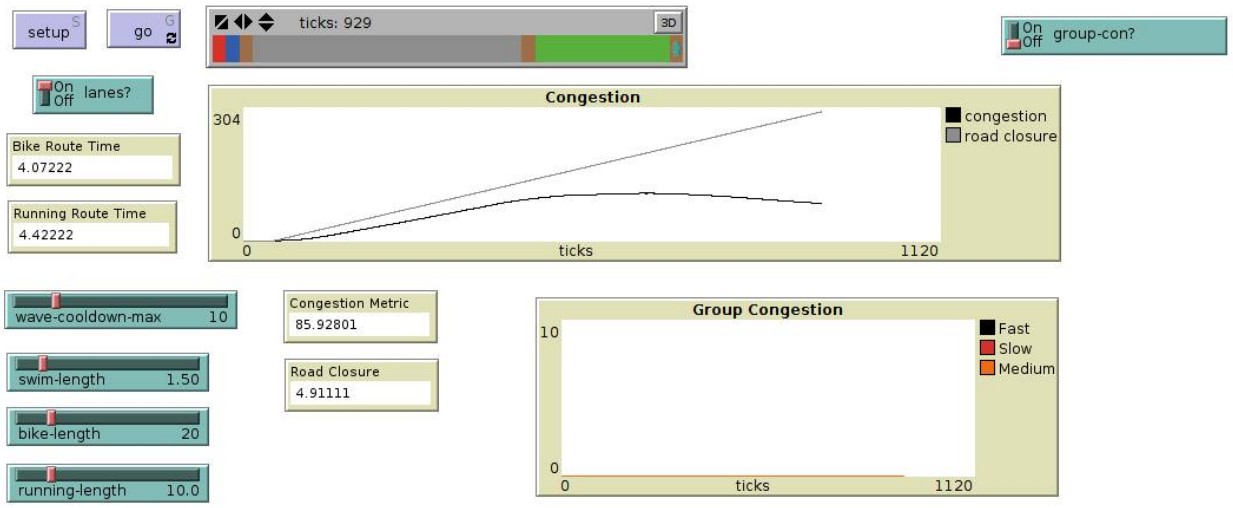


Figure 13: Time between waves: 200 s; Swim Length: 1.5 km; Bike Length: 20 km; Running Length: 10 km

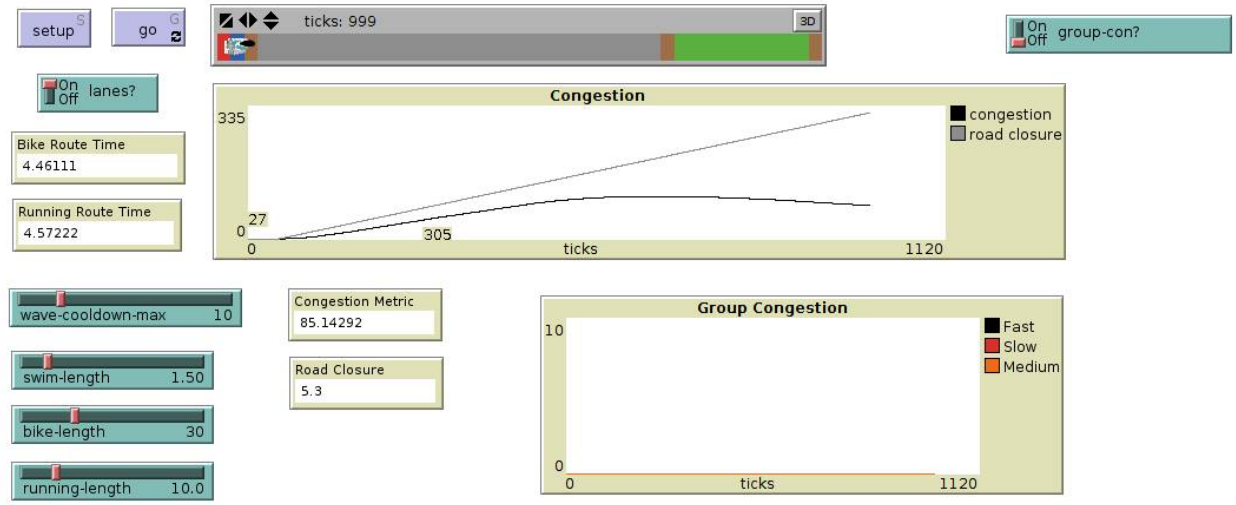


Figure 14: Time between waves: 200 s; Swim Length: 1.5 km; Bike Length: 30 km; Running Length: 10 km

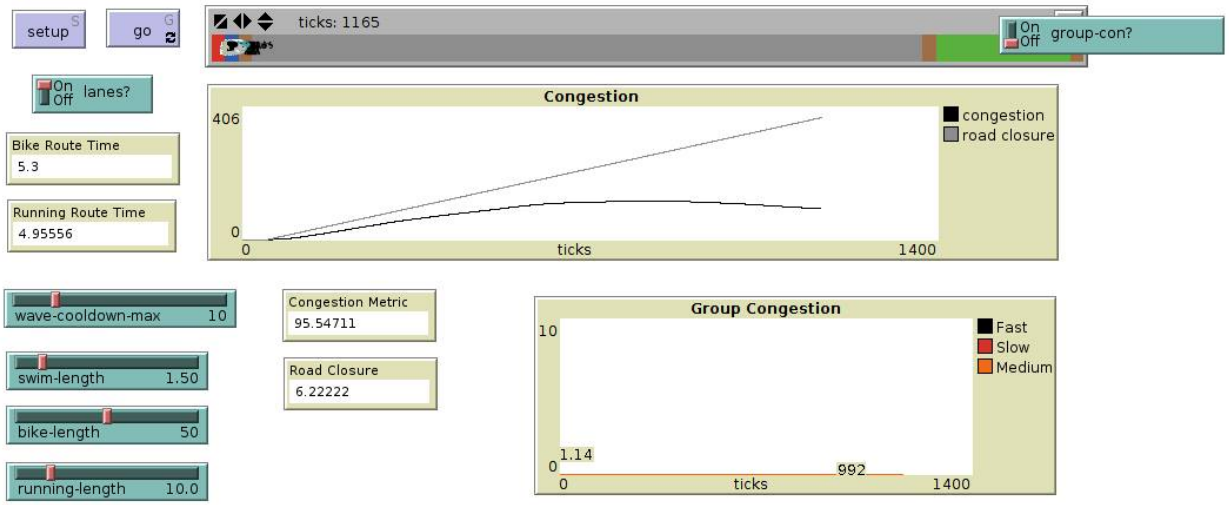


Figure 15: Time between waves: 200 s; Swim Length: 1.5 km; Bike Length: 50 km; Running Length: 10 km

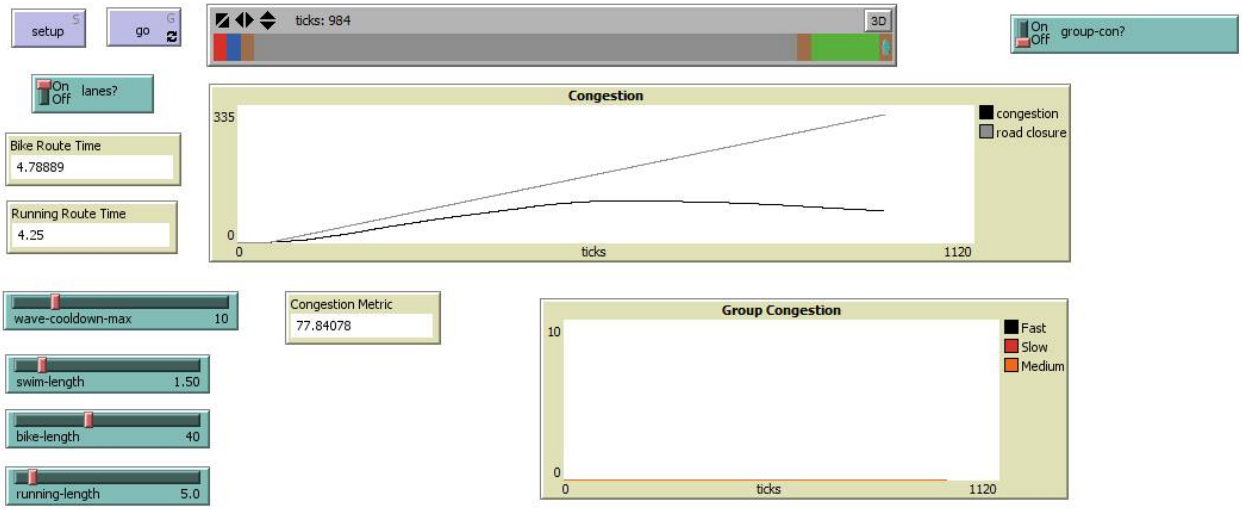


Figure 16: Time between waves: 200 s; Swim Length: 1.5 km; Bike Length: 40 km; Running Length: 5 km

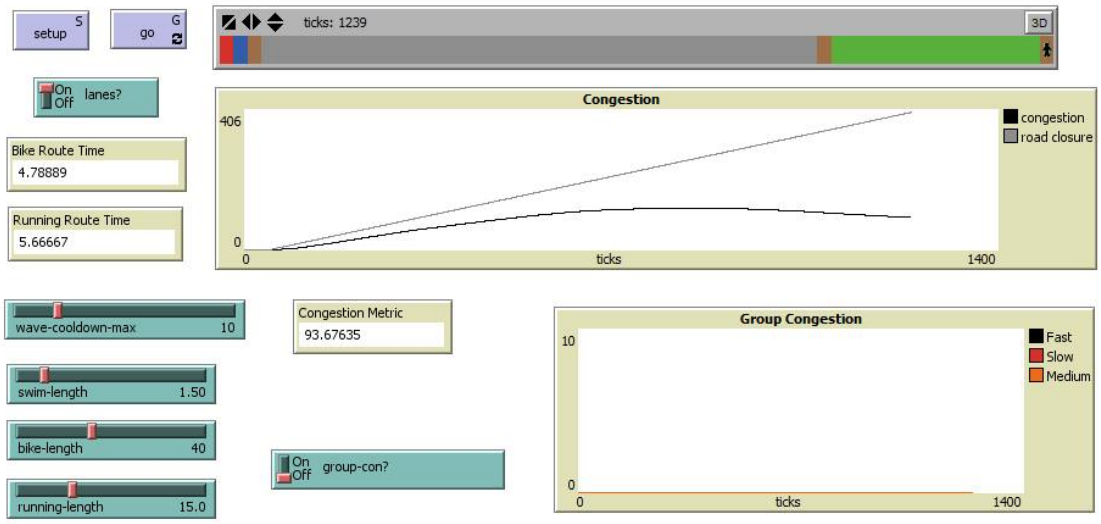


Figure 17: Time between waves: 200 s; Swim Length: 1.5 km; Bike Length: 40 km; Running Length: 15 km

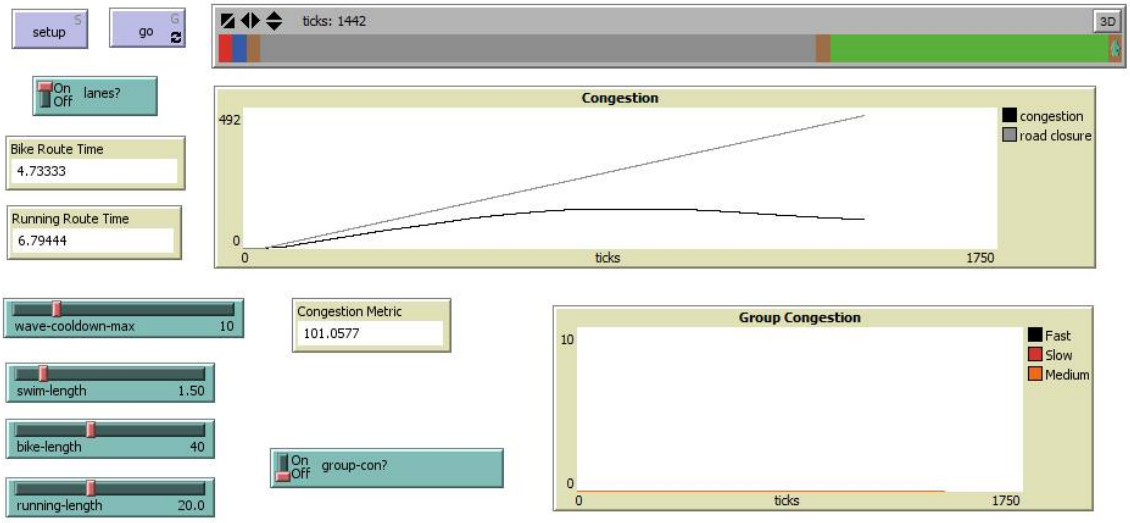


Figure 18: Time between waves: 200 s; Swim Length: 1.5 km; Bike Length: 40 km; Running Length: 20 km

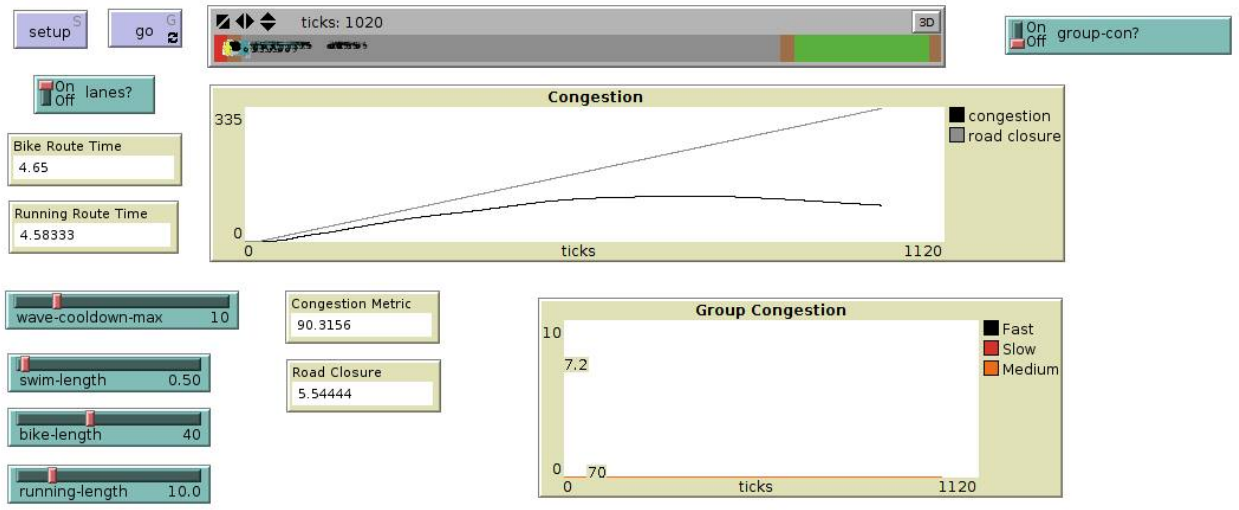


Figure 19: Time between waves: 200 s; Swim Length: 0.5 km; Bike Length: 40 km; Running Length: 10 km

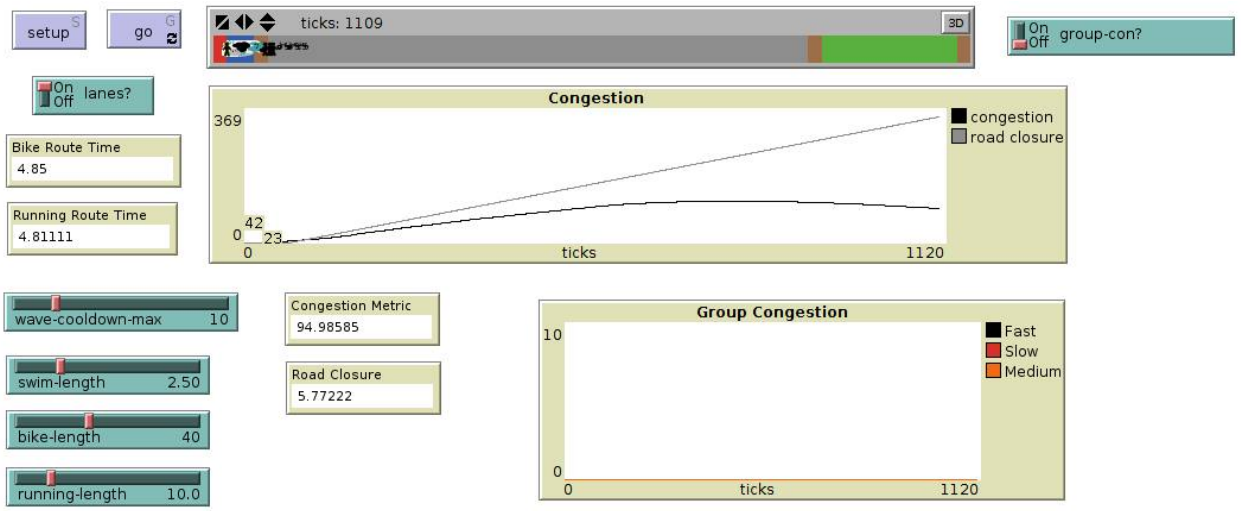


Figure 20: Time between waves: 200 s; Swim Length: 2.5 km; Bike Length: 40 km; Running Length: 10 km

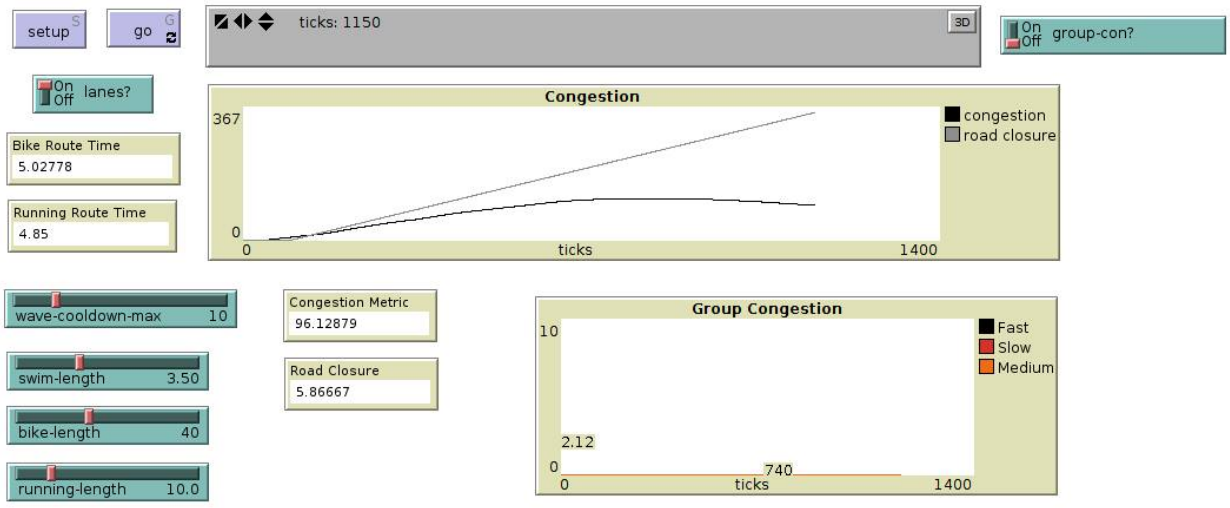


Figure 21: Time between waves: 200 s; Swim Length: 3.5 km; Bike Length: 40 km; Running Length: 10 km

Complete Wave Schedule

Time	Wave-Packet
8:00 AM	Wave-Packet 1
8:04 AM	Wave-Packet 2
8:08 AM	Wave-Packet 3
8:12 AM	Wave-Packet 4
8:16 AM	Wave-Packet 5
8:20 AM	Wave-Packet 6
8:24 AM	Wave-Packet 7
8:28 AM	Wave-Packet 8
8:32 AM	Wave-Packet 9
8:36 AM	Wave-Packet 10
8:40 AM	Wave-Packet 11
8:44 AM	Wave-Packet 12
8:48 AM	Wave-Packet 13
8:52 AM	Wave-Packet 14
8:56 AM	Wave-Packet 15
9:00 AM	Wave-Packet 16
9:04 AM	Wave-Packet 17
9:08 AM	Wave-Packet 18
9:12 AM	Wave-Packet 19
9:16 AM	Wave-Packet 20
9:20 AM	Wave-Packet 21
9:24 AM	Wave-Packet 22
9:28 AM	Wave-Packet 23
9:32 AM	Wave-Packet 24
9:36 AM	Wave-Packet 25
9:40 AM	Wave-Packet 26
9:44 AM	Wave-Packet 27
9:48 AM	Wave-Packet 28
9:52 AM	Wave-Packet 29
9:56 AM	Wave-Packet 30
10:00 AM	Wave-Packet 31
10:04 AM	Wave-Packet 32
10:08 AM	Wave-Packet 33
10:12 AM	Wave-Packet 34
10:16 AM	Wave-Packet 35
10:20 AM	Wave-Packet 36
10:24 AM	Wave-Packet 37

NetLogo Code

```
extensions[csv]
globals[
  wave ;; the current wave
  wave-cooldown ;; the wave cooldown increment variable
  per-wave ;; number of people per wave (55)
  road-bike-closed ;; the amount of time the bike road is open
  road-run-closed ;; the amount of time the running road is open
  road-closed ;; the total time roads are open
  order ;; to set orderings
  order-so-far ;; the number or orders so far
  data-row ;; a data placeholder for data input
  data ;; the spreadsheet data
  passed ;; a counter for number of times passed
  target-swim
  target-bike
  target-running
  low-swim
  low-bike
  low-running
  time
  time2
] ;; global variables that are accessible from any scope of the program
turtles-own[
  location
  swim
  t1
  bike
  t2
  running
  ordering
  type-turtle
  around-me
  total
  division
  sex
  wave-div
  new-order
  my-pass
]



to read
  file-close-all
  file-open "datas.csv"
  set data []
  while [not file-at-end?] [
    set data-row csv:from-row file-read-line
    set data lput data-row data
  ]
end
to setup
```

```
clear-all
read
set wave 0
set wave-cooldown (wave-cooldown-max - 1)
set per-wave 55
set road-bike-closed 0
set road-run-closed 0
set road-closed 0
set order 0
set target-swim interpret "0:20:13"
set target-bike interpret "1:18:15"
set target-running interpret "0:51:42"
set low-swim interpret "0:25:10"
set low-bike interpret "1:35:56"
set low-running interpret "1:06:24"
set time interpret "3:30:00"
set time2 interpret "2:45:00"
resize-world 0 swim-length + bike-length + running-length + 2.5 0 1
ask patches with [pxcor = 0][
  set pcolor red
]
ask patches with [pxcor > 0 and pxcor < swim-length][
  set pcolor blue
]
ask patches with [pxcor > swim-length + 1 and pxcor < swim-length + bike-length + 1][
  set pcolor gray
]
ask patches with [pxcor > swim-length + bike-length + 2 and pxcor < swim-length +
    bike-length + running-length + 2][
  set pcolor green
]
ask patches with [pcolor = 0][
  set pcolor brown
]
set data-row sublist data order (order + 7 - 1)
set data-row n-of 4 data-row
set order-so-far order
crt 4[
  set location 0
  set ordering order
  set order order + 1
  set type-turtle "M PRO"
  set ycor 0.5
  set xcor 0.5
  set swim interpret (item 4 (item (ordering - order-so-far) data-row))
  set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
  set bike interpret (item 6 (item (ordering - order-so-far) data-row))
  set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
  set running interpret (item 8 (item (ordering - order-so-far) data-row))
  set sex (item 2 (item (ordering - order-so-far) data-row))
]
set data-row sublist data order (order + 49 - 1)
set data-row n-of 30 data-row
set order-so-far order
```

```
  crt 30[
    set location 0
    set ordering order
    set order order + 1
    set type-turtle "M PRE"
    set ycor 0.5
    set xcor 0.5
    set swim interpret (item 4 (item (ordering - order-so-far) data-row))
    set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
    set bike interpret (item 6 (item (ordering - order-so-far) data-row))
    set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
    set running interpret (item 8 (item (ordering - order-so-far) data-row))
    set sex (item 2 (item (ordering - order-so-far) data-row))
  ]
  set data-row sublist data order (order + 2147 - 1)
  set data-row n-of 1330 data-row
  set order-so-far order
  crt 1330[
    set location 0
    set ordering order
    set order order + 1
    set type-turtle "M OPEN"
    set ycor 0.5
    set xcor 0.5
    set swim interpret (item 4 (item (ordering - order-so-far) data-row))
    set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
    set bike interpret (item 6 (item (ordering - order-so-far) data-row))
    set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
    set running interpret (item 8 (item (ordering - order-so-far) data-row))
    set sex (item 2 (item (ordering - order-so-far) data-row))
  ]
  set data-row sublist data order (order + 6 - 1)
  set data-row n-of 4 data-row
  set order-so-far order
  crt 4[
    set location 0
    set ordering order
    set order order + 1
    set type-turtle "F PRO"
    set ycor 0.5
    set xcor 0.5
    set swim interpret (item 4 (item (ordering - order-so-far) data-row))
    set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
    set bike interpret (item 6 (item (ordering - order-so-far) data-row))
    set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
    set running interpret (item 8 (item (ordering - order-so-far) data-row))
    set sex (item 2 (item (ordering - order-so-far) data-row))
  ]
  set data-row sublist data order (order + 18 - 1)
  set data-row n-of 11 data-row
  set order-so-far order
  crt 11[
    set location 0
    set ordering order
```

```
    set order order + 1
    set type-turtle "F PRE"
    set ycor 0.5
    set xcor 0.5
    set swim interpret (item 4 (item (ordering - order-so-far) data-row))
    set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
    set bike interpret (item 6 (item (ordering - order-so-far) data-row))
    set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
    set running interpret (item 8 (item (ordering - order-so-far) data-row))
    set sex (item 2 (item (ordering - order-so-far) data-row))
  ]
  set data-row sublist data order (order + 898 - 1)
  set data-row n-of 556 data-row
  set order-so-far order
  crt 556[
    set location 0
    set ordering order
    set order order + 1
    set type-turtle "F OPEN"
    set ycor 0.5
    set xcor 0.5
    set swim interpret (item 4 (item (ordering - order-so-far) data-row))
    set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
    set bike interpret (item 6 (item (ordering - order-so-far) data-row))
    set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
    set running interpret (item 8 (item (ordering - order-so-far) data-row))
    set sex (item 2 (item (ordering - order-so-far) data-row))
  ]
  set data-row sublist data order (order + 60 - 1)
  set data-row n-of 37 data-row
  set order-so-far order
  crt 37[
    set location 0
    set ordering order
    set order order + 1
    set type-turtle "CLY"
    set ycor 0.5
    set xcor 0.5
    set swim interpret (item 4 (item (ordering - order-so-far) data-row))
    set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
    set bike interpret (item 6 (item (ordering - order-so-far) data-row))
    set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
    set running interpret (item 8 (item (ordering - order-so-far) data-row))
    set sex (item 2 (item (ordering - order-so-far) data-row))
  ]
  set data-row sublist data order (order + 32 - 1)
  set data-row n-of 20 data-row
  set order-so-far order
  crt 20[
    set location 0
    set ordering order
    set order order + 1
    set type-turtle "ATH"
    set ycor 0.5
```

```
    set xcor 0.5
    set swim interpret (item 4 (item (ordering - order-so-far) data-row))
    set t1 interpret (item 5 (item (ordering - order-so-far) data-row))
    set bike interpret (item 6 (item (ordering - order-so-far) data-row))
    set t2 interpret (item 7 (item (ordering - order-so-far) data-row))
    set running interpret (item 8 (item (ordering - order-so-far) data-row))
    set sex (item 2 (item (ordering - order-so-far) data-row))
  ]
  ask turtles[
    set shape "person"
  ]
  ask turtles[
    set total swim + t1 + bike + t2 + running
    if(total < interpret "2:30:00" and sex = "M")[
      set division "male black diamond"
      set color black
    ]
    if(total >= interpret "2:30:00" and total < interpret "3:00:00" and sex = "M")[
      set division "male diamond"
      set color 84
    ]
    if(total >= interpret "3:00:00" and total < interpret "3:30:00" and sex = "M")[
      set division "male gold"
      set color yellow
    ]
    if(total >= interpret "3:30:00" and sex = "M")[
      set division "male silver"
      set color 7
    ]
    if(total < interpret "2:45:00" and sex = "F")[
      set division "female black diamond"
      set color black
    ]
    if(total >= interpret "2:45:00" and total < interpret "3:15:00" and sex = "F")[
      set division "female diamond"
      set color 84
    ]
    if(total >= interpret "3:15:00" and total < interpret "3:45:00" and sex = "F")[
      set division "female gold"
      set color yellow
    ]
    if(total >= interpret "3:45:00" and sex = "F")[
      set division "female silver"
      set color 7
    ]
  ]
  set order 0
  let agent turtles with [total > time and total <= time + 45]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 1
      set order order + 1
    ]
```

```
]
set agent turtles with [total > time + 45 and total <= time + 90]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 90 and total <= time + 135]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 135 and total <= time + 180]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 180 and total <= time + 225]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 225 and total <= time + 270]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 270 and total <= time + 315]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 315 and total <= time + 360]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
```

```
    set order order + 1
  ]
]
set agent turtles with [total > time + 360 and total <= time + 405]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 405 and total <= time + 450]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 450 and total <= time + 495]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set agent turtles with [total > time + 495 and total <= time + 540]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 1
    set order order + 1
  ]
]
set order 0
set agent turtles with [total > time2 - 495 and total <= time2 - 450]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 0
    set order order + 1
  ]
]
set agent turtles with [total > time2 - 450 and total <= time2 - 405]
foreach sort-on [ordering] agent[
  ask ? [
    set ordering order
    set new-order 0
    set order order + 1
  ]
]
set agent turtles with [total > time2 - 405 and total <= time2 - 260]
foreach sort-on [ordering] agent[
```

```
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 - 360 and total <= time2 - 315]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 - 315 and total <= time2 - 270]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 - 270 and total <= time2 - 225]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 - 225 and total <= time2 - 180]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 - 180 and total <= time2 - 135]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 - 135 and total <= time2 - 90]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
```

```
    set agent turtles with [total > time2 - 90 and total <= time2 - 45]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 - 45 and total <= time2]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 0
      set order order + 1
    ]
  ]
  set order 0
  set agent turtles with [total > time2 and total <= time2 + 45]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 2
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 + 45 and total <= time2 + 90]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 2
      set order order + 1
    ]
  ]
  set agent turtles with [total > time2 + 90 and total <= time]
  foreach sort-on [ordering] agent[
    ask ? [
      set ordering order
      set new-order 2
      set order order + 1
    ]
  ]
  reset-ticks

end
to go
  if all? turtles [location = 6][
    stop
  ]
  if any? turtles with [location = 0][
    set wave-cooldown wave-cooldown + 1
    if(wave-cooldown >= wave-cooldown-max)[
      set wave-cooldown (wave-cooldown - wave-cooldown-max)
      ask turtles with [(ordering >= wave * (per-wave / 4 ) and ordering < (wave + 1) * (
          per-wave / 4) and new-order = 0) or (ordering >= wave * (per-wave / 4 ) and
```

```
          ordering < (wave + 1) * (per-wave / 4) and new-order = 1) or (ordering >= wave *
           (per-wave / 2 ) and ordering < (wave + 1) * (per-wave / 2) and new-order = 2)][
        set location 1
        if lanes? [
          if(swim < target-swim)[
            set ycor 0.75
          ]
          if(swim > low-swim)[
            set ycor 0.25
          ]
        ]
      ]
      set wave wave + 1
    ]
  ]
  if any? turtles with [location = 3][
    set road-bike-closed road-bike-closed + 1
  ]
  if any? turtles with[location = 5][
    set road-run-closed road-run-closed + 1
  ]
  if any? turtles with [location = 3 or location = 4 or location = 5] [
    set road-closed road-closed + 1
  ]
  ask turtles with [location = 5][
    facexy 2.5 + swim-length + bike-length + running-length ycor
    fd (10 / running)
    if(xcor >= 2.5 + swim-length + bike-length + running-length)[
      set location 6
      set ycor 0.5
    ]
  ]
  ask turtles with [location = 4][
    facexy 2.5 + swim-length + bike-length + running-length ycor
    fd (1 / t2)
    set shape "person"
    if(xcor >= 2.5 + swim-length + bike-length)[
      set location 5
      if lanes?[
        if(swim < target-swim)[
          set ycor 0.75
        ]
        if(swim > low-swim)[
          set ycor 0.25
        ]
      ]
    ]

  ]
  ask turtles with [location = 3][
    facexy 2.5 + swim-length + bike-length + running-length ycor
    fd (40 / bike)
    set shape "bike"
    if(xcor >= 1.5 + swim-length + bike-length)[
```

```
    set location 4
    set ycor 0.5
  ]
]
ask turtles with [location = 2][
  facexy 2.5 + swim-length + bike-length + running-length ycor
  fd (1 / t1)
  set shape "person"
  if(xcor >= 1.5 + swim-length)[
    set location 3
    if lanes? [
      if(swim < target-swim)[
        set ycor 0.75
      ]
      if(swim > low-swim)[
        set ycor 0.25
      ]
    ]
  ]
]
ask turtles with [location = 1][
  facexy 2.5 + swim-length + bike-length + running-length ycor
  fd (1.5 / swim)
  set shape "fish"
  if(xcor >= 0.5 + swim-length)[
    set location 2
    set ycor 0.5
  ]
]
ask turtles with [location = 1][
  let x xcor
  let y ycor
  let z 1.5 / swim
  set around-me count other turtles with[ycor = y and abs(xcor - x) < abs(z - (1.5 /
      swim)) and location = 1]
  set passed passed + (myfunc around-me) * count other turtles with [ycor = y and xcor
      > x and xcor < x + z - (1.5 / swim) and location = 1]
  if (group-con?) [
    set my-pass my-pass + (myfunc around-me) * count other turtles with [(ycor = y and
        xcor > x and xcor < x + z - (1.5 / swim) and location = 1) or (ycor = y and xcor
         < x and x < xcor - z + (1.5 / swim) and location = 1)]
  ]
]
ask turtles with [location = 3][
  let x xcor
  let y ycor
  let z 40 / bike
  set around-me count other turtles with[ycor = y and abs(xcor - x) < abs(z - (40 /
      bike)) and location = 3]
  set passed passed + (myfunc around-me) * count other turtles with [ycor = y and xcor
      > x and xcor < x + z - (40 / bike) and location = 3]
  if (group-con?) [
    set my-pass my-pass + (myfunc around-me) * count other turtles with [(ycor = y and
        xcor > x and xcor < x + z - (40 / bike) and location = 3) or (ycor = y and xcor
```

```
          < x and x < xcor - z + (40 / bike) and location = 3)]
    ]
  ]
  ask turtles with [location = 5][
    let x xcor
    let y ycor
    let z 10 / running
    set around-me count other turtles with[ycor = y and abs(xcor - x) < abs(z - (10 /
        running)) and location = 5]
    set passed passed + (myfunc around-me) * count other turtles with [ycor = y and xcor
        > x and xcor < x + z - (10 / running) and location = 5]
    if (group-con?) [
      set my-pass my-pass + (myfunc around-me) * count other turtles with [(ycor = y and
          xcor > x and xcor < x + z - (10 / running) and location = 5) or (ycor = y and
          xcor < x and x < xcor - z + (10 / running) and location = 5)]
    ]
  ]
  tick
end

to-report interpret [hi]
  report floor (((read-from-string (substring hi 0 1)) * 60 * 60 + (read-from-string (
      substring hi 2 4)) * 60 + (read-from-string substring hi 5 7)) / 20); 20 seconds
      per tick
end
to-report myfunc [around]
  report ln (around + 1)
end

to-report congestion [psd]
  ifelse (ticks != 0) [ report psd / ticks ] [ report 0 ]
end

to-report group-congestion [number]
  report mean [my-pass] of (turtles with [new-order = number])
end
```

References

[1] Garrett, William E., and Donald T. Kirkendall. *Exercise and Sport Science*. Philadelphia: Lippincott Williams & Wilkins, 2000. Print.

[2] Wilenski, Uri. *NetLogo*. Center for Connected Learning, n.d. Web. 13 Nov. 2016. (`https://ccl.northwestern.edu/netlogo/`).

[3] B, Hanley. "Pacing, Packing and Sex-based Differences in Olympic and IAAF World Championship Marathons." *J Sports Sci*. 34 (2016): 1675-681. *Pubmed.gov*. Web. 13 Nov. 2016.

[4] "Triathlon." *Rio 2016*. International Olympic Committee, Web. 12 Nov. 2016. (`https://www.rio2016.com/en/triathlon`).