OLYMPIC ROAD RACE

(Programming and Problem Solving, Fall 2004-Project 4)

James Rishe
Eric Goldstein
Bhagyashree Bohra

Introduction

Professional athletes of any sport typically strive to utilize everything that makes their energy expenditure more efficient and in turn makes them better performers. Specifically, team-based bike races are focused on getting from one place to another as quickly and as efficiently as possible, so clearly a professional attitude is taken in order to be successful on the road. In the technology-centric world of today, computer simulations can often lead to very realistic re-creations of actual races and in turn allows the riders to be as prepared as possible for each potential bump in the road. The setting of this programming assignment is just that: to realistically re-create a team-based biking scenario without being too computationally complex (due to time restrictions). Because of the fairly realistic parameters of this programming challenge, techniques used to succeed in real-world bike races can be applied to this "Olympic Road Race" problem as well. In the upcoming sections, the reader will see that many of our assumptions as well as our successful strategies stemmed from accurate and sometimes clever interpretations of bike races in the real world.

Base Line strategies

The energy consumption for a player in the previous second is given by the integral of $v^{2.5}$ (1-f (d)) over one second, where v is the speed and f (d) is the drafting factor.

$f(d) = (5-d)/10$ for $2 \leq d \leq 5$

The bike length is 2 m. Hence riding behind a rider will result in a 30% energy saving. The saving reduces as the distance from the rider in front reduces and there is no saving after 5m.

This suggests that there is a marked advantage to a rider by drafting behind another rider. A further mathematical analysis is presented below.

The steady state energy consumption with no drafting (assuming that the rider rides at a constant speed for the entire race) can be found as follows:

For entire race $E = v^{2.5} * t$

where t is the time required to complete the race. Substituting t = race length /v

$\therefore E = v^{2.5} *$(race length/v)

We get v = (E/race length) $^{0.667}$

This is the ideal speed in with a single rider in the race, where the rider completes the race as fast as possible and using up all his energy. This is the optimal solution to the problem in the single rider mode. The maximum speed allowed in the race is 25 m/s. hence even if we get a higher value than 25m/s; we can simply go at the maximum allowable speed and will have energy left even after completing.

However, this is not representative of the actual race conditions. We have multiple riders per team as well as multiple riders. Hence there is an opportunity for saving energy by drafting other players. The player with more energy can move faster (unless energy is high enough to complete the entire race at full speed) and hence will win the race. So drafting plays a very important role. However this also brings into play other factors. The rider being drafted by the our rider may choose to slow down deliberately in order to

slow down our rider. On the other hand, the rider may start going faster in order to prevent our rider from taking advantage of drafting. If our rider chooses to go faster, this may result in expending more energy than we can afford and hence we won't be able to complete the race. These factors have to be taken in consideration before defining the drafting strategy

I Ride a Huffy

Motivation stemmed from a desire for this player to get from the start to the finish as efficiently and with as little detrimental movement as possible. Drafting was used as much as possible in order to minimize energy expenditure, thus increasing its maximum attainable steady-state speed.

Its performance and characteristics can be subdivided into two categories: (1) the initial startup phase and (2) the steady state phase.

The initial startup phase lasts from the time t=0 until all players have converged into a perfectly straight line, one rider in front of another. After numerous convergence strategies were discussed both within our group and in class, an average-position strategy was implemented because it allowed for convergence in the fewest number of turns. The convergence algorithm starts by retrieving the positions of all players on the current team from the simulator by storing the integer index returned from the function call `_olympics.indexOf(this)` then stepping through the team's rider array and finding the position of each rider in a for loop by calling

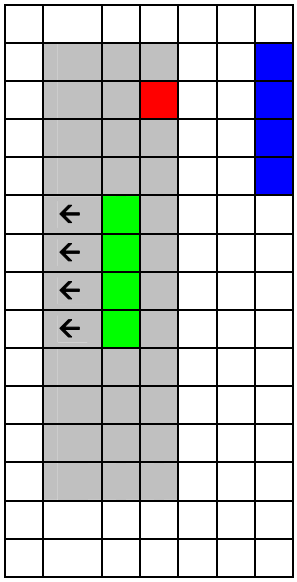
```
info[_olympics.indexOf(this)][i].lane()
```

for each array index. Then, the mean of those positions was computed, and the players then converged into the lane equal to the average lane position of all riders on the current team.

Once all riders had converged into a single line (no more than two meters apart from one another in order to gain the maximum benefit from drafting), the steady state strategy was utilized. This name is slightly misleading because the initial portion of time when the riders are in the steady state phase involves them increasing speed until they reach an optimal, constant speed. However, because riding at a steady speed occurs for a vast majority of this portion of the race, the nomenclature is appropriate. As previously alluded to, the initial portion of the steady state phase is devoted to adjusting speed so that the riders will reach the finish line with as little energy remaining as possible. Each player in the line is classified as one of three types: (1) the leader, (2) the ace, and (3) the middle of the pack. The leader is the foremost rider, the ace rider rides at the end of the line to get the maximum benefit from drafting, and the middle of the pack riders are all those in between the leader and the ace. The positions adjust dynamically during the race as players "die" (run out of energy). At the start of each turn (i.e. discrete increment of time), the ace determines the optimal speed at which it can travel in order to finish the race with as little energy as possible (and thus reach the finish line as quickly as possible). Recalculating this every turn is beneficial in case another team's player or players attempt to re-route us or slow us down because we can move away from them

then can dynamically recalculate the optimal speed then match that speed without having to move our player to a recovery strategy (the recovery is inherent when using this implementation).

Avoiding all parasitic players altogether can eliminate any chances of slowing down, so this is precisely the implemented strategy of disaster avoidance. Specifically, each rider calculates whether there is a rider from another team within a certain distance from it. The area within which each rider looks for competitors is anywhere equal to or less than 25 meters from them as well as either in the same lane or in an adjacent lane.



Graphical depiction of disaster avoidance strategy.
"I Ride a Huffy" players are green.
The grey area is the region in which green looks for competing players to move away from.
Red is too close so green will move away to avoid parasitic or detrimental actions.
Blue is a safe distance away.

As shown in the figure above, our drafting line avoids all competing riders such that they are rarely (often never) adjacent to us. Moreover, if a single rider detects that there is a competing player – parasitic, destructive, or otherwise – within the 3 lane-by-25 meter area around it, it will move away to avoid any unfavorable actions taken against it. If the competitor ends up in the same lane as us, a uniform random variable is drawn with value in the range [0,1); if its value is less than 0.5, the team collectively goes left in the same move, otherwise, if greater than or equal to 0.5, the team similarly goes right. This may lead to crossing the track back-and-forth if the competing player is persistent, but most likely, because the number of parasitic and/or destructive players per team is usually less than the total number of players on that team, the followers will die out before we do, and with less riders riding there are less people that can win, thus improving our chances of securing a medal. Because the initial intent of this player was to make as few moves as possible, if there are no competing riders in the 3-by-25 area around each of our players, the group will continue in a straight line indefinitely. However, the possibility of criss-crossing the board contradicts this goal, so recall that the primary objective is to finish the race in as little time as possible, so avoiding parasitic and destructive players before they are adjacent to the drafting line allows us to accomplish that most effectively.

Bhagyashree's Player (Player4BB)

The motivation for this player was to have three players win the race by going at the best possible speeds. This required a deeper analysis of the dynamics of the race. The drafting advantage is a desirable strategy. This is because the saving in energy will allow the rider to go at a higher speed later than if he doesn't draft. Hence, the player using the initial line formation strategy described in our baseline strategies.

Once the riders lined up, the steady state energy calculations were used to guide the accelerations of the player. Considering cases for which the number of riders is four or higher, we guide the accelerations based on the energy of the last but third rider and a 'fudge factor'. The fudge factor compensated for any approximations and to account for the fact that speed wasn't constant and some accelerations and decelerations had to take place. It was chosen based on trail and testing in various settings. Initially, while all the players are getting in line, the energies of all the players are not similar, depending on their initial lanes and how soon they start drafting. However, assuming that initial setup time is small compared to the entire race, we assume that these energy differences would not be large, hence as the race has progressed a lot, we expect the last and first rider to have the maximum and minimum energy respectively.

There are some problems that arise while following this single line strategy. Disruptive blocking behavior of the other teams has to taken into account. There is a possibility of being deliberately blocked by another rider and hence causing our own line to slow down. Another problem is having a player from another team draft our line. This behavior may not seem destructive at a first glance. But a further examination reveals that the other rider would end saving more energy and hence could potentially go at a faster speed later and could end up defeating us.

A solution to both these problems was to keep changing lanes as illustrated below. Once the players have aligned, they change lane simultaneously at every turn, if there are opening for all. The team ensures that the line is not broken and the riders change lanes only if it is possible for all of them to change lanes. In case (a), they would change the lane. However, in case (b) they wouldn't change lanes since the line would break. Line breaking is not done since it is harder to reform the line and valuable energy is wasted by a loss of draft. This is a good decision since riders at our side do not affect us. The behavior we want to prevent is blocking and parasites.

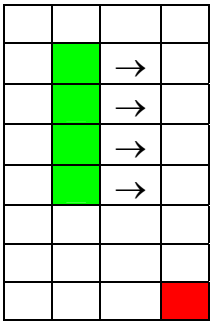


Fig (a)

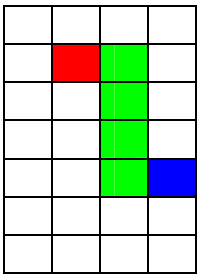


Fig (b)

The lane change at each turn is decided randomly as left, right or no change with an equal probability. Hence the chance of being blocked and slowed reduces, since there is now only a one third probability at every turn that the blocker will be able to make the same lane change as our riders. Similarly if a player is following our line, the probability that the player can continue to draft us reduces by one third at every turn. The drawback of this preventive strategy is evident is a very crowded area. Since the restriction imposed all players take the same action, in a crowded area, it may be possible that there are no open spaces in the adjacent positions of all the players. Hence the lane won't be changed for multiple and we could successfully get blocked or drafted by opponents.

The final game strategy implemented was having the last two players break out of the line towards the end of the race and ride to the finish line at maximum possible speeds. The speed is already optimized to have last but second player finish the race. Hence the last but one and last riders, who have more energy, can complete the race at higher speeds and hence need to break out of the line. The decision to break out was implemented in two ways. If the last rider or the last but one rider has enough energy to complete the race with the maximum allowable speed, they would simply break out of the line and complete the race. If this does not occur, for the last 3% of the race, the last two riders break out and complete the race at maximum possible speed. This number was chosen based on trail and testing. We also ensure that if both riders break out, they go into different lanes and don't end up blocking each other. The same problems of being blocked and drafted exist for the riders that have broken out. The same strategy of lane weaving described earlier is employed to reduce these harmful behaviors.

This player also works in the same fashion for fewer than four players, the only difference being the energies are computed based on the last but one riders and last riders respectively instead. In the single rider mode, it is simply a matter of steady state calculations.

One of the pitfalls was that escaping blocking and parasitic behavior in crowded race was less effective. The tournament results are also indicative of this and are discussed in further detail below. The breakout point could also have been chosen in a more effective way, with a deeper analysis and testing, given more time. The player also did not have any parasitic behavior. We thought that having a player based mainly on parasitic

behavior would not perform as well. We predicted that the energy benefits could be outweighed by the risk of being slowed down or going too fast. This strategy also seemed less attractive, based on the fact that it was not very hard to get rid of a parasite and class discussions indicated most groups planned to implement such counteractions. However, the player would have benefited by implemented some partial parasitic behavior. An example would be drafting when possible (and not aggressively) to draft would have resulted in some energy saving and as result, better speed and possibly more wins.

Compassion Cruiser

This player focused on the premise of getting all the riders into a straight line, and having all of the riders finish in as little time as possible. Knowing how it actually feels to be in road race and feel exhausted, we wanted to design a "compassionate" player that allowed all of its riders to survive to the finish line, but still finish in a decent time.

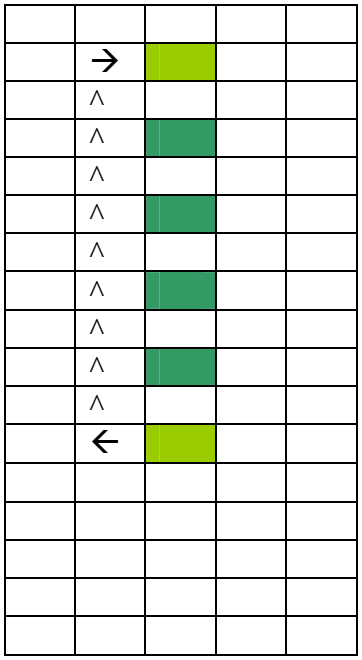
The players began by lining up in the lane that was the middle spot between the leftmost and rightmost riders. To avoid having our own riders obstructing each other while lining up, we had the ones who had already reached the destination lane accelerate at a slightly higher rate than the others until they were all in line.

Once the riders were in line, we had them determine a target speed by the formula

$$speed = (E/D)^{2/3}$$

where E is the player's remaining energy, and D is the remaining distance to the finish line. This speed is theoretically the speed that would have a single unobstructed rider going at constant speed all the way to the finish line, where it would use the last of its energy.

Having the riders go along this way would clearly put all of the burden on the rider in front, so every thirty rounds, we check for an opponent rider directly to the left or right of our line, and if one side is unobstructed, the lead rider moves aside and drops back to the rear of the line. This places a fresher rider at the front of the line, and gives the old leader the benefit of a slipstream. The desired speed is recalculated every round, so the extra energy of the new leader allows the team to increase its speed.



Consideration is also taken into account as far as undesired interference from other players. Every round, the team checks for opponents within 3m of the front or last rider, or anyone in between, and if there is anyone, the team checks to make sure either the left or right is completely unobstructed, and moves in a clear direction.

Even avoiding exhaustion for its riders, this player showed relative promise. In the in-class, simulations, it was usually fairly close behind the "optimal strategy" players, and was in one of the first two groups behind the optimal strategy players. The obvious difference between this one and those optimal players is that all of its riders were still active at a point of the track where some of the optimal team's riders had collapsed from exhaustion.

This player also performed fairly well in our own simulations under a wide variety of conditions, but even though it had a higher chance of having all of its riders surviving to the finish line, it performed slightly slower in most cases than Player4BB, so we submitted that one and not Compassionate Cruiser.

Above: Diagram of the leader-cycle process that takes place every thirty rounds, provided a leader is not currently in the process of cycling to the back.

Tournament Results and Analysis:

Crowded Board: (Teams in race =10)

'I ride a huffy' finished fourth in the base tournament and in the top half for the base tournament with 7 riders.

'Player4BB' didn't do so well in the crowded board.

Sparse Board:

In the case of a single team mode time trail), our players did pretty well.'Player4BB' finished third and 'I ride a Huffy' finished in the top half.

With two teams, the performance of the players was even better, with 'Player4BB' and 'I ride a Huffy' finishing second and third respectively.

As the number of teams increased to four, the performance dropped, but 'I ride a Huffy' still finished in the top half.

Base configuration with multiple dumb players:

Our players performed the best in this configurations, with 'Player4BB' finishing first and 'I ride a Huffy' finishing third.

Two identical players:

'Player4BB' finished third in this configuration.

The results indicate that our players were less effective in a crowded board and did very well in sparser board. The strategies for counteracting harmful behavior were not as effective in a crowded board for 'Player4BB' as alluded to earlier. For 'I ride a Huffy', a more thorough mathematical model for the speed may have resulted in an improvement in the performance.

However, in a sparse board our players did pretty well, indicating that our strategies were quite effective.

With DumbPlayers, our players gave a very good performance as well. This indicates that they were very good in handling randomized interferences. They had the first and third highest average scores against DumbPlayer.

With two instances of itself, Player4BB did quite well. The reason for this would be the fact that no parasitic strategies were implemented and hence the interference was more of a random nature. The player had strategies to avoid disruptive behavior, which worked well when interference was of a random nature.

Conclusion

This problem gave us a lot of insight into how one might use a simulator to strategize an Olympic bicycle race, and try to discover which strategies work better than others. We had three players, each operating to varying degrees of attempts to keep the players in good condition: one which aimed to keep them all alive, one which aimed to keep three alive, and one which aimed to keep just one alive. We understood that our player had little chance of beating the players with the "optimum strategy" but we wanted to try some other approaches, believing from experience in the in-class simulations that we had a good chance of defeating several of the other players when the optimal strategy players were out of the way, and we believe that the tournament results reflect that prediction.

Work

All three team members contributed to the base player, which served as the basis for the three end players.

Eric did most of the modifications on the base player that went into the creation of "I Ride a Huffy"

James did most of the modifications on the base player that went into the creation of "Compassion Cruiser"

Bhagyashree did most of the modifications on the base player that went into the creation of "Player4BB"

This report was the work of all three team members.