Project 4: Olympic Road Race

Group 5
John Cieslewicz (johnc@cs.columbia.edu)
Ratul Malli (rm2236@columbia.edu)
Chris Murphy (cdm6@columbia.edu)
Lawrence Wang (lcw2001@columbia.edu)

December 13, 2004

Introduction

This project simulates a long bicycle race, similar to those run in the Olympic Games. Though medals are presented to the riders who finish first, second, and third (i.e., individual accomplishment is awarded), riders are part of a team, so teamwork is possible in order to maximize performance. Unlike a real bicycle race, however, in which factors like stamina, motivation, and fortune all figure into the outcome, this project puts all riders on equal footing, so that success should be determined by strategy alone.

Each rider on the team starts with some level of energy (we'll call it E), and energy is consumed by the rider as he goes through the race. The incremental consumption of energy is $v^{2.5}$, where v is the rider's velocity. When the rider runs out of energy, he can no longer continue the race. However, riders can conserve energy by "slipstreaming" another rider, i.e. by riding just behind another rider so that he does not have to encounter as much wind resistance. The amount of energy saved by "drafting" another rider is a function of distance behind the other rider.

For this project, each development team needs to create a "coach" that is fully aware of the position, lane number, velocity, and energy levels of all riders in the race (not just the riders on that particular team). Each second of the race, the coach tells the riders on the team what to do (in terms of changing lanes and accelerating), and the race is over when all three medals have been awarded.

This document starts off by explaining our initial thoughts about this problem, based on our discussions as a group and in class. It then lists some of the approaches and strategies that we considered and prototyped, before discussing the implementation that we ultimately settled on. Finally, this report summarizes our results in the various tournaments that were performed, and finishes with some closing thoughts and ideas on how we could have improved.

Background and Initial Findings

Based on our initial discussions as a group and in class, we felt the following were the most important ideas to use in this project:

Forming a line and drafting

From the very beginning, it was quite clear that any given rider is better off by staying behind another rider, because of the savings in energy consumption. As the race goes on, a rider who is not facing wind resistance is going to have a very large surplus of energy compared to a rider who is not drafting at all; this difference in energy could be advantageous as riders sprint to the finish at the end of the race. Any team with a goal of putting a rider in first place would almost certainly need to use this strategy. The initial idea expressed by many groups was to put all riders in a straight line, with each rider as close as possible to the one in front of him. By riding at a certain so-called “optimal velocity”, the riders at the front of the line would expire (run out of energy) one by one until the very last rider crossed the finish line just before his energy ran out.

As a group, we agreed with this strategy wholeheartedly, seeing that it would be very difficult to win a race if the rider were not taking advantage of drafting. However, we realized over time that keeping all of our riders in the *same* line was not always necessary, as described in the next section.

Calculating optimal velocity

The optimal velocity, i.e. the fastest velocity at which a rider can go and still finish the race, is a function of the rider’s initial energy and the distance of the race. Calculating the optimal velocity is rather straightforward in the case when the rider is riding by himself (i.e., not drafting another rider), but more complicated when considering the effects of slipstreaming. Though some groups appeared to resist the idea of performing the complex math, it seemed after a while that the groups that did this calculation were the groups that tended to perform better in the in-class simulations.

We decided that calculating the optimal velocity would be of great benefit, because any other strategy would still rely on knowing this velocity. The calculations are described in the next section.

Working with other groups

There was much discussion in class about whether it would be beneficial to work with another group during the race, so that each team could benefit from each other, and somehow “trade off” finishing results so that one team did better than the other only half the time. This sort of cooperation could be implemented via such strategies as drafting in a very long line of riders, or prohibiting other groups from getting ahead of the riders. However, it seemed that, for whatever reason, many groups were hesitant to work with others, and this explicit cooperation was not widely accepted as an ideal solution.

Like everyone else (as far as we know), we decided not to work explicitly with another group, but we did try to *take advantage* of other groups by drafting them and benefiting from their strategies. This is described in the “Follow the leader” section below.

Working against other groups

Because the goal is to “win” the race, and not to “not lose” the race, it seems that all of a group’s focus should be on finishing as fast as possible, and not on distracting or interfering with other riders. Strategies were discussed such as blocking/surrounding other riders, slowing down in front of them, or constantly moving. Many of these strategies seemed to require more investment than what was actually gained: for instance, it might take two riders to block one rider from the opponent’s team. At that, focusing on just one other rider in a race with dozens of riders does little to help one’s performance overall.

Therefore, we did not attempt to implement any specific strategies for interfering with or harming other riders, though we did consider ways to prevent other teams from interfering with us.

Strategies and Concepts

This section describes some of the concepts and ideas that we experimented with and discussed past the initial stages. Though some of them were ultimately thrown out, we feel that a future project may want to explore some of these further.

Calculation of optimal velocity

As discussed in class, there may be no "optimal" strategy in the true sense of the word, but mathematically there is a maximum velocity at which riders can travel and still finish the race. The calculations we used are as follows:

Assume a team has riders in a line with energies E_1 through E_n, a distance D from the last rider to the finish, and the use of energy per second is $v^{2.5}$ to move and $0.7v^{2.5}$ if the rider is drafting another player. We're looking for the value of v such that the last rider dies right after crossing the finish line.

The optimal single-player strategy runs all the players into the ground: the front-most player dies first, then the next, and so on. So we thought about the problem by breaking the total distance into segments at the points where each rider dies. The following example is for four riders, but the solution scales to n.

$$D = vt1 + vt2 + vt3 + vt4$$

Here, t1 through t4 are the times spent completing each segment. Each time equals the energy spent in that segment (s1 through s4) by the front-most player at the time, divided by $v^{2.5}$.

$$D = v/v^{2.5} (s1 + s2 + s3 + s4)$$
$$v = ((s1 + s2 + s3 + s4)/D)^{2/3}$$

Now, to solve for v, we need those energy expenses.

After the first rider dies, you know he's run out of all his energy, E_1. That's the energy consumption for the first segment. All other riders use $0.7\ E_1$ during this segment.

During the second segment, between the death of the first rider and the death of the second, we know the second rider expends all of its remaining energy, namely $E_2 - 0.7\ E_1$. All other riders use $0.7(E_2 - 0.7\ E_1)$ during this segment.

During the third segment, the third rider expends all of its remaining energy, namely $E_3 - 0.7(E_2 - 0.7\ E_1) - 0.7\ E_1$, or $E_3 - 0.7\ E_2 - 0.21\ E_1$. The fourth rider uses 0.7 times that.

Similarly, during the fourth and last segment, the fourth rider expends its total energy minus the energy used in each of the preceding segments: $E_4 - 0.7(E_3 - 0.7(E_2 - 0.7E_1) - 0.7E_1)$, or $E_4 - 0.7\ E_3 - 0.21E_2 - 0.063E_1$.

Adding the energy each front-most rider uses in its segment gives us the sum:

$E_4 + 0.3\ E_3 + 0.09E_2 + 0.027E_1$

The general form is $\mathrm{SUM}(0.3^{(n-i)} * E_i, i=1 \text{ to } n)$, where n is the number of riders.

Note that since each rider in front of the last gets another power of 0.3 in its coefficient, it contributes roughly one order of magnitude less energy to the total, so you have a ceiling on the maximum useful number of riders for this strategy. That maximum is the number of riders that lets the last rider draft for the whole race, where $v = (E/0.7D)^{2/3}$. We assume that that should be the limit of the general formula as the number of players increases to infinity.

Follow the leader

In the previous section, we described the idea of getting in line and drafting another player, pointing out that a player who is behind another uses less energy while traveling at the same velocity, and thus could go faster at the end of the race. To take this idea to its fullest extent would mean to stay behind another player for as long as possible, and to use the additional energy to speed up and overtake the rider at the very last instant, thereby winning the race. From this idea came the strategy of “follow the leader”.

The strategy, as was said in class, “takes advantage of other riders’ strengths *and* of other riders’ weaknesses.” It is clear how it takes advantage of the strengths: if another team is traveling at an optimal velocity, or has a very strong strategy for winning the race, then following them at the same velocity and mimicking their actions from behind allows our player to have the same benefits, but also to have extra energy as a benefit of drafting. Additionally, acting as a “parasite” can feed off another team’s weakness: if the team has slightly miscalculated the optimal velocity, or has another error in its strategy, then our rider would be able to follow that team as far as possible, but would not be susceptible to the same error at the end of the race because of the energy saved.

This strategy was based on assumptions of how other teams would act: specifically, that they would ride in a line at the optimal velocity, that they would not change lanes too frequently, and that they might make small errors in calculation of how much energy would be required to finish the race. However, as our implementation of this strategy carried on, and we began discussing it in class, it became clear that its usefulness in this project could only last so long. Other teams started to come up with ways to avoid being followed (specifically, by moving around a lot), and the calculations and estimations of velocities and energy levels became more precise. Therefore, a rider implementing only this strategy was increasingly unlikely to defeat another team with a stronger strategy.

A pessimist might say that “following the leader” was a quick and dirty way to try to win the race with as little effort as possible, but an optimist might say that by implementing this strategy and warning other teams about it, we spurred the class on to develop better strategies and better calculations.

Follow one player

One difficulty that could be encountered in most strategies based on drafting, especially the “follow the leader” approach, is that it is difficult to always keep all riders in a single line. Any number of situations could cause that line to break, even when the players are programmed to all move together. For instance, if the riders on a team are implementing a “follow the leader” strategy and the leader changes lanes, then the riders should also attempt to change lanes to stay behind the leader. However, it may be the case that some riders are prohibited from changing lanes because other riders are obstructing them. What should be done in that case? Should the riders stay in line but not change lanes? That may allow the leader to get too far away, and the riders would not benefit from slipstreaming the leader.

Alternatively, the riders could break the line and try to follow the leader independently. Though they would suffer some cost of absorbing wind resistance while catching the moving leader, this could be more beneficial to the team overall, in that they would still (eventually) be following the leader. We started to develop this approach as an alternative to the pure “follow the leader” strategy, which always tried to keep the riders in a line. In our early development, we did see some improvement, since the riders were more likely to get behind the leader if they could move on their own.

As we developed this idea, we realized that it could be extended to a more general case, in which only one rider is “intelligent” (i.e., is implementing any sort of strategy) and the second player simply follows and mimics him. The third player follows the second, the fourth follows the third, and so on. This was the idea we labeled “follow one player”. It is much easier to implement and yet the results should be identical.

For instance, consider the case in which the four riders of a team (call them A, B, C, and D) are implementing the “follow the leader” strategy.

- In our original approach, we would try to keep them all in a line; if the leader moved, and one of the four players was blocked from moving, then they would not move at all.
- In the second approach, each player would move when it was able to, and would recalculate the position of the race leader and would try to move behind it.
- In the “follow one player” approach, however, only one rider (for instance, rider A) would calculate the position of the race leader and try to move behind it. Rider B would follow rider A, rider C follows rider B, and rider D follows rider C.

Though the last two approaches seem identical, the “follow one player” approach requires less calculation (since we always know where player A is, though we don’t always know where the race leader is). More importantly, if player A changes strategy (like trying to overtake the leader at the end of the race), then the other players do not have to explicitly have this strategy coded as well; they just essentially follow player A.

Handling obstructions

Regardless of the strategy that players implement, they will always need to be able to handle obstructions caused by other riders. This could include:

- riders from another team slowing down in front of you
- riders from another team stopping you from changing lanes
- riders interfering with the initial line-forming (merging) at the beginning of the race
- riders from another team being between you and the leader

In many cases, the pure “follow one player” approach described above could be a hindrance. Only the first player would figure out how to deal with the obstructions; the other players would try to follow it, but would not have the "intelligence" to deal with the conflict. For instance, if the players (A, B, C, and D) are following the leader, and a slow-moving rider is between them and the leader, rider A might move to the right to try to pass it. In the time that it takes for B to realize that A has changed lanes, however, the obstructing player might slow down and also move right, preventing B from moving right and getting behind A. Thus, riders C and D would be waiting for rider B to move, when they could potentially get behind A on their own. In the pure "follow one player" approach, riders B, C, and D would not think to speed up to pass the obstructing rider, or to slow down and draft it for a while; they would stubbornly continue to try to move right. This would cause one of them to take on an additional energy loss, and would weaken the overall team performance.

To avoid these situations, the "follow one player" approach would need to be modified so that the "dumb" players can be more "intelligent" when they are unable to follow the one player. This would need to entail acceleration and lane changes, so that they can quickly get behind the rider they are supposed to be following.

Determining the leader

Essential to the "follow the leader" strategy is determining which rider is actually the leader of the race. Typically it is impossible to get *right behind* the race leader, because the leader's teammates will almost certainly be in a close line behind him. Similarly, trying to follow the rider that is leading the race at any given point is not always the best strategy, because the leader may (intentionally or otherwise) be going at a rate that is not sustainable for the whole race, causing the riders using the "follow the leader" strategy to burn out too quickly.

One approach to this is to look at each opposing team and find the rider that is furthest back on each team. Then, amongst all of those riders, find the rider who is furthest ahead. The assumption is that the lead team will almost certainly be riding in a line, and the one at the back of that line is the rider that we should get behind. In the event that the rider at the back of the line is already being tailed by another team/player implementing the same strategy, we could look at other teams and look at the positions of their furthest-back riders. The assumption here is that there is likely to be more than one team running a race using the "optimal speed", and being behind any of them is sufficient.

Winning the race

As the race comes to its end, it is obvious that the "follow the leader" strategy must be abandoned, and that the team needs to make use of its extra energy and outrun the leaders to the finish line. In our initial implementations, we waited until the race was 99.8% finished in terms of distance, almost waiting until the very end to try to pass the leader. This number was mostly based on trial-and-error tests we did against other teams. Waiting too long introduced the risk of not finding an open lane in which to pass, and trying to pass too early meant that energy would be used up fighting wind resistance.

We were excited (and somewhat surprised) by how much success we were having with this strategy. In fact, we were often winning all three medals if we waited until the very end to try to overtake the leaders. We soon realized that much of our success in these races was based on slight miscalculations that other teams were making in regards to the amount of energy they would have at the end of the race. Some other teams were running out of energy literally meters from the finish line, and so we were not so much “passing” them as we were “outlasting” them.

As a result, we changed this strategy so that the “follow the leader” players would try to pass those leaders after 95% of the race had finished. At that time, they would calculate their optimal velocity (so as to go as fast as possible and still have one player finish the race) and find an open lane to ride in. The hope here was that the energy saved by drafting the leaders would translate into a slightly faster speed at the end of the race, and 5% of the distance should give the players enough time to overtake the leaders.

Implementation

This section explains our algorithms for implementing the major features of the player that we submitted.

Group5K2: Leapfrog

Though over the course of this project we implemented a number of players, each with slightly varying strategies, the one that we have submitted is a hybrid of all of the approaches described above. The Leapfrog player uses the "follow the leader" strategy for its general approach, the "follow one player" methodology for determining lane changes, and the calculation of the "optimal velocity" for determining acceleration. We feel that these three ideas are each strong on their own, and can be used together to create a very strong player.

Race Phase 1

Key to the "follow one player" approach is to figure out which of our riders should lead, which rider should follow it, which rider should follow that rider, etc. This is done on the very first turn of the race. We use a simple calculation in which the middle lane (amongst the riders) is chosen as a meeting point, the rider closest to it becomes the leader, the rider second-closest to it follows the leader, the rider third-closest to it follows the rider second-closest to it, and so on. This simplifies the problem of getting the riders into a single line, based on the assumption that the leader will be the first one to arrive in that lane, and the others will get there (in the desired order) after him.

In order to avoid some of the problems with the beginning phase of the race, in which other teams are trying to get in line and others are starting out at a fast speed, for the first 5% of the distance of the race, the Leapfrog riders simply stay in one line, traveling at the optimal velocity for finishing. Assuming that other teams implementing the optimal velocity strategy are doing the same, there is nothing lost from doing this for only 5% of the race, but it does avoid problems with startup conditions.

Race Phase 2

For the next 90% of the race, the riders implement the "follow the leader" strategy. The calculation of the rider to consider as the leader is described in the previous section: we don't actually look for the race leader, but rather the farthest-ahead rider of all riders who are farthest-behind in their own teams (assuming that the actual leader is riding in a line with his teammates). In addition, we check three things about the rider we want to follow: the rider isn't moving too fast, the rider isn't moving too slow, and the rider isn't too far ahead.

To calculate the upper bound of the speed of a rider we would consider the leader, we look at the rider in the last position on our own team, and calculate that speed as
$(E / 0.7d)^{2/3}$, where E is the rider's energy and d is the distance remaining in the race. The reason for using this is that, at any higher speed, the last rider on the team would run out of energy and be unable to finish the race.

To calculate the lower bound, we simply look at the optimal velocity for our own team, and don't consider any rider that is traveling slower than that. The reason, of course, is that at such a low speed, we could simply travel without following the leader and still win the race.

We also look to see whether the leader is too far away. This prevents us from chasing a “rabbit” who starts the race very quickly with the intention of drawing out other teams and forcing them to use too much energy. We only consider riders who could be caught in 50 seconds, assuming that we are moving at a speed that is 1m/s faster than they are.

After calculating the rider that we want to follow, the leader of our group accelerates and changes lanes as necessary to catch up with that rider. The other riders of our group follow the one in front of them, accelerating and changing lanes if they are obstructed from getting to their desired position. In the case that our riders are being blocked by another rider or another team, either intentionally or accidentally, the lead rider will move to get around those riders, and the other riders will follow suit.

In the situation in which our team finds itself in the lead (which does not happen very often but certainly could), we merely calculate the optimal velocity every 100 rounds, and follow that path until someone overtakes us.

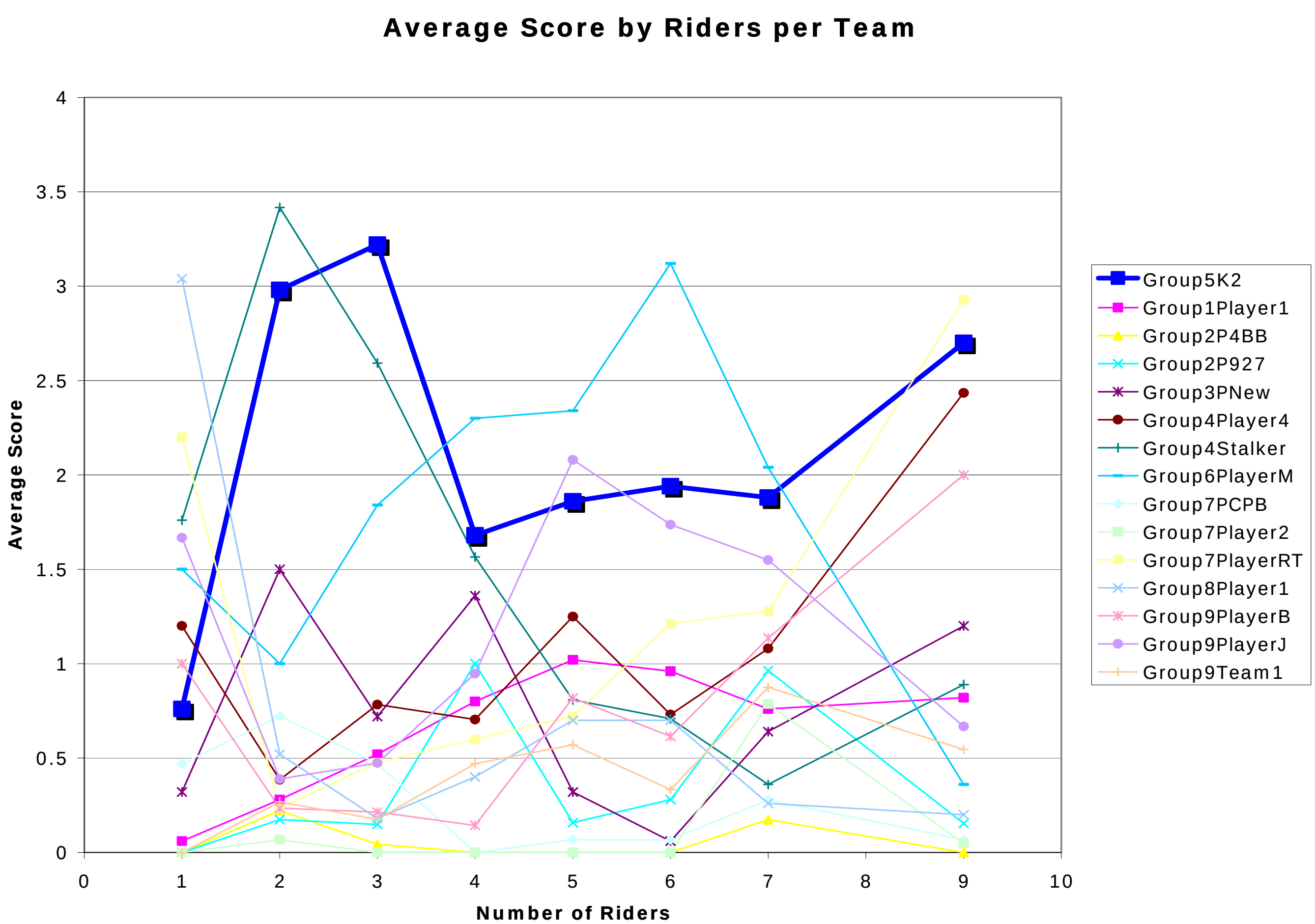
Race Phase 3

After the race is 95% over, it is time for our players to “leapfrog” the race leader and cruise to victory. As described in the previous section, the extra savings in energy (that come from having our lead rider draft another rider) should be enough to give our riders a slightly faster velocity for the last part of the race. The lead rider of our team looks for a nearby empty lane (i.e., a lane with no other team in it between our riders and the finish line), and moves to it. The other riders follow, and though not all of the riders will finish the race, the last rider should finish slightly ahead of the other teams.

Tournament Results

This section describes our player's tournament performance. We divide the analysis by the types of races run: variations on riders per team, variations on the number of teams, variations on the number of lanes, variable energy levels and race lengths, and the sanity check races against only dumb player and ourselves.

Riders Per Team



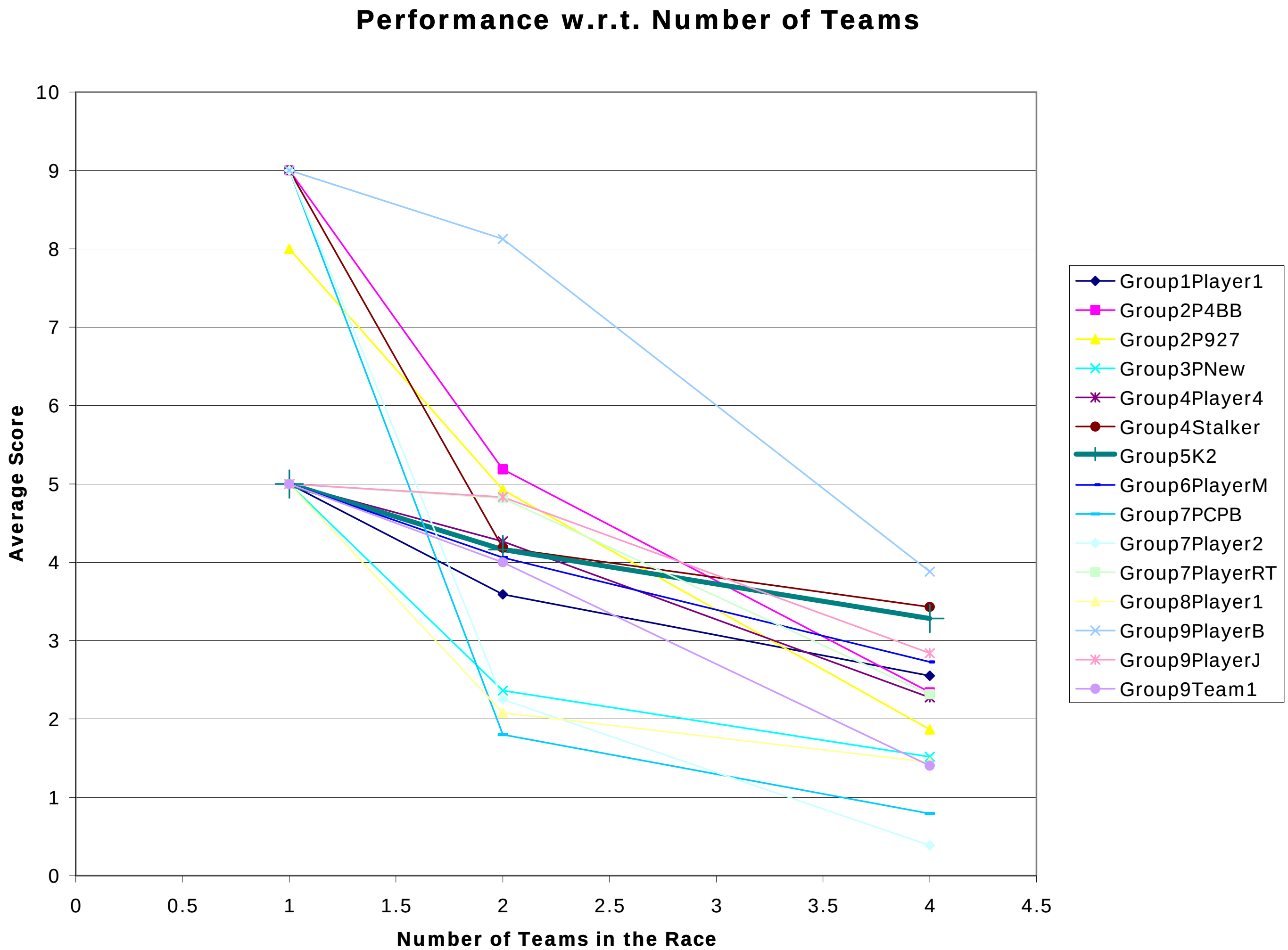
Different numbers of riders per team can create advantages as well as introduce challenges. The obvious advantage of additional riders is that there is a greater opportunity for drafting. As was discussed in class, the marginal benefit of each additional rider decreases quickly since drafting is not cumulative. The main challenge posed by different number of riders is that as the number of riders increases, strategies that attempt to maintain a line of riders must be made more robust to potential breaks in the line since each rider represents a chance that the line will break during a lane change due to a lane change conflict with another rider.

The above figure shows the average scores for each team plotted against the number of riders in the race. Our team's average score is among the top three averages for all races except those with only one rider per team. Though we have the highest average score only in the three-rider race, we finish with the second highest average score in five of the remaining seven race configurations.

As one can see from the graph, our poor performance in the single-rider race shows that our team emphasizes teamwork over individual action. Our highest average scores are when there are only a few riders per team. This is because the marginal benefit of our follow-the-leader strategy is the greatest since by following another team we are effectively gaining another player on our team.

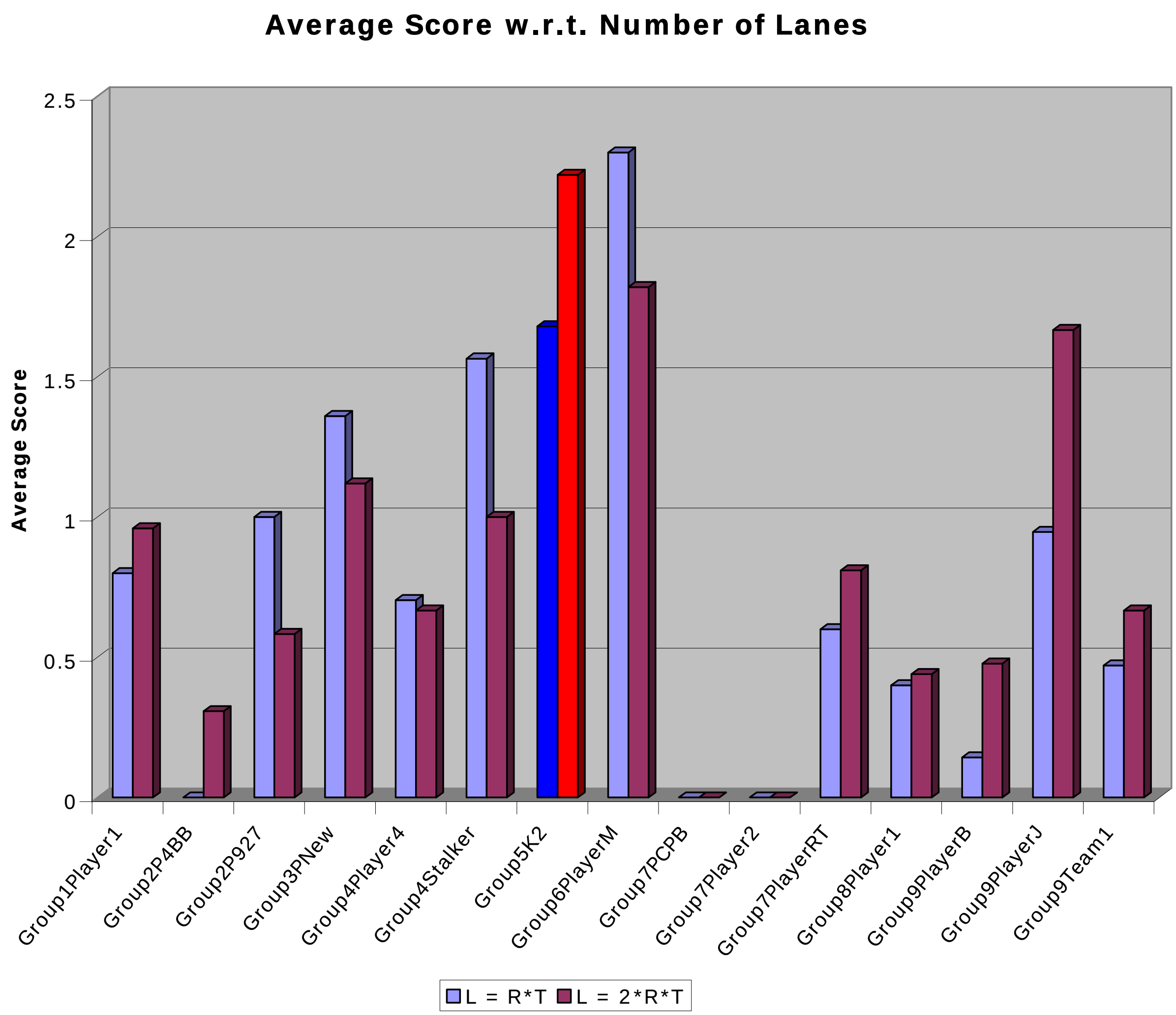
Our consistent performance in the top three teams over all races where the number of riders is greater than one shows that our strategy is robust to many different team sizes, whereas other team's performance varies greatly with the number of riders per team. Further, our good performance even with large team sizes shows that our team robustly maintained its line of riders.

Number of Teams per Race



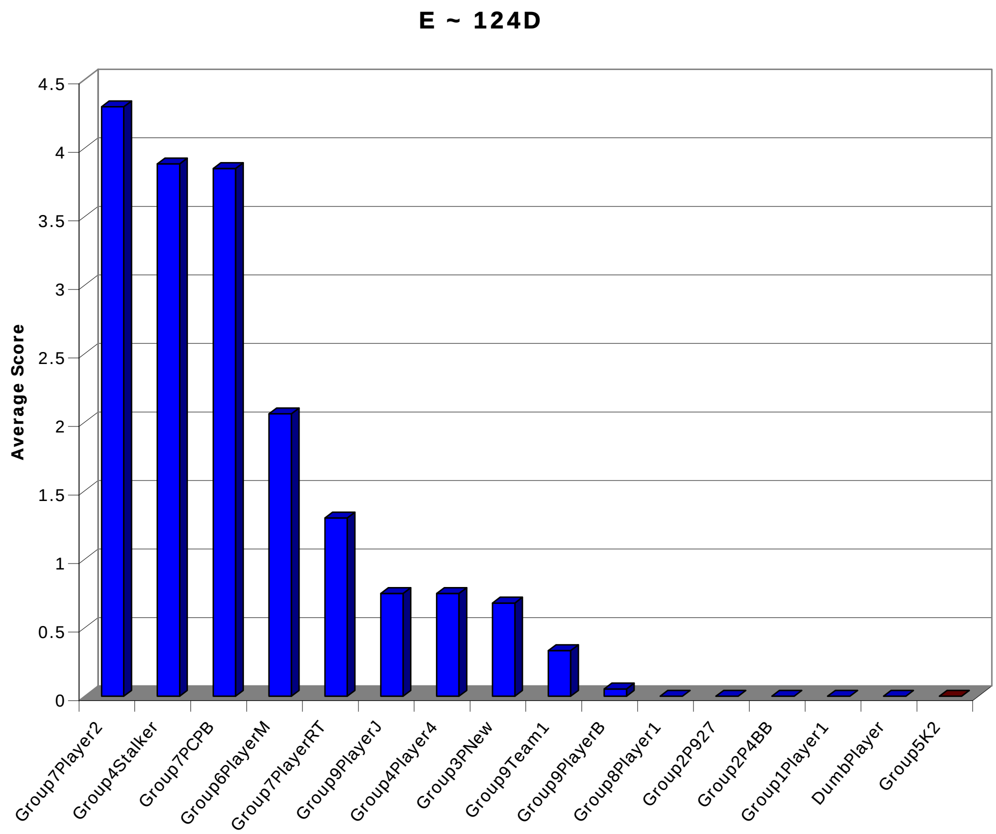
From the above chart we see that as the number of teams in the race is increased, our average score decreases. This however gives a false impression of deteriorating performance. On the contrary, our overall performance improves with more teams as though our average score decreases, our average rank actually increases. This is due to the fact that drafting plays a very important part of our race strategy and increasing the number of teams and hence the number of players increases drafting opportunities in the race and hence improves our overall performance. It is also notable that even with varying team numbers we are relatively consistent in our results and do not exhibit largely shifting scores as is seen in some other teams.

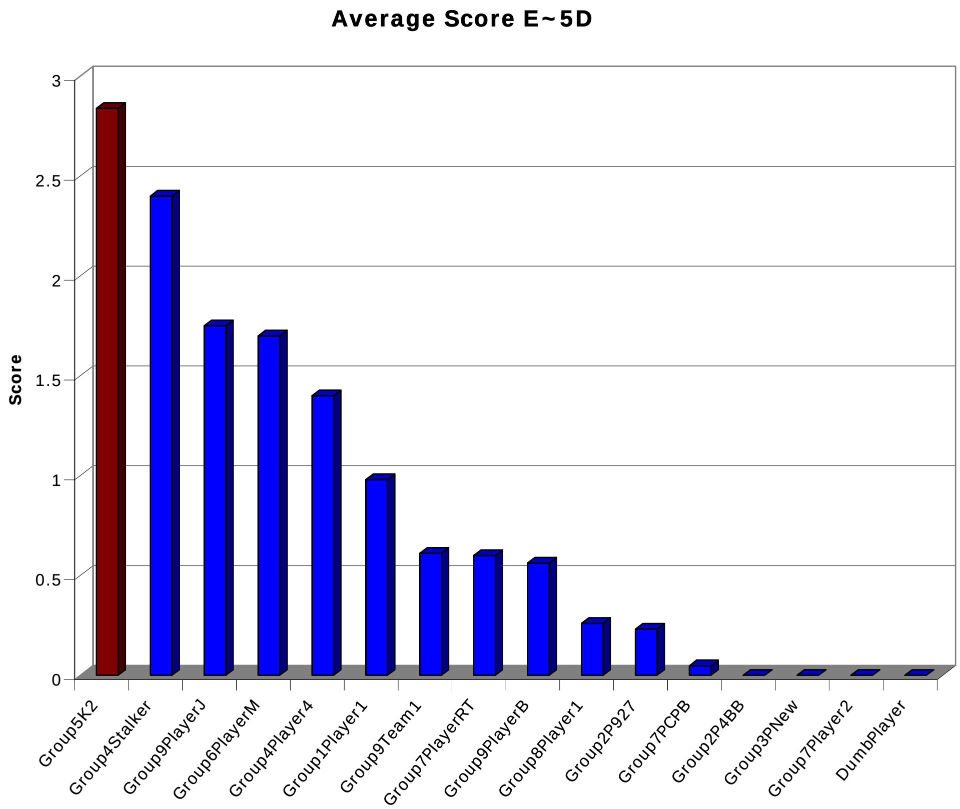
Number of Lanes



It is evident from the chart that our team performance is helped by the increase in the number of lanes. We follow a strategy of taking advantage of drafting other players and therefore rely largely on being able to form and maintain drafting groups behind other players. This is felicitated by the presence of more lanes. The increase in number of lanes reduces the congestion experienced early on in the race. It is therefore easier to get into line behind a selected rider and form lanes early on. Once in a drafting group, the availability of more empty lanes makes it easier to maintain the group for longer periods of time. There is less breaking of the group when the lead rider shifts lanes. We are also less susceptible to blocking as it is now easier to get out of a lane filled by a slow moving player and to catch up with a desired group.

Energy Level and Race Length





Two race configurations tested team performance in environments where energy was plentiful and scarce with respect to the race length. The above graphs show that our player performed relatively poorly in races where energy was proportional to 124 times the race distance, but we outperformed all of the competition in scarce races where energy was proportional to only five times the race distance.

These results are expected since we did not optimize our player for scenarios with plentiful energy. Instead, we focused on the more realistic scenario where energy is relatively scarce. The excellent performance during the scarce energy race demonstrates that our follow-the-leader, leapfrogging strategy really pays off when small amounts of energy saved make a big difference. The follow-the-leader, leapfrogging strategy provides the dual benefit of a robust line of riders, while also taking advantage of other teams by drafting them when possible. One would expect, and this race scenario confirms, that such a strategy has the greatest impact when energy is relatively scarce since this strategy seeks to maximize the energy savings while also completing the race as quickly as possible.

Special Races

Two special, “sanity check” races were included in the tournament. One consisted of numerous instances of the “dumb player” and one instance of a group’s team. The other race configuration consisted of two instances of the same team. The goal of the first race configuration was to check that the team could contend with dumb player’s erratic, sub optimal behavior. The goal of the second race configuration was to make sure that teams did not interact pathologically with their own code.

In the first race configuration our player faired well, but did not win one of the races. Though we have been unable to recreate this loss, dumb player must have randomly succeeded in fooling our player into wasting too much energy chasing it. A plurality of teams had an average score of five for this race which suggests that they planned their races such that exactly one player finished the race, sacrificing other players for the sake of speed. Four teams, however, finished with an average score greater than five, suggesting that they either plan from the outset to finish the race with multiple players alive, or may detect the condition that only their players are left in the game and then seek to maximize score over finish time. Our player is designed to follow the former strategy of finishing the race as quickly as possible with only one player surviving to the end. If not for the one fluke loss, our team would have also averaged a score of in these races.

In the second race configuration, a majority of teams faired well, winning half of all races. This is expected since with two instances of the same player, only one can win a given race. Four groups failed to win half of the games, meaning that in head to head competition with their own player neither instance finished the race. Our player performed well, with one instance of our player winning all of the races.

Conclusion and Lessons Learned

Areas for improvement

Though we are extremely proud of what we accomplished in this project, we realize that there are some aspects of our player that could be enhanced.

In class we discussed the so-called “arms race”, in which groups would have good ideas, other groups would counter them with their own ideas, and so on. We feel that we were instrumental in keeping this going, and contributed to the overall advancement of this project. However, it can be said that “follow the leader” described our development approach, as well as our player strategy. That is, because the “follow the leader” strategy depends on taking advantage of other players, we were intrinsically bound to be one step behind. Bugs could typically only be found when we tested against other players; by the time we fixed those bugs, the other players had kept progressing, so we were always trying to catch up. A future implementation that uses this sort of strategy should do some rigorous testing in advance, especially given the relatively tight deadlines of a project like this.

Accomplishments

Despite the fact that there may have been room for improvement, we consider this project to be successful based on the following:

- We **pioneered a new idea**, “follow the leader”, that combined a goal-oriented strategy (of winning the race) with a more deterministic strategy (of traveling at the optimal velocity). This concept was flexible enough to handle situations in which ours was the only team in the race, or when there were many other teams competing against us.
- By having a strategy that put other teams on the defensive, we **contributed to the overall progress** of the project by developing a player that took advantage of other teams’ weaknesses, thus forcing them to improve their own player. The “follow the leader” strategy pointed out flaws in other teams, and helped the class as a whole to create better ideas.

Acknowledgements

- We acknowledge the “Oops, I fell” player for demonstrating that there were vulnerabilities in our strategy, by showing that our player needed to be able to handle obstructions caused by slow-moving riders.
- We acknowledge Team 3 for demonstrating that our player made a poor assumption that the leader of the race would necessarily be traveling at the optimal velocity.

Group member contributions

- **John Cieslewicz** was responsible for experimenting with independent "follow the leader" riders and analyzed the tournament results.
- **Ratul Malli** was responsible for assisting in the development of all implementations and analyzing the tournament results.
- **Chris Murphy** was responsible for developing the first "follow the leader" implementation and much of the documentation.
- **Lawrence Wang** was responsible for the calculations of optimal velocity and for implementing the player that was ultimately submitted.