Project 4: Olympic Road Race

13-December, 2004

Andy McDaniel (awm2001@columbia.edu), Hari Kurup (hgk2101@columbia.edu), Pavan Kumar (pj2111@columbia.edu)

Introduction:

This project aims at simulating a cycling road race. The course is divided into lanes and more than one rider can be in a lane, one behind the other. The course is assumed to be straight and flat. It is decided that all riders are homogenous and they all have the same amount of energy. This is an assumption that is different from real life scenario. A team scores 5 points for a gold medal, 3 points for silver, and 1 point for bronze. The ranks of the other riders, including whether a rider finishes the race or not, do not matter.

A bicycle is assumed to be 2 meters long, so that riders in the same lane must be at least 2 meters apart, measured from the front of each rider's bicycle. Riders cannot pass other riders in the same lane. To pass, a rider must move left or right to an adjacent lane, and then try to pass. If the adjacent lanes are occupied at a rider's position, then the rider cannot change lanes. If two riders simultaneously try to switch to the same lane, then the rider who is ahead has right of way. (In the unlikely event that two riders are at exactly the same position, the rider on the left, i.e., the one in the lane with the smaller number, has priority.) Note that the far-left and far-right lanes allow switching in only one direction. Two riders cannot exchange lanes if their positions overlap.

A rider who rides without anyone in front is exposed to the full brunt of wind resistance. This reduces the rider's energy at a faster rate than others who are riding behind (closely) other riders. Energy consumption per second is equal to $\mathbf{v}^{2.5}$ units where v is the speed of the rider. However, if there is a rider in the same lane at a distance d in front, then energy consumption is reduced by a factor f(d), i.e., the energy required is multiplied by (1 – f(d)) for 2 <= d <= 5, and 0 otherwise. In other words, there is a 30% saving in energy for a rider immediately behind another rider, and that benefit decreases linearly to zero as d approaches 5.

The speed of any rider cannot exceed 25m/sec.

STRATEGIES

The Simple Strategy

The winner or loser of the race is decided solely on strategy. The number of different strategies that a team can come out with is only limited by their imagination. It is always better to approach a complex problem in a piecewise manner, the simplest strategy first and then build upon it for more complex strategies and ideas.

Our first player was based on a very simple strategy – maintain a uniform velocity throughout the entire race. The only exception is at the beginning when the riders have to accelerate to the required velocity. We decided to have our players accelerate to an optimum velocity and maintain that velocity till the end. The optimum velocity ensures that all riders can complete the race, though they will have zero energy left once they reach the finish line. The optimum velocity is derived as follows:

D is the distance left to finish the race. E is the energy left with the player.
So, we have $(v^{2.5})*T = E$ -- (1)
$v*T = D$ -- (2)
where v is the velocity with which the player moves and T is the time required for the player to finish off the race with no energy left.
From (2) we have $T = D/v$. Using this in (1) we get $\mathbf{v = (E/D)^{1.5}}$

Line 'em up!!

The simple strategy described above was a common between all teams. The next natural evolution of strategy was to get all the players in one line and make use of the slipstream to save energy for players who have someone else ahead of them. This is a sound strategy since making using of slipstream saves 30% energy as compared to riding in front, and thus the riders can go at a speed higher than the optimum.

This kind of strategy can be found in real life cycle races. Members of opposing teams try to make use of the slipstream by staying behind the leaders, but at the same time trying not to be lagging way behind. The idea is to save energy early in the race to make use of it towards the end, for an all-out dash to the finish line. For our project all riders are considered homogenous, and so it does not make any difference as to which riders reached the finish line first.

The speed at which the players can move when aligned in a continuous line can be found as described below.

R is the number of players in the line and D is the distance left for the last player to finish the race and E is the energy required for the last player to finish the race. So, as discussed in the class, we use the formula $\mathbf{v = ((E/D)(10/7)(1-0.3^R))^{(2/3)}}$. We calculate v using this formula once when the players are lined up and every time the number of riders lined up R changes due to the death (energy becomes zero) of the players.

Making use of the slipstream increases the maximum velocity at which a player can go and we can still ensure that atleast one member crosses the finish line. Once all members of a team are in a line the entire line moves with a constant acceleration. The strategy is simple in itself, however there are some issues to be considered in bringing all team members in one continuous line.

While co-coordinating the movement of all the players in a team to bring them in one line, it is necessary to watch out for riders from other teams interrupting your line pattern.

This can happen inadvertently because other teams may have adopted the same strategy of bringing their players in one line and in the early formation stages the players from different teams may be interspersed in the same lane. The other possibility, sounds evil in some sense, is a strategy adopted by some team (enemy) to post one or a few of their players in between the line you are forming. This serves the enemy team two purposes - 1) they have interrupted your line formation and thus made it difficult for your team to get efficient performance and 2) they have managed to save energy by riding in your slipstream. There are ways to get rid of this interruption, but it will require a few maneuvers by your team.

In such situations, we decided to change lanes in group (all members of the team change lanes) and also accelerate the rider who has a competitor ahead of them in their lane. The riders who are ahead of the competitor maintain their acceleration.

	A1						
	A2					A1	
	B		$\Rightarrow$		B	A2	
	A3						
	A4					A3	
						A4	

(Before) *(After lane change)*

As can be seen from the diagram above, a rider from team B is between the line formation of team A. Riders of team A decide to change lanes. A1 and A2 have the rider B ahead of them so they accelerate and change lanes, whereas A3 and A4 are ahead of rider B so they maintain their acceleration. This allows A1 and A2 to narrow the gap and catch up with A3 and A4. This lane change and acceleration method is repeated till all the players of the team make one continuous line and the interrupting team's rider has been nudged out. In the process of changing lanes, the riders of team A may reach the extreme end lanes, in which case they start changing lanes towards the other end. The same strategy is adopted if some team right in front of us blocks our players.

The Blockers

Once we had tuned our players to ride in a single line, we decided to make the strategy dependent on the number of members in a team. Most other teams by this time had adopted the line strategy and were tuning their formula give a few extra decimal points of increase in their speed or save some units of energy. We decided to come up with a strategy that was unique. This was one of the players we submitted for the finals, and it was named Group4Player4.

If your team had more than 6 players ($R > 6$) we termed that as a scenario with a "lot of players". Under such circumstances, we would send out 3 of our players, usually players R, R-1 and R-2, as blockers. Their role is to block other teams by forming a wall of 3 in front of them and then reduce the velocity of that team to zero. Once that aim is achieved the 3 riders will accelerate and move towards another team and block them. Of course, since these players are moving at a speed higher than the safe speed they die out by around one-third of the distance. However, while all this action is happening, the remaining players are moving in a line and thus have gained some distance on the other teams. During testing we found that this strategy was very sound and worked well with other teams whose only strategy was to use the line formation. If our blockers were able to block the one or two strong players in the class, especially the 'Spear LegWeak' player, our team usually ended up being in the medals.

If the number of riders per team is less than 7, we would only do a line strategy.

The following diagram explains the blocking feature:

		B		
		B		
	A	A	A	

Riders of team A have blocked the B team.

If team B now decides to switch lanes on either side, it will still be behind team A, and during the next move A can move again to ensure B stays behind.

	B			
	B			
	A	A	A	

Riders of team B moves left to escape the blockade.

	B			
	B			
A	A	A		

Riders of team A moves left to maintain the blockade.

Team B can escape the blockade by breaking their line pattern and forming atleast two lines. When this happens our blockers will be able to block only one line effectively while team B can move their second line out of the blockade and move on. However, while this counter-strategy works, team B has lost a major advantage of having all their players in a line. So, in that aspect the blockers would have done their job.

Enter the Stalkers

Quite a bit of discussion time in class was devoted to the concept of a parasite player – one that consistently follows other players and thus makes use of their slipstream. We decided to give this player a try and this gave rise to our ‘stalker’ player. We call our player ‘Sunflower Stalker’ since our team color was yellow.
In contrast to establishing a cooperative line and implementing the so named “Optimal Strategy,” the individual riders of this team have very minimal cooperation among each other. Instead, each player behaves more or less as an individual. Because an individual is highly unlikely to win on it’s own without any drafting of other players, the stalker player tries to get behind enemy riders as much as possible.

On a very high level, the strategy followed by every rider of the team can be described as such:

- Every turn; calculate the *sprint* time and maximum velocity of every player on the board. *Sprint* refers to a straight ride to the finish line, assuming no drafting bonus for energy savings.
- Compare sprint time of every other rider to your own. Then classifying riders into *red* players, ones which can unconditionally beat you to the finish line in a straight race, and *green* players, those which can be beat in a straight race.
- Using a laterally biased forward-looking window, select a red rider to “stalk”. Attempt to match this rider’s speed and move into their lane. Do not stalk riders from your own team
- If you are drafting a player, allow yourself to ride a little faster to keep up instead of falling behind.
- Try to avoid being blocked by green (beatable) players
- Very close to the end of the race, stop drafting other players, and instead move into a clear lane and move at your calculated sprint velocity.

The majority of relevant calculations are done in the class MinTimeEstimate. Every turn, it updates the sprint times and velocities for every player. Through a series of methods, the player class can compare any two pairs of riders to see which one has an advantage. The formula used for calculating this is:

$$v = (E / d)^{(2/3)}$$
$$t = d / v$$
(d = distance remaining, E = energy remaining, t = time to finish, v = max velocity)

Besides keeping track of various rider states, the class has a method called *findStalk*, which searches a defined window space for a given rider, and returns a list of red players, sorted by how strong the opponent player is in terms of difference between the estimated finishing times. For example, when considering two targets, one which can finish in 1200 turns, and the other in 2700 turns, since the first rider is “stronger”, it will be put at the head of the list, and therefore selected as a target over the second. The definitions of the window allow the player to restrict the selection of stalking targets to those which are as close by, so that the rider does not waste time or energy pursuing a target which is far away.

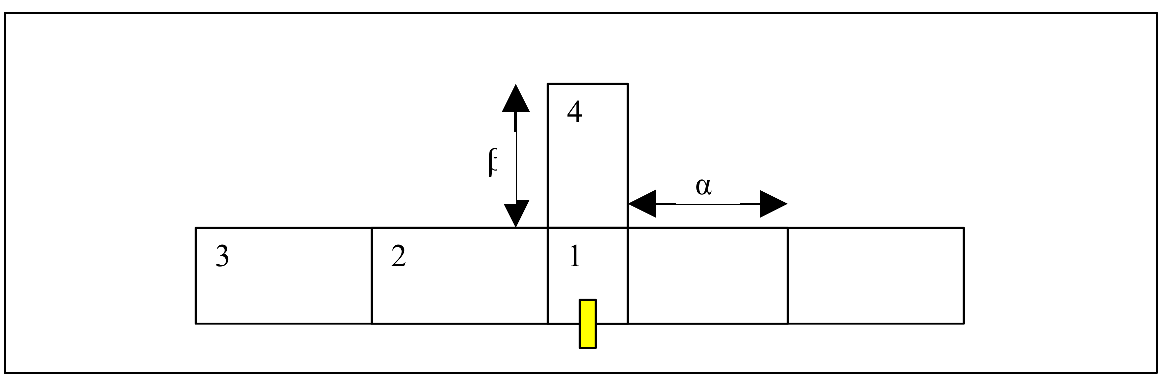
There are several constants that the player class uses to make decisions regarding rider behavior:

- *b* – a coefficient that allows a rider to exceed his calculated sprint velocity
- *l* – this defines the fraction of the race after which the rider will no longer participate in the stalker behavior, and instead sprint towards the finish of the race.
- *α, β* – these are the window interval values. *α* is how many lanes the window expands by after unsuccessfully searching for targets in a smaller window. *β* is the forward distance the window increases by after expanding the search window to cover all the possible lanes.

The values for the constants were derived experimentally. After running numerous tests we fixed the following values for the constants:

- *l* = .001
- α = 1
- β= 100
- b = 1.30

The window for target selection is always positioned directly in front of the player, and defined by a rectangle n lanes wide and m distance units long. If a suitable target is found for the rider to stalk, then the player moves to change lanes with that player, and catch up behind them so that they can draft them. If there are no suitable targets, then the window is expanded and searched again. The window is always expanded laterally first. Because lane changes don't cost anything (compared to the extra energy which must be expended to catch up to a player ahead of you) the window is always expanded laterally first. Only until all lanes have been searched is the window reset to the original lane width and then expanded distance wise. This process is repeated for a number of times until a preset number of tries have occurred (usually L*3) at which point the algorithm gives up, and the rider defaults to a draft less sprint, hoping to find a stalking target later on.



Progressively larger search windows

Lastly, if a rider is drafting another rider, or actively trying to catch one, the rider is allowed to exceed its energy budget by multiplying the sprint velocity by the constant *b*. The reason for this is that the calculated velocity is assuming no draft savings, but if there is the possibility of doing so in the near future, the rider takes a "risk" in the hopes of savings later on and gaining an advantage in the race.

TOURNAMENT RESULTS

The tables below summarize the results of the tournament.

Tournament	Games Played	Group4Player4		
		G	S	B
BASE T	27	1	4	2
R =1	25	2	5	5
R=2	26	0	1	7
R=3	23	2	2	2
R=5	24	2	6	2
R= 6	26	0	4	7
R=7	25	2	5	2
R=9	23	10	2	0
D= 30L	20	2	2	1
L= 2*t*R	33	1	4	5
E~124*D	24	1	3	4
E~5*D	25	3	3	8
TOTAL		26	41	45

Medal tally for Group4Player4 (line and blocking strategy)

Tournament	Games Played	Group4Stalker		
		G	S	B
BASE T	23	6	1	1
R =1	25	5	6	1
R=2	24	10	8	4
R=3	27	7	10	5
R=5	26	1	4	4
R= 6	24	1	1	9
R=7	25	0	1	6
R=9	27	1	5	4
D= 30L	30	5	6	2
L= 2*t*R	17	3	0	2
E~124*D	26	13	10	12

E~5*D	25	6	9	3
TOTAL		58	61	53

Medal tally for Group4Stalker

As can be seen both our players did fairly well. The stalker player performed remarkably well, proving that the parasite strategy is indeed a good one.
Since we had submitted two players, each of them played approximately half the total number of games in each tournament.

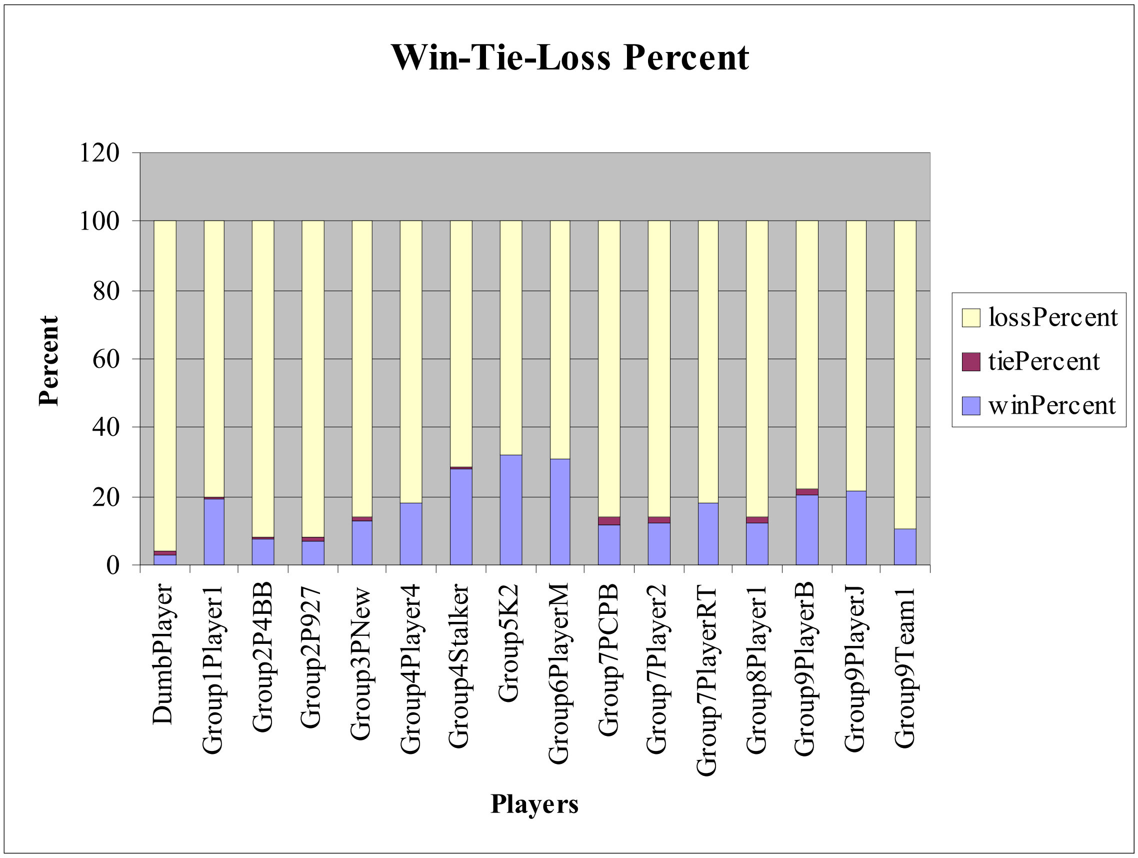
The results of the tournament with R=9 shows that the blocking strategy of Group4Player4 worked very well when there were too many players per team. Our player gathered 10 gold medals. As for the performance of this player in other tournaments, we see that it did not perform as well as it did during tests. This may be due to the fact that some teams may have introduced their own blocking strategy before the final submissions and thus it may have had a counter effect on our strategy.

As for our 'stalker' player the results have been exceptionally good. In the base tournament, we placed third in total number of wins (with 6 wins, behind Group52K and Group6PlayerM) and 2nd for percentage of games with a win (since we played fewer games, 6 wins is actually 26% success rate for achieving a single medal). Other interesting tournaments include the long distance, (D = 30*L) race, where we had the highest percentage of wins, with 16.6%, and our closest competitor, Group6PlayerM had 16% win rate. For the high-energy races (Where E=124*D) we also did well, tying with two other players for the highest raw number of wins (12), and having the second highest percentage win rate (46%).

In trying to explain why it did so well, it highlights some things about this project that may have been overlooked. First of all, through our own testing against other players, it seems that one of the largest factors in determining success is how much luck the riders have at the start of the race. If a rider can get behind another rider at the beginning of the race, or does well in drafting a number of players for an extended period of time, then it would generally win a medal of some sort. Other times, riders would die early or near the middle of the race, having expended too much energy in trying to chase drafters, and not having the investment paying off, or get stuck in a broad line as they all tried to stalk the same target.

Another thing that seemed to be happening was that the actions of each rider acting independently would become disruptive (somewhat unintentionally) to other teams. While a rider tangled up in such a mess would generally not win, it would sometimes slow down the strongest and most organized competitor enough to allow another team rider, usually one following a team less organized or concerned with others drafting behind them, to win a medal of some sort instead of the highly organized and optimal riders. Occasionally we would see behavior that can be described as a type of thrashing among teams. A team would spend so much time trying to escape or throw off a stalker rider their advantage evaporated or the lack of robustness in their line forming code would prevent them from racing effectively if a stalker had not been trying to follow them.

The chart below gives an indication of percentage of wins, ties and losses encountered by each team. A win here refers to winning the gold medal; anything else is regarded as a loss.



Once again we can see that our players have done better than most other teams. The 'stalker' ranks third in overall wins and Group4Player4 with its line and blocking strategy comes in at seven. The above chart takes into account the number of games each player from each team played. The above chart does not include results from the robustness test – two instances of the same player and games against dumb player. We had an almost 100% result in the excluded results, but we feel that those tournaments were more of a robustness test rather than a competitive race for medals that could be weighed as a win or loss.

TEAM MEMBERS AND ROLE

Andy McDaniel: Creator of the 'Stalker' player and also did the write up for this player.

Hari Kurup: Worked with Pavan on creating Group4Player4. Did the project report and presentation.

Pavan Kumar: Worked with Hari on creating Group4Player4. Creator of the blocking strategy. Completed the write up for the formulas used in our players.

CONCLUSION

We believe the line and parasite strategy are very effective for this project. However, we also think that the line strategy is very basic and can easily be thwarted by blocking strategies. The parasite is more versatile in that aspect since all riders in the team can behave independently and thus will not all be caught behind a blockade. Moreover, the line strategy involves lot of coordination between riders of the team and this increases code complexity. We think a good combination of line and parasite strategy would be the ideal strategy for this game.