Robot Kabaddi

Kyle Klein Subhash Suri

Department of Computer Science
University of California
Santa Barbara, CA 93106, USA

1 Introduction

Imagine a situation in which two (or more) robots are required to capture a single rogue robot. We draw inspiration from *Kabaddi*, a popular team sport originating in South Asia, and study a discrete version of the game as a vehicle to study this kind of multiagent pursuit evasion problems. The game of *Kabaddi* involves two teams occupying opposite halves of a field, each team taking turns to send an "attacker" into the other half, in order to win points by tagging or wrestling members of the opposing team [11]. The attacker must *hold his breath during the whole attack* and successfully return to his own half—the attacker continuously chants "kabaddi, kabaddi, $\cdots$" to demonstrate holding of the breath. There are several elements of this game that distinguish it from other well-studied and similar sounding games such as *man-and-the-lion* [7, 10], *cops-and-robber* [4, 1, 5], *robot-and-rabbit* [4], and *pursuit-evasion* [9, 3]. Perhaps the most significant difference is that in kabaddi players of *either* side may capture an opponent, while in these other games one side always plays the role of captors and the other evaders. But there are several other differences as well, some of which we highlight below.

In the man-and-the-lion game, for instance, the chase occurs inside a circular region, both players have the same maximum speed, and it is known that the man can evade capture indefinitely. In contrast, the game of kabbadi pits a single attacker against multiple defenders. The other games such as the *cops-and-robber* and *pursuit-evasion* differ from kabaddi in the way capture occurs as well as the *information* about the evader's position. For instance, the current position of all the players is public information in kabaddi while the position of the robber or evader is often assumed to be unknown to cops or pursuers. Furthermore, it is also typically assumed that each cop (robot) follows a fixed trajectory that is *known* to the robber (rabbit). This makes sense in situations where the defenders (cops) have fixed patrol routes, but not in interactive games like kabaddi. The problems and results in the graph searching literature are also of a different nature than ours [2, 6]. The pursuit-evasion games have also focused on *visibility* based capture, where it is sufficient for some pursuer to "see" the evader—both infinite visibility or limited-range visibility models have been considered [3, 5]. By contrast, kabaddi requires a physical capture that leads to a very different set of strategies and game outcomes. (The book by Nahin [8] offers a nice historical perspective on various pursuit-evasion type games.) With this background, let us now formalize our model of kabaddi.

1.1 The Standard Model

We assume that the game is played on a $n \times n$ grid S, whose cells are identified as tuples (i, j), with $i, j \in \{1, 2, \dots, n\}$. We focus largely on the game between one attacker and two defenders, which already proves to be quite challenging and intricate to analyze. We use the letters A and D to denote the *attacker* and a *defender*, respectively. When there are multiple defenders, we use subscripts such as D_1, D_2, etc. We need the concepts of *neighborhood*, *moves*, and *capture* to complete the description of the game.

Neighborhood. The *neighborhood* $N(p)$ of a cell $p = (i, j)$ is the set of (at most) 9 cells, including p itself, adjacent to p, or equivalently the set of all cells with L_∞ distance at most 1 from p. In Figure 1, the neighborhood of A is shown with a box around it. Slightly abusing the notation, we will sometimes write $N(A)$ or $N(D)$ to denote the neighborhood of the current position of A or D.

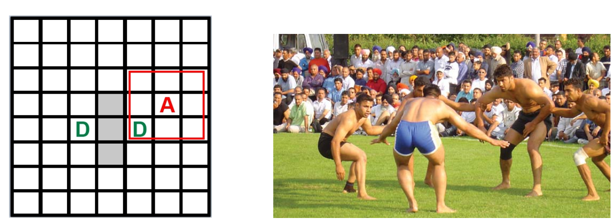

Figure 1: The standard model of kabaddi. A can capture the defender closer to it, which is inside $N(A)$. The defenders can capture A at any position in the shaded region, which is the common intersection of their neighborhoods. The right figure shows how defenders try to surround the attacker in the real game of kabaddi.

Moves. The attacker and the defenders take turns making their moves, with the attacker moving first. In one step, the attacker and the defenders can move to any cell in their neighborhood. *All the defenders can move simultaneously in one step.*

Capture. A *captures* a defender D if it is the unique defender lying inside the neighborhood of A. That is, with two defenders, D_1 is captured when $D_1 \in N(A)$ and $D_2 \notin N(A)$. (Notice that A only needs to enclose a defender within its neighborhood to capture it.)

Conversely, the defenders capture the attacker, when A lies in the common intersection of the two defenders' neighborhoods. That is, $A \in (N(D_1) \cap N(D_2))$.

Game Outcome. The attacker *wins* the game if he can capture all the defenders, and the defenders win the game if they can capture the attacker. If neither side wins, then the game is a tie.

This particular form of capture has a tendency to make defenders always stick together, and fails to model the real world phenomenon where defenders try to "surround" the attacker—see figure above. We therefore introduce a *minimum separation* condition on the defenders in the following way:

> *no defender can be inside the neighborhood of another defender.*

These rules together define our *standard model* of kabaddi. Other models can be obtained by varying the definition of the *neighborhood* and relaxing the separation condition for defenders, and we obtain some results to highlight the impact of these modeling variables.

In the actual game of kabaddi, the attacker must hold his breath during the attack. We handle this constraint *implicitly* by analyzing the worst-case number of moves it takes to achieve a win for the attacker—that is, the duration of the game acts as a proxy for the length of time the breath must be held.

1.2 Our Results

We first consider the simple setting of a single attacker A against a single defender D, and show that the attacker can always capture D in $O(n)$ number of steps, which is clearly optimal, upto a constant factor, in the worst-case

The game becomes more challenging to analyze with two defenders, where the attacker continuously runs the risk of being captured himself, or have the defenders evade him forever. Our main result is to show that the attacker has a winning strategy in worst-case $O(n^2)$ moves. One important aspect of the standard model is the *separation* requirement for the defenders—each must remain outside the neighborhood of the other. Without this restriction, we show that the two defenders, whom we call *strong defenders* to distinguish from the standard ones, can force a draw: neither the attacker nor a defender can be captured. A further modification of the model, which disallows the *diagonal* moves, tips the scale further in the favor of strong defenders, allowing them to capture the attacker in $O(n^2)$ steps.

Extending the analysis to more than two players is a topic for ongoing and future work, and seems nontrivial. In the standard model, it is not obvious that even $\Theta(n)$ defenders can capture the attacker, nor it is obvious that the attacker can win against k defenders, for $k > 2$. (The definition of capture remains the same: two defenders are enough to capture the attacker.) Due to lack of space, most of the proofs are omitted from this abstract.

2 Playing against a Single Defender

We begin with the simple case of the attacker playing against a single defender. Besides being of interest in its own right, it also serves as building block for the more complex game against two defenders. We show that in this case the attacker always as a winning strategy in $O(n)$ moves.

Throughout the paper, we assume that the grid is aligned with the axes, and use $\Delta x = |D_x - A_x|$ and $\Delta y = |D_y - A_y|$, resp., for the x (horizontal) and the y (vertical) distance between A and D.

Theorem 1 *The attacker can always capture a single defender in a $n \times n$ game of kabaddi in $O(n)$ moves.*

Proof. The attacker's basic strategy is to chase the defender towards a wall, keeping him trapped inside a continuously shrinking rectangular region. Specifically, as long as $\min\{\Delta x, \Delta y\} > 0$ on its move, the attacker makes the (unique) diagonal move towards the defender, reducing both Δx and Δy by one. Because the grid is $n \times n$, the attacker can make at most n such moves before either Δx or Δy becomes zero. Without loss of generality, suppose $\Delta x = 0$. From now on, the attacker always moves to maintain $\Delta x = 0$ while reducing Δy by one in each move. Because Δy can be initially at most n, the attacker can reduce to it one in at most $n - 1$ moves, at which point it has successfully captured the defender because both Δx and Δy are at most 1. This completes the proof. □

3 Playing against Two Defenders

The game is more complex to analyze against two defenders, and does not seem to admit a simple non-adaptive strategy. We begin by isolating some necessary conditions for the game to terminate, or for the next move to be safe. We then discuss the high level strategy for the attacker, and show that it can pursue the defenders using that strategy *without being captured* itself. Together with a bound for the duration of the pursuit, this yields our main result of $O(n^2)$ steps win for A in the standard model. We denote the two defenders by D_1 and D_2, and use D to refer to a non-specific defender when needed. Throughout the game, *we ensure that whenever A makes a move, it is safe in the sense that it cannot be captured by the defenders in their next move.*

Lemma 2 *On A's turn, if $\max\{\Delta x, \Delta y\} \leq 2$ for at least one of the defenders, then A can capture a defender in one step. Conversely, on the defenders' turn, if $\max\{\Delta x, \Delta y\} > 2$ for one of the defenders, then they cannot capture A on their move.*

Proof. We first observe that neither defender can be inside the neighborhood of A, namely, $N(A)$. This holds because a single defender inside $N(A)$ must have been captured in A's last move and if both the defenders are inside $N(A)$, then they would have captured A in their last move. Thus, we must have $\max\{\Delta x, \Delta y\} \geq 2$ for both the defenders.

Let D_1 be the defender that satisfies the conditions of the lemma, meaning that $\max\{\Delta x, \Delta y\} = 2$. If both the defenders satisfy the condition, then let us choose the one for which $\Delta x + \Delta y$ is smaller; in case of a tie, choose arbitrarily. Without loss of generality, assume that D_1 lies in the upper-left quadrant from A's position (i.e. north-west of A). We now argue that A can always capture D_1 as follows. See Figure 2.

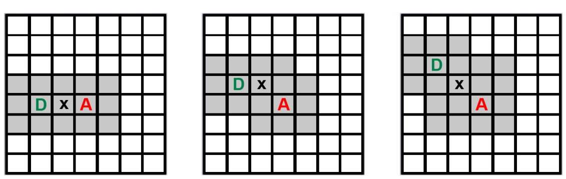

Figure 2: Illustrating the three cases in Lemma 2: $\Delta x + \Delta y = 2, 3$ and 4. The shaded area is the region that cannot contain the second defender.

If $\Delta x + \Delta y = 2$, then we must have either $\Delta x = 2, \Delta y = 0$ or $\Delta x = 0, \Delta y = 2$. In the former case, A can capture D_1 by moving to its x-neighbor (shown in the left figure), and in the latter by moving to its y-neighbor. Since the second defender must lie outside $N(A) \cup N(D_1)$, this move cannot cause A to be captured. Similarly, if $\Delta x + \Delta y = 3$, then we have either $\Delta x = 2, \Delta y = 1$, or $\Delta x = 1, \Delta y = 2$. In both cases, A captures D_1 by moving to its north-west neighbor $(A_x - 1, A_y + 1)$, as shown in the middle figure. Observe that, by the minimum separation rule, if there is a defender at $(A_x - 2, A_y + 1)$, then there can't be one at $(A_x - 1, A_y + 2)$, and vice versa ensuring the safety of this move—there also cannot be a defender at $(A_x, A_y + 2)$ because that would contradict the choice of the closest defender by distance.

Finally, if $\Delta x + \Delta y = 4$ (shown in the right figure), then A captures D_1 by moving to $(A_x - 1, A_y + 1)$. This is a safe move because the only position for D_2 that can capture A is at $(A_x, A_y + 2)$, but in that case D_2 is the defender with the minimum value of $\Delta x + \Delta y$, contradicting our choice of the defender to capture. This completes the first claim of the lemma. For the converse, suppose that $\Delta x > 2$ for defender D_1. Then, after the defenders' move, A is still outside the neighborhood of D_1, and so A is safe. This completes the proof. □

The attacker initiates its attack by first aligning itself with one of the defenders in either x or y coordinate, without being captured in the process. The following two technical lemmas establish this.

Lemma 3 *A can move to the boundary in $O(n)$ moves without being captured.*

Lemma 4 *By moving along the boundary, A can always force either $\Delta x = 0$ or $\Delta y = 0$ for one of the defenders in $O(n)$ moves, without being captured.*

The Second Phase of the Attack

Having reached the starting position for this second phase of the game, we assume without loss of generality that A is at the bottom boundary, and that after A's last move, $\Delta x = 0$ for one of the defenders. From now on, A will always ensure that $\Delta x \leq 1$ for one of the defenders *after each of A's moves*. The x-distance can become $\Delta x = 2$ *after the defenders' move* but A will always reduce it to 1 in its next move.

By Lemma 2, if both Δx and Δy are at most 2, then A can win the game. On the other hand, if the players are too far apart, then both sides are safe for the next move. Thus, all the complexity of the game arises when the distance between A and the defenders is 3, requiring careful and strategic moves by both sides. We show that A can always follow an attack strategy that ensures a win in $O(n^2)$ steps, while avoiding capture along the way.

In order to measure the *progress* towards A's win, we use the distance from A's current position to the top boundary of the grid *while ensuring that $\Delta x \leq 1$ continues to hold*. In particular, define $\Phi(A)$ as the gap between the current y position of A and the top boundary. That is, $\Phi(A) = (n - A_y)$, where this gap is exactly $n - 1$ when the second phase begins with A on the bottom boundary. We say that A *makes progress* if $\Phi(A)$ shrinks by at least 1, while Δx remains at most 1 for some defender. Clearly, when the $\Phi(A)$ reaches zero, A has a guaranteed win (by Lemma 2). If the attacker succeeds in capturing a defender, then we consider that also progress for the attacker.

Lemma 5 *On A's move, if $\Delta y = 3$ and $\Delta x = 0$ holds for some defender, then A makes progress in one move.*

Lemma 6 *On A's move, if $\Delta y = 3$ and $\Delta x = 1$ holds for some defender, then A makes progress in $O(n)$ number of moves.*

Lemma 7 *If $\Delta y = 3$ and $\Delta x = 2$ for some defender say D_1 then A may make progress in $O(n)$ moves.*

We can now state our main theorem.

Theorem 8 *In the standard model of kabaddi on a $n \times n$ grid, the attacker can capture both the defenders in $O(n^2)$ worst-case moves.*

4 Playing Against Strong Defenders

In the standard model, each defender must remain outside the neighborhood of other defenders; that is, $D_i \notin N(D_j)$, for all i, j. The defenders become more powerful when this requirement is taken away. Let us call these *stronger* defenders. In this case we explore what happens when we remove the stipulation that the defenders cannot be within each other's neighborhoods. This creates two stronger defenders and as a result creates a game where ideal play means not only can the attacker not win, but the defenders cannot either. We assume that play starts with defenders already in a side-by-side position, that is, $\Delta x + \Delta y = 1$ with respect to D_1 and D_2's coordinates.

Theorem 9 *Under the* strong *model of defenders, there is a strategy for the defenders to avoid capture forever. At the same time, the attacker also has a strategy to avoid capture.*

5 Strong Defenders with Manhattan Moves

Thus, in the standard model but with strong defenders, we have a tie, and neither side can guarantee a win. In the following, we show that if we disallow the *diagonal moves*, permitting a player to move only to its left, right, up, and down neighbors, then the defenders have a winning strategy. That is, the movement metric is Manhattan metric—a player can only move to a cell within the L_1 distance of 1 from its current cell. The definition of the capture, however, remains the same as in the standard model.

Theorem 10 *Two strong defenders playing under the Manhattan moves model can always capture the attacker in $O(n^2)$ moves.*

6 Closing Remarks and Extensions

We introduced an abstract discrete model for the game of kabaddi. To the best of our knowledge, this is the first attempt to formally model and analyze the game. Our analysis shows that even with two defenders the game reveals significant complexity and richness. The game also shows surprising sensitivity to the modeling parameters that control the moves and the power of players.

Our work poses as many open questions as it answers. First, can the attacker win against two defenders in $o(n^2)$ moves? We suspect that an $O(n)$ winning strategy exists. The game against more than two defenders seem particularly interesting. What is the outcome of the game in the standard model against $d > 2$ defenders? The minimum separation rule leads to some pesky modeling problems because the attacker could sit in a corner cell and not be captured. So some modification is needed in the rules to avoid such deadlocks.

References

[1] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1 – 12, 1984.

[2] D. Bienstock and P. Seymour. Monotonicity in graph searching. *J. Algorithms*, 12(2):239–245, 1991.

[3] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. Visibility-based pursuit-evasion in a polygonal environment. *IJCGA*, 9(5):471–494, 1999.

[4] B. Halpern. The robot and the rabbit–a pursuit problem. *The American Mathematical Monthly*, 76(2):140–145, 1969.

[5] V. Isler and N. Karnad. The role of information in the cop-robber game. *TCS*, 399(3):179 – 190, 2008.

[6] A. LaPaugh. Recontamination does not help to search a graph. *J. ACM*, 40(2):224–245, 1993.

[7] J. E. Littlewood. *Littlewood's Miscellany*. Cambridge University Press, 1986.

[8] P. J. Nahin. *Chases and Escapes*. Princeton University, 2007.

[9] T. D. Parsons. Pursuit-evasion in a graph. In Y. Alavi and D. R. Lick, editors, *Theory and Application of Graphs*, pages 426–441. Springer-Verlag, Berlin, 1976.

[10] J. Sgall. Solution of david gale's lion and man problem. *Theor. Comput. Sci.*, 259(1-2):663–670, 2001.

[11] Wikipedia. Kabaddi. http://en.wikipedia.org/wiki/Kabaddi, 2009.