Decision Mathematics and its Applications in Sports

Kang Ji

July 2023

1 Introduction

Decision mathematics as a field of study can be broadly defined as mathematics pertaining to the making of a decision. Naturally, it is therefore an umbrella term, covering a wide range of topics. We can split the majority of what comes under decision mathematics into two main categories; discrete mathematics and decision theory. Discrete mathematics refers to anything involving a fixed set of variables and in the case of this essay mainly refers to networks, algorithms and linear programming despite covering a much wider range of topics such as cryptography while decision theory in the context of this essay effectively refers to game theory, as the games covered are all with multiple players whose payoffs are dependant on the actions of other players.

To cover a greater range of applications, I have structured this document into two parts, one part covering the applications of discrete mathematics and modelling to a more physical sport, badminton. and another part covering the applications of the game theory aspect of decision mathematics to more mental sports, such as chess.

2 Modelling Badminton Footwork

The aim of this section is to identify the best badminton footwork, defined as the shortest path starting from the middle of the court and ending back at the middle of the court which allows the player to hit any possible shot that the opponent can hit.

If we look at a badminton court, we can assign the different points of a network at different points along the court. Based on the assumption that the player can hit any shot within a radius of r around them, we can form graphs and models of a badminton court. For the best footwork path, we need as few nodes as possible, i.e. we need to fill up the entire side of a badminton court with circles with a radius of r. One method to do this could be to use bin packing algorithms.

Firstly, a value for r must be calculated

$R = R_T = R_R + R_A = 0.59m + 0.56m = 1.15m$ Where R_T = total reach, R_R = reach from racquet and R_A = reach from arm

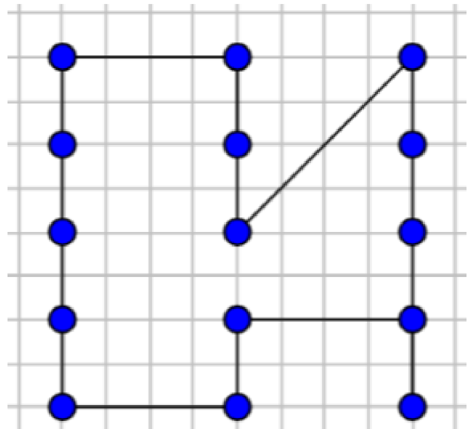
For the length of the court (6.71m) a minimum of 4 circles would be needed, however, in this case, there would still be gaps between the circles. To prevent gaps, we would need to form a stadium, meaning that the length covered by each circle (discounting the 2 "outer" circles) is given by $r(n+1)$

Since that must equal 6.71, we can calculate the number of circles required to cover the entirety of a 6.71m x 2m rectangle. Given r is 1.15m, we can therefore solve the equation 1.15(n+1) = 6.71 to give n= 4.8348 meaning that the minimum number of circles required to fully cover the length is 5. We can then divide the width of the court by the width of each rectangle formed (2r = 2.3m), 6.1 / 2.3 = 2.65 therefore a reasonably accurate estimate for the total number of circles required is 3 x5 = 15 which roughly correlates with the rough estimate of $6.71 \times 6.1/(1.15^2 \times \pi) = 9.85$ The additional five circles come from the fact that the circles cannot perfectly fit inside the rectangular court.

2.1 Obtaining the optimum solution for the model

We can therefore model the court as 15 points arranged in a 5x3 rectangle. The points can then be numbered from 1 to 15 left to right with the top left point being 1 and the bottom right being 15. For the current model, we will assume that each of the nodes is equally important, i.e., the best footwork path is the path which covers all 15 nodes in the shortest possible amount of time. As we assume the speed of the player is constant, we only need to look at minimising the distance travelled by the player. To do this, we could form a minimum spanning tree by using Kruskal's algorithm to generate a minimum spanning tree. In general, there are only 3 lengths we really need to consider, $r, 2r, and\sqrt{3r^2}$ there are multiple different minimum spanning trees of which figure 2 is the worst path for movement (assuming you start at point 5, $5 \rightarrow 1 \rightarrow 6 \rightarrow 10 \rightarrow 6 \rightarrow 11 \rightarrow 15$ and back; ie, there are lots of repeating nodes) a better path of movement must have fewer repeating nodes so we look to connect the nodes in such a way that they display a path which represents the minimum distance required to be travelled if a player starts at node 8 and travels to all of the other nodes before ending back at node 8 (middle of court)

This ends up being a hamiltonian cycle (by definition) and a travelling salesman problem. As a result, it is very difficult to come up with the optimal path (doing so would require going through 14! possible permutations, same start and end point, can immediately eliminate all options which have greater lengths) and so we can use several different algorithms to find reasonably strong paths to work out a reasonably strong path, we could use the following steps 1, Work out lower bound 2, Greedy algorithm (nearest neighbour) 3, Tour improvement algorithm 4, Ant algorithm 5, Simulated annealing 6, Concorde programme

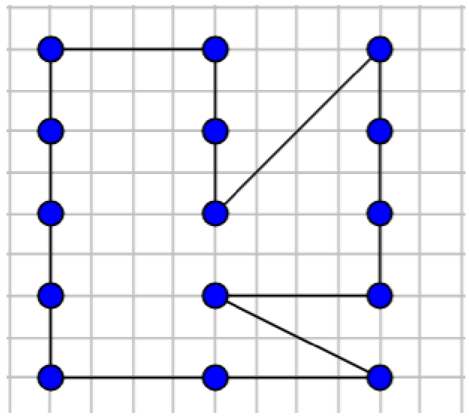


2.1.1 Approximation algorithms

Using Prim's algorithm, we can work out the lower bound using the minimum spanning tree, figure 3 giving us a lower bound of $(11 + 4 + 2)r$.We'll use this later to verify our answer. To work out an initial path, a greedy algorithm can be used such as the nearest neighbour algorithm which forms the figure below however, this is clearly a very inefficient route given that the point 12 is passed through twice.

An alternative approximation algorithm is the Beardwood-Halton-Hammersley Theorem (Beardwood, Halton, Hammersley, 1959) which uses the fact that the length of the optimal solution converges in probability to a constant. However, this approach scales very well and becomes more accurate as n tends to infinity, with a smaller value for n (15), the approximation that this method would provide would not be useful.

Next, we can try improving this path which clearly seems non optimal using the tour improvement algorithm. Immediately, we can eliminate the path $11 \to 12 \to 15 \to 9$ given the repeated point (12) and we can replace it with $11 \to 15 \to 12 \to 9$. While this is not the optimal path (since heuristic techniques have been used) , it is a reasonably good path which we can get very quickly. Since this is a travelling salesman problem, there is no easy way to prove that the path obtained is the shortest possible path other than going through all possible iterations. Given that our instance of TSP is in a metric space, we can also use Christofide's algorithm (Christofide, 1976) to find a very close approximation of the optimal path, which we could then attempt to improve to obtain the optimal solution by using the cutting plane method (Dantzig, 1957). Annoyingly, this only provides the same solution as the minimum spanning tree (Christofide's algorithm combines pairs of points together and the minimum spanning tree to work out the shortest path to connect all of the pairs of points together, one step is to make a set of vertices that have an odd degree, in the minimum spanning tree, there is only one point with an odd degree)



2.2 Iterative techniques

To obtain a better result, several other more complex approaches can be taken such as simulated annealing and the ant algorithm. Both algorithms use the idea of iteration and probabilities to slowly work towards the optimal path, the ant algorithm relies on "ants" (randomly generated paths) and the "pheromones" they leave behind (the greater the amount of times an edge has been used, the more likely it is to be chosen again). The ant system algorithm (Dorigo, Maniezzo, Colorni, 1996) is the most useful from all ant colony optimisation algorithms. Each "ant" obeys the following rules

1. It must visit each point exactly once;

2. A distant point has less chance of being chosen

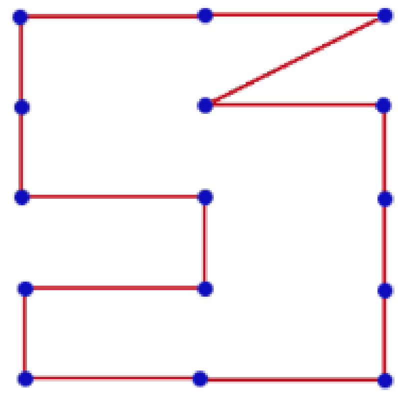
3. The more intense the pheromone trail laid out on an edge between two cities, the greater the probability that that edge will be chosen

4. Having completed its journey, the ant deposits more pheromones on all edges it traversed, if the journey is short (we can say that an ant deposits a fixed number of pheromones on its journey; number of pheromones on each edge in a path of one ant = K/W where W is the total weight of all of the edges travelled through

5. After the nth iteration, the pheromones from the iteration n-8 are removed

To apply this, we could construct formulae for the probability of an ant choosing a given edge. The general algorithm is on the right (probability = amount of pheromone x attractiveness of the move divided by the sum of all possible moves and their corresponding pheromone and attractiveness levels, attractiveness and pheromone levels are also raised to alpha and beta respectively). For our purposes, we could simplify this. Firstly, we only need to consider all sides$2\sqrt{2}r$ or less (given the longest length in the minimum spanning tree is $2\sqrt{2}r$) and we can also let alpha and beta both equal 1. Simulated annealing (Kirkpatrick, Gelatt, Vecchi, 1983) is similar to the ant algorithm in that it attempts to model useful real life occurrences. Rather than ants, it models heating up a material and attempting to find the state (permutation of the hamiltonian cycle) with the least internal energy (least weight for a path). It starts with an arbitrary temperature (in our case, we can use $(15 + 2\sqrt{2} + \sqrt{5})r$ as this is the weight of the improved minimum spanning tree) and slowly decreases the temperature to find the optimal solution. To avoid getting stuck in local maxima (paths where small changes don't make worse paths but its still possible to form a better solution) the model encourages making bad moves earlier on at higher temperatures while being more precise and more fine tuned the lower the temperature gets. After generating a similar permutation of the current path, we can work out the temperature of the new state. if it is of a lower temperature, then it is accepted while if it is of a higher temperature, then the probability of accepting the change is $e^(\Delta E/T)$ ie, the higher the temperature, the greater the probability of accepting a change of a fixed change in energy. We could further add on the concept of ant algorithms and previous iterations to further improve this; previous iterations contribute to the probability of accepting a change. Independently, both the ant algorithm and simulated annealing have the issue of getting stuck at local maxima. If we add the concept of repulsive pheromones, we can adjust how much the algorithm varies. This comes with the drawback that it will take a longer time to compute, but it also means that there is a greater chance of obtaining the global maxima rather than a local maxima. We can adjust the algorithm from $p(a) = e^(\Delta E/T)$ to $p(a) = e^(\Delta E/(T \times P)$. An alternative way to avoiding local maxima could also be to either have several simulated annealing algorithms running at once from different start points (positive pheromones could be added, which impact all of the algorithms) but this would effectively be a brute force approach albeit much faster; we can plot the results on a graph, with a base starting path with a variance value of zero, we can plot the different programmes as lines and see if they tend towards the same solution, the more times this is repeated, the more likely the solution achieved is the correct solution. However, this is arguably slower than just using a brute force algorithm.

Using these algorithms provide us with permutations of the same path Given the symmetry of the graph, there are multiple variations of this graph with the same total path length, there are 3 key aspects 1, The path goes around in a cycle around all of the outer points 2, The path joins the middle point and another inner point (inner is defined as a point which is not an outer point in a U shape 3, The path joins the 3rd inner point not yet in the cycle with a Z shape This generalisation is much more useful to us, as it's much easier to apply to a range of situations. It also means that we can avoid some of the issues which arise from the assumptions made. The generalisation also means we can draw some conclusions about the pathing of players.



2.3 Evaluation

In conclusion, this approach gives the optimal solution given that the assumptions made are correct, which is not always true. However, despite this it still exhibits footwork patterns commonly found in Badminton such as the "Z" footwork shape. It's also a "one size fits all approach; some factors such as r, the weighting of each point, etc vary however, it can certainly be said that it is a very strong footwork pattern

Overall, there are several assumptions which have been made for this model in order to simplify it as well as multiple limitations,

2.3.1 Assumption 1, Each node has an equal weighting

one of which is the assumption that all points have an equal weighting which is of course unrealistic (weight in this case refers to the "importance of each point rather than the distance between points). To assign weights, we can look at the frequency of shots into the area of each point and give each point a weighting equal to the proportion of shots hit by the opponent which end up in the range of the point (r) . Naturally, the exact values vary from opponent to opponent however generally some shots are much better strategically (corners) than others and so the average values correlate to the values of most opponents,(excluding extreme outliers). In addition, weights would also be conditional on the current position of the player and the opponent (eg, if the opponent just hit a shot to the back of the court, they are likelier to hit the next shot to the front of the court to make the player move) and so the model becomes much more complex as the number of moves increases (due to a lack of data and too many possibilities, much faster to rely on instinct). However, this is a smaller issue than expected due to the fixed moves and locations that the shuttle could be hit to meaning that rather than exponentially increasing per move, every possible can be calculated and can be applied irrespective of the point in the rally (the number of moves played) does not reduce the utility of the previous calculations of optimal paths; it's still very important to hold the entirety of the court in the shortest possible distance and therefore the shortest possible amount of time.

2.3.2 Assumption 2, It is equally easy to travel in any direction in a straight line of a given length

Another assumption made is that it is equally easy to travel in a straight line of a given length as it is to travel forwards and backwards for an overall distance of the same given length which is obviously not true. While this is a more extreme example, this problem occurs several times in different optimal paths (in different situations) albeit on a smaller magnitude. The reason behind this assumption is obviously to simplify the problem.

However, after working out the optimal path and given the symmetry of the problem, it's not hard to work out a path without this assumption; we can identify the variation of the cycle which gives the shortest time under the new conditions. It's important to note that this comes with a tradeoff; the path would favour travelling in longer lines (to minimise the amount of time spent changing direction and accelerating) which would result in a path which spends a long period on one side of the court before moving onto the other

However, it is important to note that rather than travelling forwards and then suddenly turning left, the player is much more likely to move in a curve between the two points. While the optimum path calculated is meant to be the best possible path, it is unlikely to used/perform the exact footwork in an actual match (the entirety of the court may not be covered, etc). While the actual application of this footwork is very limited given the fast paced nature of the game, it is still useful as a generally good footwork.

3 Applications of Game theory to Chess and variations of chess

While Von Neuman identified chess as a well defined form of computation rather than a game, certain aspects of game theory can and are applied by chess players, such as the concept of a Nash equilibrium, combinatorial game theory and backwards induction. Backwards induction is the easiest to cover; rather than predicting the possible moves that could be played by the opponent

When looking at chess and similar games, it's important to first classify and describe the type of game that chess is. We can describe standard chess as a symmetric zero sum, sequential game of perfect information. Chess is inextricably linked to mathematics; many prominent chess players have previously had a history in mathematics, such as Emmanuel Lasker.

Game theory can be applied to chess in several ways, such as attempting to solve chess and applying game theory techniques to playing chess.

Mathematics has focused heavily on attempting to solve chess, what's sometimes less focused on is the proof that chess has a solution One way to attempt this proof is to use a game tree and backwards induction. Obviously, with every move, the number of permutations of the board exponentially increases, however we can look at a simplification which can then be applied. Firstly, we need to greatly simplify the game tree by saying that, for every point, there are 3 possible moves, a move which makes white win, a move which makes white draw and a move which makes white lose. If it is black's turn to move, they choose the move which makes white lose, if it is white's turn, they choose the move which makes white win. Starting at the leaves of the tree, let the penultimate choice be that of either white or black, then label that node with the outcome of the leaf given that white/black plays optimally, then the node above that is either black or white and the node is labelled with the outcome of the leaf and so on. This collapses into a small tree with white to move (given that white starts first) with two possible moves, of which white plays the winning move. We can now start addressing our assumptions, firstly there is no first move as white which guarantees a victory as white as the majority of choices either worsen the position of white or keep the strength of the white pieces constant with little to no moves to strengthen whites position (dependant on black playing sub optimally). As the ideal path can be extrapolated, we can say that white has at least one line along which white gains the best possible outcome assuming black plays optimally thereby proving that a solution exists for white. Whether this solution is a win or a draw for white is much harder as that would force you to consider the actual outcomes of each move rather than just saying the best move in any position never worsens white's position and is much harder to calculate.

Another application of game theory to chess comes in "armageddon" situations as a tiebreak in chess tournaments. An armageddon is where a decisive game must be played and so a draw for black counts as a win for black. To compensate for the drawing odds, white is given additional time compared to black, in more recent armageddons, instead of time per side being preset, players are given the option of bidding the smallest amount of time they would play black for rather than times being preset and sides drawn by random in order to prevent complaints about the unfairness of times (given that this is the purpose, a second price sealed bid auction which encourages players to bid their true valuation may be better) . Given the bid of their opponent is unknown and they only have one opportunity to bid, it is a first price sealed bid auction. We can assume 3 different scenarios, where both players have an equal value for black, where player A values black more than player B and where player B values black more than player A. An assumption made is that the players do not know how much their opponent values the draw odds (this depends on context, so we can split situations into incomplete and near complete/complete information scenarios). While this shares many characteristics with a first price sealed bid auction, it's important to note that, given chess is zero sum, the value gained for one player is equal to the value lost for the other, i.e. the players want to

1, Maximise their utility (greatest amount of time as black) 2, Minimise the amount of value that their opponent gains if they win

Given that information is asymmetric and players don't know the valuation of their opponent, the equilibrium is for both players to play their true bids. Although, players would likely have a rough approximation of how strong their opponent is as black, it's very hard to precisely predict the valuation of the opponent and so the asymmetric assumption holds true and so we can say that in all cases, the best option is to bid their true valuation

Overall, decision mathematics has a very wide range of applications and especially the discrete mathematics aspect has a great potential for use in sports, not only for competitive play, but also for fun.

References

Beardwood, J. and Halton, J.H. and Hammersley, J.M. (2008) "The Shortest Path through Many Points Available at: https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/abs/shortest-path-through-many-points/F1C28B5730B94887F4659FCBF8A1F2BB (Accessed: 1st July 2023)

Chritofide, N. "Worst Case Analysis for a New Heuristic of the Travelling Salesman Problem" Available at: https://apps.dtic.mil/dtic/tr/fulltext/u2/a025602.pdf (Accessed: 29th June 2023)

Dorigo, M. and Maniezzo, V. and Colorni, A. (1996) "The Ant System: Optimization by a colony of cooperating agents" Available at: http://www.cs.unibo.it/babaoglu/courses/cas05-06/tutorials/Ant$_{C}olony_{O}ptimization.pdf(Accessed:28thJune$2023)

Kirkpatrick, S. and Gelatt, C. D. Jr. and Vecchi, M. P. (1983) "Optimisation by Simulated Annealing" Available at: http://cs.gettysburg.edu/ tneller/cs371/OptBySimAnneal.pdf (Accessed:30th June 2023)