How sequential games can be solved by backward induction economics essay

Game theory has been widely acknowledged as an important tool in many fields. Its attraction is prominent and its importance is explained by Robert Aumann and Oliver Hart in the following way:

“ Game theory may be viewed as a sort of umbrella or ‘ unified field’ theory for the rational side of social science…it develops methodologies that apply in principle to all interactive situation” (Aumann and Hart, 1992).

It is not difficult to understand the enthusiasm towards theories of games developed from various game types and game-solution analyses. This essay will focus on a particular sort of games, namely, sequential game and the solving method of backward induction.

Sequential games are those in which players take turns moving or make moves at different times. This means that players moving later in the game have additional information about the course of others’ actions. This also means that players who move first can often influence the game. Each player makes decisions conditional on what other players have done.

Consider a sequential game where there is an incumbent (Macrosoft) and an entrant (Microcorp). Macrosoft decides on a marketing strategy for its new software game. It can choose either a slick campaign or a simple campaign. Macrsoft faces potential competition with “ legal clones” of its game from Microcorp. It moves first and Microcorp observes its action. Regardless of what Macrosoft chooses, Microcorp then has two options: entering the market, or staying out of the market. The two firms’ payoffs are displayed in table 1:

Table 1: The payoffs for software game

Macrosoft’s ad campaign

Slick Simple

(-$250, $380)

($100, $400)

($0, $430)

($0, $800)

Payoff (in $1, 000s)

Microcorp’s entry decision

Stay out

Enter

Figure 1: The game tree for software game

($380, 000, -$250, 000)

($430, 000, $0)

($400, 000, $100, 000)

($800, 000, $0)

Microcorp’s entry decisionIn order to establish the set of strategies for either firm, it is important to identify clearly not only the players’ moves but also the order in which these moves are chosen and the information available to players when they make decisions. An effective way of organizing this information is by using a game tree. A game tree will depict a path of play in addition to the players, actions, outcomes and payoffs. The game tree for the software game, thus, appears as follow:

Payoffs:

(Macrosoft, Microcorp)

enter

b

simple

slick

Macrosoft

Microcorp

enter

stay out

stay out

Microcorp

a

c

Macrosoft has two strategies: choose slick or choose simple. Microcorp, however, has four strategies since there are two nodes to consider, b and c, and two possible actions at each node, enter or stay out. These strategies are:

Choose to enter regardless of which campaign Macrosoft chooses (enter, enter).

Choose to enter if Macrosoft chooses slick, otherwise choose to stay out (enter, stay out).

Choose to stay out if Macrosoft chooses simple and vice versa (stay out, enter).

Choose to stay out in both cases when Macrosoft chooses slick or simple (stay out, stay out).

Table 2 shows the strategic form of the game:

Table 2: Strategic form of the software game

Macrosoft

slick simple

(- $250, $380)

($100*,$400*)

(- $250, $380)

($0, $800*)

($0*, $430*)

($100*, $400)

($0*, $430)

($0, $800*)

(Payoffs in $1, 000s)

(enter, enter)

Microcorp

(enter, stay out)

(stay out, enter)

(stay out, stay out)

There are two pure strategy Nash equilibria to this game which are {slick, (stay out, enter)} and {simple, (enter, enter)}. These are the optimal outcomes of the game as no player would wish to deviate from his chosen strategy given the other's choice. However, the question is which of these equilibria is more reasonable. The best outcome can be found through a procedure called backward induction. This process assumes that players act rationally at each node. This means that they will act in their own best interests. Knowing this, a player working to solve a game tree can confidently remove suboptimal actions to his rivals until only the most likely path remains. By doing this, an opponent's possible moves from the initial node to the payoff can be depicted; allowing the player to devise a strategy for each of those probable moves and eventually finds the equilibrium. The software game can thus be solved using this method of reasoning:

Figure 2: Game tree of the software game

($380, 000, -$250, 000)

($430, 000, $0): A

($400, 000, $100, 000): B

($800, 000, $0)

Payoffs:

(Macrosoft, Microcorp)

slick

c

enter

Microcorp

b

stay out

Macrosoft

enter

a

Microcorp

simple

stay out

At node b, entering the market gives Microcorp a loss of $250, 000, while staying out gives it a zero-payoff. Therefore, Microcorp would rationally choose to stay out. Similarly, the possibility that Microcorp will stay out at node c can be eliminated since its payoff for enter is higher than that for stay out. Therefore, of the four strategies available to Microcorp, backward reasoning indicates that its only optimal strategy is to choose enter at node b and stay out node c.

By pruning the non-optimal moves from Microcorp's decision nodes, Macrosoft's choices now look as follows:

Figure 3: The new game tree of software game

Payoffs:

(Marcrosoft, Microcorp)

($430, 000, $0)

($400, 000, $100, 000)

simple

slick

Macrocorp

Macrosoft's optimal strategy is obvious-choosing slick as this yields a payoff of $430, 000 instead of $400, 000 from adopting simple campaign. Therefore, by looking ahead and taking its opponent's entry decision into account Macrosoft can avoid making a mistake of $30, 000. Consequently, the strategy profile – {slick, (stay out, enter)} is called the sub-game perfect equilibrium (SPNE); it is also a Nash equilibrium (NE) of the game. Since backward induction holds that players will play their optimal action at each decision node, the resulting strategies will thus lead to a NE. However, it is important to note that a NE is not always a SPNE. In particular, the other NE of the software game – {simple, (enter, enter)} is not a SPNE. This is because it violates the rules of backward induction which assumes that Microcorp would never choose enter at node b.

On the other hand, Microcorp may want to arrive at the NE – {simple, (enter, enter)}. Since Microcorp prefers outcome B of ($400, 000, $100, 000) to outcome A ($430, 000, $0) (figure 2), but it cannot get there unless Macrocorp adopts the simple strategy. Microcorp may, therefore, threaten to always choose enter. If Macrosoft were to believe the threat, it would believe that it would earn only $380, 000 by choosing slick and $400, 000 by choosing simple. However, Microcorp's threat to enter is not credible and Macrosoft knows that once it chooses slick, Microcorp will choose stay out regardless of its commitment as stay out is simply its best move at node b. In this case, Macrosoft has the advantage by becoming the first mover and can therefore induce its rival to stay out of the market. While Microcorp suffers the disadvantages of a second mover unless it could credibly commit to always adopt the strategy (enter, enter).

Figure 3: Centipede game

I II I Payoffs to (I, II)

(8, 19)

(0, 0) (-1, 10) (9, 9)

Effective as it is, backward induction has revealed some limitations. One of these has been disclosed in the well-known centipede game. Figure 3 illustrates the game in which two players alternate in choosing between stopping and continuing the game. If a player stops the game, each will receive a zero payoff at that point. But if a player chooses to continue, he is fined £1 while the other is rewarded with £10. The game continues until one of the players stops or both reach the final payoffs of £8 and £19 respectively.

Go

Go

Stop

Stop St

Stop

Go

Backward induction suggests that player I should stops the game at the very first move and gets a zero payoff. Suppose that the game has reached the final decision node where player I makes the last move. At this point, player I has to choose between Stop and Go. The only rational choice here is to stop and pocket £9 rather than deciding to continue and receiving a less payoff of £8. This means that at the previous decision node, player II will choose to stop the game, taking into account that player I, who is rational, responds by choosing Stop on the next move. This in turn implies that player I, at the first decision node, now effectively considers between Stop and receiving a zero payoff or Go and losing £1 when player II rationally stops the game at the succeeding node. Player I, therefore, should stop the game immediately. This outcome is the unique SPNE. However, it would be better if player I continues the game until he can get £9 by stopping at the penultimate node, or, as a second best, until the final round where he gets £8. The question is that if player I, in practice, really chooses to stop the game at the first decision node.

Experimental evidence by Kelvey and Palfrey (1992) and El-Gamul et al. (1993) shows that the logic of backward reasoning is seldom followed by decision-makers. In particular, in a four-legged centipede game experimented by Kelvey and Palfrey, only 7% of players stopped the game at the very first move with a maximum payoff of $6. 40 at its head. When the payoff was increased to $25. 60, 15% chose Stop at the first decision node. Even at the final node, only 69% of players (in the high-payoff centipede) and 85% of players (in the low-payoff centipede) chose to end the game.

In conclusion, the rationale of applying backward induction seems strong since it can help narrow the number of possible Nash equilibria. By looking forward and reasoning backward, each player can predict what other players will do at subsequent stages of the game. Therefore, he can judge the consequences of his possible moves, assuming that players are rational, and therefore; decides on the optimal move. However, backward induction exhibits some limitations as discussed in the centipede game where the argument rests on the prediction of behaviour off the equilibrium path. This arguably leads to the challenge of rationality assumption of game theory which needs further justification.

Word count: 1499