Greedy Algorithm Application in Big Two Card Game

Alexander Jason – 13521100
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13521100@std.stei.itb.ac.id

***Abstract*—The application of greedy algorithms in Big Two card game provides an effecient approach to decision-making and strategic gameplay. In Big Two, players aim to play cards from their hand to beat the current card on the table. The greedy algorithm focuses on selecting the best strategy according to the number of cards in the player's hand. When there is still a lot of cards, prioritize selecting lowest card with lowest card in the hand that can beat the table card and vice versa. This strategy can maximizing the player's chances of winning each round. Despite its limitations, the greedy algorithm provides strategic planning in the Big Two card game. By considering the available card combinations and selecting the lowest card to beat the table card, players can enhance their chances of success in each round. The abstract concludes by acknowledging the significance of the greedy algorithm in providing a systematic and strategic framework for gameplay in the Big Two card game.**

***Keywords*—Big Two card-game, greedy, solutions**

I. Introduction

Card games are one of the most favorite games in the world. There are a lot of variation of card games, such as Chinese Poker, Bridge, BlackJack, Solitaire, and many more. In this paper, writer will discuss more about Big Two. Big Two, also known as Chinese Poker or Cap Sa, is an exciting and strategic card game that has gained huge popularity in various parts of the world. It is a intense-type game that requires both skill and tactics, making it a favorite among card enthusiasts and casual gamers. With its simple rules and challenging gameplay, Big Two offers hours of entertainment and intense competition for players of all skill levels. Big Two use Poker card to play the game. Poker card itself consists of 52 cards with 4 symbols (diamonds♦, clubs♣, hearts♥, and spades♠). Each symbol consists of 13 cards from number 2 to 10, Jack, Queen, King, and Ace. Usually, Ace has the highest value in poker, but for Big Two, the highest value is number 2 itself. Each symbol also has its power where Spade♠ is the highest, followed by heart♥, then club♣, and lastly diamond♦.

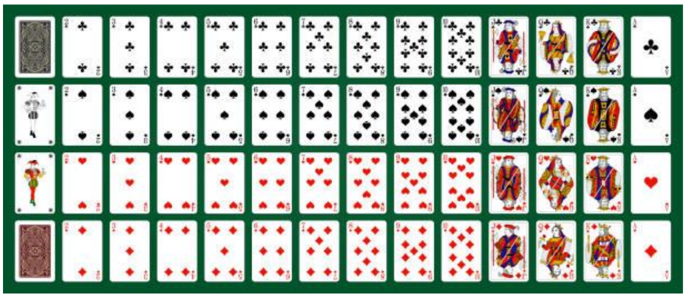


Fig. 1 One Pack of Poker Cards (source: https://www.istockphoto.com/id/foto-foto/kartu-remi)

In Big Two, players aim to be the first to get rid of their cards by playing combinations that beat the previously played hand. The game is played with a standard deck of 52 cards, and the ranking of the cards follows the traditional hierarchy, with the deuces (twos) being the highest-ranking cards. Each player starts with an equal number of cards and takes turns playing valid combinations, such as pairs, three-of-a-kind, straight flushes, and more. The strategic aspect of the game lies in timing the use of powerful combinations to outsmart opponents and gain an advantage. With the blend of luck and skill, Big Two provides an addictive and thrilling gaming experience that keeps players engaged from start to finish.

The rule is simple, player who have the smallest value of card (3♦ Diamonds) will start the first round. Player can play their card in the following ways:

- Single Cards (1 card)
- Pairs (2 cards)
- Three of a Kind (3 cards)
- Poker Hands (5 cards)

Other players must beat the card from the previous player with the same type of combination. For example, if the round is lead with pair of 3, the next player must beat it with a higher pair, such as pair of 4, etc.

II. Fundamental Theorem

A. Greedy Algorithm

Greedy Algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't really matter whether the current best result will bring the most optimal result. Greedy never looks back to earlier decisions even if the choice was wrong.

Greedy algorithm consists of these elements:

1. Candidates set, C: List of candidates that will be chosen in every step (e.g. node in graph, task, coins, things, etc.)
2. Solutions set, S: List of chosen candidates
3. Solution function: determine that list of candidates have already given solution
4. Selection function: choose candidate based on specific greedy heuristic strategy
5. Feasibility function: check if the candidate that has been chosen can be put into solutions set
6. Objective function: maximize or minimize

Greedy algorithm works with a top-down approach. Because of their simplicity, greedy algorithms are frequently straightforward and efficient. Greedy algorithms make a sequence of choices, each being in some way the best at that time. As its progresses, each choice involves taking a step towards to build a solution to the problem.

In general, greedy works with this scheme:

1. Initialize S with an empty value.
2. Select an element (a candidate solution) using a selection function from C.
3. Remove the selected element from C.
4. If the element is eligible to be included in the solution set, add it to the solution set. Ignore elements that are not eligible.
5. If the solution set has provided the desired solution (or if the candidate set is empty), the algorithm will terminate. If not, proceed to iterate to step 2.

B. Big Two Card Game

The basis of Big Two is a race to get rid of your cards. It supports 3 or 4 players with one deck. Each player will be dealt 13 cards in each game, and can play these cards in four different ways:

- Singles (just one card)
- Pairs (two cards of matching values)
- Triplets or three of a kind (three cards of matching values)
- Poker Hands (five cards forming a straight, flush, full house, four of a kind or straight flush)

The game consists of a number of hands, each consisting of a number of rounds. The player who has the 3 of Diamonds starts and must make a play involving this card to begin the first round. For instance, they could just play the 3 of Diamonds as a single, or if they had a straight involving this 3, they could play with that.

Whichever type of play is led, the next player clockwise around the table must play a higher card (or combination of cards) of the same type. For instance, if the player with the 3 of Diamonds plays a pair of 3's to start the round, the next player must play a pair of a higher value.

Players can choose to pass if they don't want to play, and must pass if they cannot play. When all other players have passed, the last player to successfully make a play has possession and can begin a new round with whatever play they wish.

Whatever type of play begins a round, all subsequent plays must be of the same play. For example, if a player begins a round with a straight, the next player must play a higher straight or a better poker hand (e.g. a flush, full house, four of a kind or a straight flush). The game ends when there is only one player left.

Big Two is a card game that use poker card or rummy card to play with. The game is called Big Two because the highest card to play is a 2, which makes the order of values in this games goes from 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A, 2. Furthermore, the symbols are ranked in the following order (from bottom to top): Diamonds♦, Clubs♣, Hearts♥, and Spades♠. Therefore, the lowest card in the game is the 3 of Diamonds♦, and the highest card is the 2 of Spades♠. For example: In a round of singles, if a player leads the King of Clubs, the next player can follow with a higher ranked King (King of Hearts or Spades), or a higher ranked card of any symbol (an Ace or a Two).

There five types of playable Poker Hand combination. In ascending order they are: straight, flush, full house, four of a kind, and straight flush.

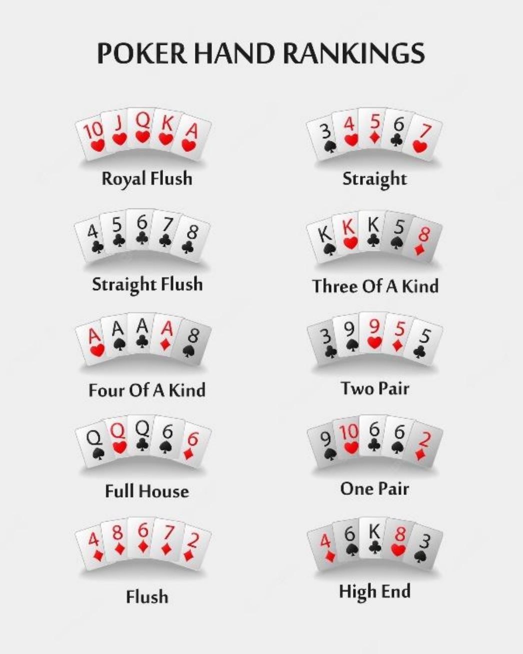


Fig. 2 Poker Hand Type (source: https://www.freepik.com/premium-vector/poker-hand-ranking-combinations_2386537.htm)

1. **Straight**

Consists of five cards of consecutive value with mixed symbols. A straight with higher ranks beats a lower one, irrespective of the symbols of the cards. When the ranks are the same, the symbol of the highest card determines which is higher. For example, K♦-Q♠-J♣-10♥-9♥ beats Q♥-J♦-10♦-9♦-8♠. Twos do not rank high in straights, but below the 3, so the highest straight is A-K-Q-J-10 including the ace of spades.

2. **Flush**

   Consists of any five cards of the same symbols. A flush in a higher symbol beats a flush in a lower symbol, irrespective of the ranks of the cards. Between two flushes in the same symbol, the one with the higher top card is better. For example, 9♥-7♥-6♥-5♥3♥ beats 2♣-J♣-9♣-6♣-4♣

3. **Full House**

   Consists of three cards of one rank and two of another rank; between two full houses, the one whose triple is of higher rank is better. For example 10-10-10-K-K beats 8-8-8-3-3.

4. **Four of a kind**

   Made up of all four cards of one rank, plus any fifth card. The fifth card must be included, four equal cards by themselves are not a playable combination. Between two fours of a kind, the value of the four cards determines which is higher.

5. **Straight Flush**

   Consists of five consecutive cards of the same symbols. Basically, it's a combination of straight and flush. The rank of the highest card determines which of two straight flushes is higher; between two equal ranked straight flushes, the one in the higher symbol is better, so the Royal Flush in spades A♠-K♠-Q♠-J♠-10♠ is the highest straight flush and the best five-card combination.

III. Implementation

Big Two need 3-4 players that will be distributed 13 cards for each player. Greedy algorithm will help player to build strategy to win the game. In this paper, the writer will use greedy implementation to determine the next step with greedy by *value*, greedy by *combinations*, and greedy by *number of cards in player hands*.

Players can also use greedy to form their card combination. First, map the greedy elements:

1. Candidates set, C: Set of cards from a player
2. Solutions set, S: List of cards combination
3. Solution function: determine that list of candidates have already fulfilled the combination rule.
4. Selection function: choose cards to form combinations
5. Feasibility function: check if the candidate that has been chosen to have fulfilled the combination rule.
6. Objective function: minimize count of combinations, and maximize count of cards in each combinations

Then sort all 13 cards in the players' hand. Here is the logic implementation for checking combinations in player's hand.

```
# Function to generate all combinations of cards

def generate_combinations(start_cards):

    # Convert start_cards to a set=

       start_cards_set = set(start_cards)

       process_cards = start_cards


    # Generate all possible combinations of cards

 all_combinations = []

 while(process_cards):

   cards ={}

     # check possible combination

   if(checkPlayerStraightFlush (process_cards)[0]):

      cards = checkPlayerStraightFlush (process_cards)[1]

      start_cards_set.add(cards)

      process_cards.remove(cards)

     continue

   if(checkPlayerFourOfAKind (process_cards)[0]):

      cards = checkPlayerFourOfAKind (process_cards)[1]

      start_cards_set.add(cards)

      process_cards.remove(cards)

      continue

  if(checkPlayerFullHouse (process_cards)[0]):

      cards = checkPlayerFullHouse(process_cards)[1]

      start_cards_set.add(cards)

      process_cards.remove(cards)

      continue

  if(checkPlayerFlush (process_cards)[0]):

      cards = checkPlayerFlush(process_cards)[1]

      start_cards_set.add(cards)
      process_cards.remove(cards)
       continue
if(checkPlayerStraight
(process_cards)[0]):
               cards                    =
checkPlayerStraight(process_cards)[1]
               start_cards_set.add(cards)

process_cards.remove(cards)
               continue
if(checkPlayerThreeOfAKind(
process_cards)[0]):
       cards  =  checkPlayerThreeOfAKind(
process_cards)[1]
       start_cards_set.add(cards)
       process_cards.remove(cards)
       continue
if(checkPlayerPair(process_cards)[0]):
       cards = checkPlayerPair
   (process_cards)[1]
       start_cards_set.add(cards)
       process_cards.remove(cards)
       continue
# if there is no combination
process_cards.clear()
```

Code 1. Greedy Combination Card Implementation (source: writer's archive)

A. *Greedy By Value*

Greedy by value strategy is focusing to play the highest or lowest card value in the hand when the player doesn't go first. The combination does not matter, because this strategy is to play based on table card type (like pair, single, etc.).

1. Candidates set, C: Combination of cards from a player
2. Solutions set, S: one list of cards to play
3. Solution function: determine that list of candidates have higher value than the card on the table.
4. Selection function: choose cards to play
5. Feasibility function: check if the candidate that has been chosen to have higher value than the card on the table.
6. Objective function: play the highest/lowest card in hand

Here is the logic implementation in python:

```
Greedy Lowest Value
def play_card(hand, table_card):
    # Sort the hand in ascending order
    sorted_hand = sorted(hand)


    # Find the lowest card that beats the
table card
    for card in sorted_hand:
        if card > table_card:
            return card
    # If no card beats the table card,
return None to Pass
    return None
```

```
Greedy Highest Value
def play_card(hand, table_card):
    # Sort the hand in ascending order
    sorted_hand = sorted(hand)


    # Find the lowest card that beats the
table card
    for card in sorted_hand:
        if card > table_card:
            return card


    # If no card beats the table card,
return None to Pass
    return None
```

Code 2. Greedy by Value Implementation (source: writer's archive)

B. *Greedy By Combination*

Similar to greedy by value, greedy by combination is focusing to which card combination type (like straight, flush, etc.) to play when the player goes for the first round.

1. Candidates set, C: Combination of cards from a player
2. Solutions set, S: one list of cards to play
3. Solution function: determine that list of candidates is fulfill the combination rule
4. Selection function: choose cards to play
5. Feasibility function: check if the candidate that has been chosen is fulfill the combination rule
6. Objective function: play the highest/lowest card type in hand

Here is the prioritization in ascending order: single, pair, three of a kind, straight, flush, full house, four of a kind, straight flush.

C. Greedy By Number of Cards in Hand

This strategy is a combination of both previous strategies. Greedy By Number of Cards in Hand is a dynamic greedy algorithm where we prioritize both previous strategies based on the number of the card in hand. If player have more than 7 cards in hand, player use greedy by lowest value and lowest combination strategy. If there are less than 7 cards, player use greedy by highest value and highest combination strategy.

1. Candidates set, C: Combination of cards from a player
2. Solutions set, S: one list of cards to play
3. Solution function: determine that list of candidates is fulfill the combination rule
4. Selection function: choose cards to play
5. Feasibility function: check if the candidate that has been chosen is fulfill the combination rule
6. Objective function: play the lowest card type and value in hand if numbers of card more than 7, and vice versa

IV. Game Analysis

Let's analyze a game that using Greedy by Number of Cards in Hand strategy. This game is played on https://chipo.edwinzhou.com/ website. Here is the player hand:

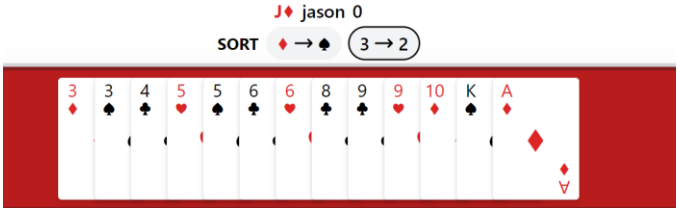


Fig. 3. Jason's Deck

First, we need to find all possible combinations in Jason's deck using greedy algorithm. After using greedy algorithm, here is the optimal combinations by greedy.

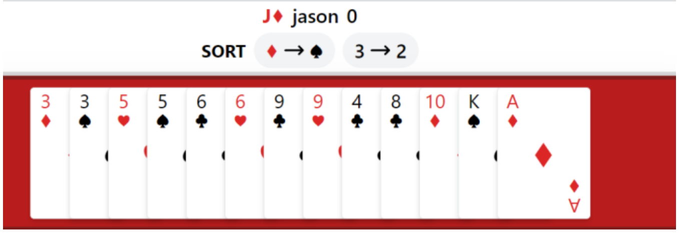


Fig. 4 Jason's Deck after using Greedy Algorithm

From the figure above, the deck has been sorted from highest type to lower. Now to begin the round because Jason's deck has 3♦, the player will go first.

With greedy, player will play pair of 3 first because it is the only way to play. Player can't play only 3♦ because it is a pair combination and cannot be discarded to single.

After the player play pair of 3, the other player starting play their card. Bot2 play pair of 7, bot 3 play pair of queens and bot4 pass the round. Player can't play other card because player don't have any higher pair, so player pass.

Then new round start. The last card that has been played is J♠. According to greedy algorithm, the player will choose the lowest card in hand but still higher than the card in the table. So Player will play K♠.

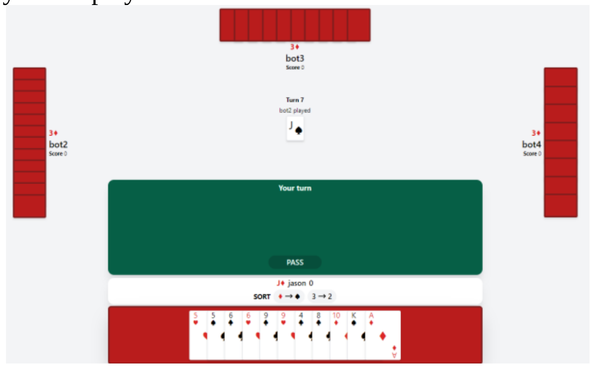


Fig. 5 2nd Round

All the other players pass, and Jason get to play the first card. Because card in hands still more than 7, Jason will play lowest combination with the lowest value, which is 4♣

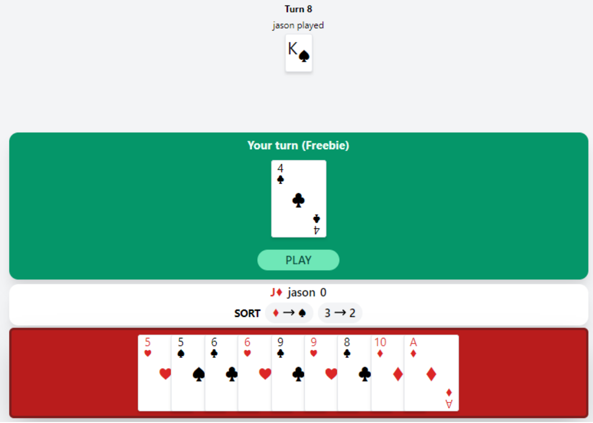


Fig. 6. Jason's Turn

After certain turns, here is the remaining cards

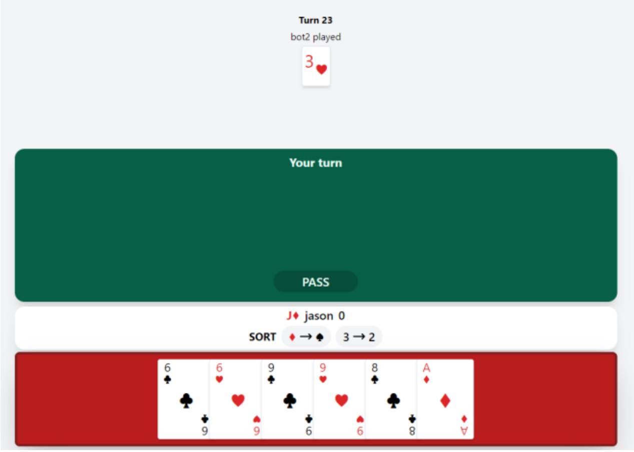
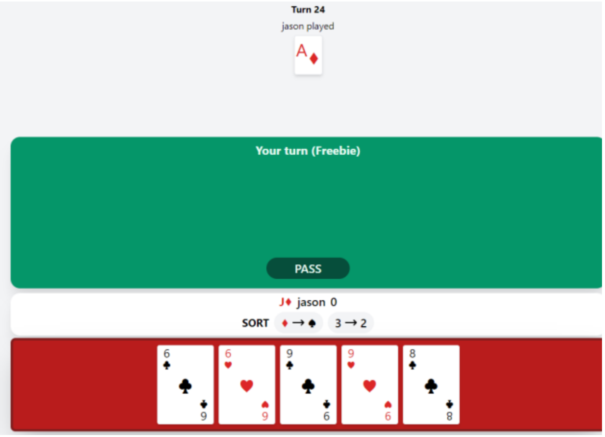


Fig. 7. Jason's Turn

Because the remaining cards is less than 7, Jason will play highest value of card, which is A♦.



Turn 24

jason played

Your turn (Freebie)

Fig. 8 Jason's Round

The remaining cards is less than 7, to begin the round Jason will play highest value of card of highest combination, which is pair of 9. Other players pass because they don't have pair, but bot3 beat Jason with pair of 2 and win.

From this game, player Jason can win some rounds, but still lose at the end of the game. This can be caused by some factors such as luck and greedy prioritization.

After doing some other games, greedy can bring player to winnings, this conclude that greedy can be an optimal strategy for Big Two but there is also some other factor such as luck and other players strategies.

V. Conclusion

In conclusion, the greedy algorithm can be applied effectively in Big Two card game. By employing a greedy strategy, players can make decisions based on current benefits and prioritize playing the higher value cards to beat the current card on the table.

The idea behind the greedy approach in Big Two is if there is still lot of cards in player's hand, prioritize to select the lowest card in the hand that can beat the card on the table. Other way, prioritize to select the highest card in the hand if there are few numbers of cards in player's hand. This ensures that the player maximizes their chances of winning each round by playing a card that is both valid and strategically advantageous.

However, it is important to note that the greedy strategy does not consider long-term planning or potential future moves. It focuses solely on the immediate situation and maximizing the value of the current play. While this can lead to short-term optimal solution, it may not always result in the most optimal overall game strategy.

Video Link at YouTube

Here is the link for the video: https://youtu.be/ii-7Fa4eprc

Acknowledgment

The writer would like to thank all IF2211 lecturers especially Dr. Nur Ulfa Maulidevi, S.T., M.Sc as lecturer in first class IF2211 for Strategy Algorithm, for teaching and supporting students to write these paper. I have gained much better understanding in combinatorics and its application. I also would like to thank Dr. Ir. Rinaldi, M.T, who provided students with plenty of resources on Strategy Algorithm at the website.

References

[1] Munir, R. (2022). *Greedy Bagian 1*. Informatika Rinaldi Munir. https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf. Accesed on May 21, 2023

[2] Munir, R. (2022). *Greedy Bagian 11.* Informatika Rinaldi Munir. https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag2.pdf, Accesed on May 21, 2023

[3] How, W. (2021). *How to play big two: 10 steps (with pictures)*. wikiHow. https://www.wikihow.com/Play-Big-Two. Accesed on May 22, 2023

[4] Rules, G. (2021, December 7). *Big two*. Game Rules. https://gamerules.com/rules/big-two-card-game/. Accesed on May 22, 2023

[5] Programiz, P. (2020). *Greedy algorithm*. Programiz. https://www.programiz.com/dsa/greedy-algorithm. Accesed on May 22, 2023

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2023

Alexander Jason 13521100