Solving two-person zero-sum sequential games via efficient computation of Nash equilibria: proof-of-concept on Kuhn's 3-card Poker

DOHMATOB Elvis [1,2]

[1]Parietal Team, INRIA
[2]Université Paris-Sud

Some necessary history ...

- Blaise Pascal, and other giants ($\sim$ 1623)...
- Zemelo 1913 (subgame perfect equilibrium, backward induction)
- Von Neumann and Oskar Morgenstern (Cooperative Games, *Theory of Games and Economic Behavior, 1944*),
- John F. Nash (Non-cooperative games, Nash Equilibrium)
- L. Shapley (concept of value for bargaining games, etc.),
- Tucker, Kuhn, and co. (equivalence between LCP and problem of computation of Nash Equilibria; subgame perfect equilibrium again).

Some necessary history ...

- Blaise Pascal, and other giants ($\sim$ 1623)...
- Zemelo 1913 (subgame perfect equilibrium, backward induction)
- Von Neumann and Oskar Morgenstern (Cooperative Games, *Theory of Games and Economic Behavior, 1944*),
- John F. Nash (Non-cooperative games, Nash Equilibrium)
- L. Shapley (concept of value for bargaining games, etc.),
- Tucker, Kuhn, and co. (equivalence between LCP and problem of computation of Nash Equilibria; subgame perfect equilibrium again).

Some necessary history ...

- Blaise Pascal, and other giants ($\sim$ 1623)...
- Zemelo 1913 (subgame perfect equilibrium, backward induction)
- Von Neumann and Oskar Morgenstern (Cooperative Games, *Theory of Games and Economic Behavior, 1944*),
- John F. Nash (Non-cooperative games, Nash Equilibrium)
- L. Shapley (concept of value for bargaining games, etc.),
- Tucker, Kuhn, and co. (equivalence between LCP and problem of computation of Nash Equilibria; subgame perfect equilibrium again).

Some necessary history ...

- Blaise Pascal, and other giants ($\sim$ 1623)...
- Zemelo 1913 (subgame perfect equilibrium, backward induction)
- Von Neumann and Oskar Morgenstern (Cooperative Games, *Theory of Games and Economic Behavior, 1944*),
- John F. Nash (Non-cooperative games, Nash Equilibrium)
- L. Shapley (concept of value for bargaining games, etc.),
- Tucker, Kuhn, and co. (equivalence between LCP and problem of computation of Nash Equilibria; subgame perfect equilibrium again).

Some necessary history ...

- Blaise Pascal, and other giants ($\sim$ 1623)...
- Zemelo 1913 (subgame perfect equilibrium, backward induction)
- Von Neumann and Oskar Morgenstern (Cooperative Games, *Theory of Games and Economic Behavior, 1944*),
- John F. Nash (Non-cooperative games, Nash Equilibrium)
- L. Shapley (concept of value for bargaining games, etc.),
- Tucker, Kuhn, and co. (equivalence between LCP and problem of computation of Nash Equilibria; subgame perfect equilibrium again).

Some necessary history ...

- Blaise Pascal, and other giants ($\sim$ 1623)...
- Zemelo 1913 (subgame perfect equilibrium, backward induction)
- Von Neumann and Oskar Morgenstern (Cooperative Games, *Theory of Games and Economic Behavior, 1944*),
- John F. Nash (Non-cooperative games, Nash Equilibrium)
- L. Shapley (concept of value for bargaining games, etc.),
- Tucker, Kuhn, and co. (equivalence between LCP and problem of computation of Nash Equilibria; subgame perfect equilibrium again).

Problem

Given a sequential two-person zero-sum game (e.g Texas Hold'em Poker), construct an optimal player.

Two-person zero sum Sequential games

- There are 3 players: nature (player 0), Alice (player 1), and Bob (player 2).
- Players take turns in making moves
- Sets of moves of any two distinct players don't overlap (i.e $A_p \cap A_q = \emptyset$ if $p \neq q$)

Two-person zero sum Sequential games

- There are 3 players: nature (player 0), Alice (player 1), and Bob (player 2).
- Players take turns in making moves
- Sets of moves of any two distinct players don't overlap (i.e $A_p \cap A_q = \emptyset$ if $p \neq q$)

Two-person zero sum Sequential games

- There are 3 players: nature (player 0), Alice (player 1), and Bob (player 2).
- Players take turns in making moves
- Sets of moves of any two distinct players don't overlap (i.e $A_p \cap A_q = \emptyset$ if $p \neq q$)

Two-person zero sum Sequential games

- The game tree T (from Alice's perspective) is defined as follows. T has a set $V(T)$ of nodes (aka vertices) and a set $E(T)$ of edges.
- Each node v belongs to a single player, $p(v)$, called "the player to act at v"; $p(v) := 3$ if v is a leaf node. At node v, the player $p(v)$ has a set $C(v) \subset A_p$ of possible moves.
- For each player $p \in \{0, 1, 2\}$, the set of all nodes at which p acts is called the nodes of p, denoted $V_p(T)$.
- There are two kinds of nodes: the leaf nodes $L(T)$, at which the game must end and decision nodes $D(T)$, at which the player to act must make a move. For example in Poker, a leaf node is reached at countdown, or when a player folds, or when the player runs out of money.
- There is a special node root(T) defined by $root(T) := (/, p_0)$,
- In accordance with the rules of the game, every other node $y \in V(T) - \{root(T)\}$ is of the form $y = v.(c, p(v))$

Two-person zero sum Sequential games

- The game tree T (from Alice's perspective) is defined as follows. T has a set $V(T)$ of nodes (aka vertices) and a set $E(T)$ of edges.
- Each node v belongs to a single player, $p(v)$, called "the player to act at v"; $p(v) := 3$ if v is a leaf node. At node v, the player $p(v)$ has a set $C(v) \subset A_p$ of possible moves.
- For each player $p \in \{0, 1, 2\}$, the set of all nodes at which p acts is called the nodes of p, denoted $V_p(T)$.
- There are two kinds of nodes: the leaf nodes $L(T)$, at which the game must end and decision nodes $D(T)$, at which the player to act must make a move. For example in Poker, a leaf node is reached at countdown, or when a player folds, or when the player runs out of money.
- There is a special node root(T) defined by $root(T) := (/, p_0)$,
- In accordance with the rules of the game, every other node $y \in V(T) - \{root(T)\}$ is of the form $y = v.(c, p(v))$

Two-person zero sum Sequential games

- The game tree T (from Alice's perspective) is defined as follows. T has a set $V(T)$ of nodes (aka vertices) and a set $E(T)$ of edges.
- Each node v belongs to a single player, $p(v)$, called "the player to act at v"; $p(v) := 3$ if v is a leaf node. At node v, the player $p(v)$ has a set $C(v) \subset A_p$ of possible moves.
- For each player $p \in \{0, 1, 2\}$, the set of all nodes at which p acts is called the nodes of p, denoted $V_p(T)$.
- There are two kinds of nodes: the leaf nodes $L(T)$, at which the game must end and decision nodes $D(T)$, at which the player to act must make a move. For example in Poker, a leaf node is reached at countdown, or when a player folds, or when the player runs out of money.
- There is a special node root(T) defined by $root(T) := (/, p_0)$,
- In accordance with the rules of the game, every other node $y \in V(T) - \{root(T)\}$ is of the form $y = v.(c, p(v))$

Two-person zero sum Sequential games

- The game tree T (from Alice's perspective) is defined as follows. T has a set $V(T)$ of nodes (aka vertices) and a set $E(T)$ of edges.
- Each node v belongs to a single player, $p(v)$, called "the player to act at v"; $p(v) := 3$ if v is a leaf node. At node v, the player $p(v)$ has a set $C(v) \subset A_p$ of possible moves.
- For each player $p \in \{0, 1, 2\}$, the set of all nodes at which p acts is called the nodes of p, denoted $V_p(T)$.
- There are two kinds of nodes: the leaf nodes $L(T)$, at which the game must end and decision nodes $D(T)$, at which the player to act must make a move. For example in Poker, a leaf node is reached at countdown, or when a player folds, or when the player runs out of money.
- There is a special node root(T) defined by $root(T) := (/, p_0)$,
- In accordance with the rules of the game, every other node $y \in V(T) - \{root(T)\}$ is of the form $y = v.(c, p(v))$

Two-person zero sum Sequential games

- The game tree T (from Alice's perspective) is defined as follows. T has a set $V(T)$ of nodes (aka vertices) and a set $E(T)$ of edges.
- Each node v belongs to a single player, $p(v)$, called "the player to act at v"; $p(v) := 3$ if v is a leaf node. At node v, the player $p(v)$ has a set $C(v) \subset A_p$ of possible moves.
- For each player $p \in \{0, 1, 2\}$, the set of all nodes at which p acts is called the nodes of p, denoted $V_p(T)$.
- There are two kinds of nodes: the leaf nodes $L(T)$, at which the game must end and decision nodes $D(T)$, at which the player to act must make a move. For example in Poker, a leaf node is reached at countdown, or when a player folds, or when the player runs out of money.
- There is a special node root(T) defined by $root(T) := (/, p_0)$,
- In accordance with the rules of the game, every other node $y \in V(T) - \{root(T)\}$ is of the form $y = v.(c, p(v))$

Two-person zero sum Sequential games

- The game tree T (from Alice's perspective) is defined as follows. T has a set $V(T)$ of nodes (aka vertices) and a set $E(T)$ of edges.
- Each node v belongs to a single player, $p(v)$, called "the player to act at v"; $p(v) := 3$ if v is a leaf node. At node v, the player $p(v)$ has a set $C(v) \subset A_p$ of possible moves.
- For each player $p \in \{0, 1, 2\}$, the set of all nodes at which p acts is called the nodes of p, denoted $V_p(T)$.
- There are two kinds of nodes: the leaf nodes $L(T)$, at which the game must end and decision nodes $D(T)$, at which the player to act must make a move. For example in Poker, a leaf node is reached at countdown, or when a player folds, or when the player runs out of money.
- There is a special node root(T) defined by $root(T) := (/, p_0)$,
- In accordance with the rules of the game, every other node $y \in V(T) - \{root(T)\}$ is of the form $y = v.(c, p(v))$

Strategy profiles, simplexes and complexes

Definition

- The *n-simplex*, denoted Δ_n, is the subset of $\mathbb{R}^{n+1}$ defined by

$$\Delta_n := \left\{x \in \mathbb{R}^{n+1} \mid x_i \geq 0 \ \forall i, \text{ and } \sum_{i=1}^{n+1} x_i = 1\right\} \quad (1)$$

- Each kink $\delta_i := (0, ..., 0, 1, 0, ..., 0)$ of Δ_n represents a *pure-strategy*. Any non-kink point x of Δ_n represents a *mixed-strategy*. Thus Δ_n is simply the *convex hall* of $(n+1)$-dimensional pure strategies δ_i.
- A *complex* is an abstract *polyhedron* whose kinks are themselves simplexes. Points on a complex correspond to *realization plans*.

Strategy profiles, simplexes and complexes

Definition

- The *n-simplex*, denoted Δ_n, is the subset of $\mathbb{R}^{n+1}$ defined by

$$\Delta_n := \left\{ x \in \mathbb{R}^{n+1} \mid x_i \geq 0\ \forall i, \text{ and } \sum_{i=1}^{n+1} x_i = 1 \right\} \tag{1}$$

- Each kink $\delta_i := (0, ..., 0, 1, 0, ..., 0)$ of Δ_n represents a *pure-strategy*. Any non-kink point x of Δ_n represents a *mixed-strategy*. Thus Δ_n is simply the *convex hall* of $(n+1)$-dimensional pure strategies δ_i.
- A *complex* is an abstract *polyhedron* whose kinks are themselves simplexes. Points on a complex correspond to *realization plans*.

Strategy profiles, simplexes and complexes

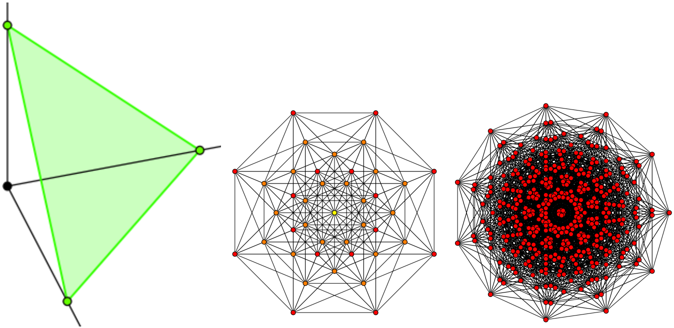
Definition

- The *n-simplex*, denoted Δ_n, is the subset of $\mathbb{R}^{n+1}$ defined by

$$\Delta_n := \left\{x \in \mathbb{R}^{n+1} \mid x_i \geq 0\ \forall i,\ \text{and}\ \sum_{i=1}^{n+1} x_i = 1\right\} \tag{1}$$

- Each kink $\delta_i := (0, ..., 0, 1, 0, ..., 0)$ of Δ_n represents a *pure-strategy*. Any non-kink point x of Δ_n represents a *mixed-strategy*. Thus Δ_n is simply the *convex hall* of $(n+1)$-dimensional pure strategies δ_i.
- A *complex* is an abstract *polyhedron* whose kinks are themselves simplexes. Points on a complex correspond to *realization plans*.

Simplexes and complexes (examples of)



Left: 2-simplex **Middle**: 7-simplex **Right**: 10-simplex

Sequences and payoff matrix

Definition

Given a node $t \in V(T)$, and a player $p \in \{0, 1, 2\}$, let $\sigma_p(t)$ be the sequence of player p's moves along the path from the root node to t.

- Let S_p be the sequences of moves for player p. Then any $\sigma \in S_p$ is either the empty sequence set or can be written as $\sigma_{h_p} c$ where h_p is an information set of p and c is a choice at h_p.
- Precisely,

$$S_p = \{\emptyset\} \cup \{\sigma_h c \mid h \in H_p, c \in C_h\} \tag{2}$$

- The payoff matrix $A = (a_{\sigma,\tau})$ by

$$a_{\sigma,\tau} := \sum_{\text{leaf } t:\ \sigma_1(t)=\sigma, \sigma_2(t)=\tau} \beta_0(\sigma_0(t))a(t) \tag{3}$$

Sequences and payoff matrix

Definition

Given a node $t \in V(T)$, and a player $p \in \{0, 1, 2\}$, let $\sigma_p(t)$ be the sequence of player p's moves along the path from the root node to t.

- Let S_p be the sequences of moves for player p. Then any $\sigma \in S_p$ is either the empty sequence set or can be written as $\sigma_{h_p} c$ where h_p is an information set of p and c is a choice at h_p.
- Precisely,

$$S_p = \{\emptyset\} \cup \{\sigma_h c \mid h \in H_p, c \in C_h\} \tag{2}$$

- The payoff matrix $A = (a_{\sigma,\tau})$ by

$$a_{\sigma,\tau} := \sum_{\text{leaf } t:\ \sigma_1(t)=\sigma, \sigma_2(t)=\tau} \beta_0(\sigma_0(t))a(t) \tag{3}$$

Sequences and payoff matrix

Definition

Given a node $t \in V(T)$, and a player $p \in \{0, 1, 2\}$, let $\sigma_p(t)$ be the sequence of player p's moves along the path from the root node to t.

- Let S_p be the sequences of moves for player p. Then any $\sigma \in S_p$ is either the empty sequence set or can be written as $\sigma_{h_p}c$ where h_p is an information set of p and c is a choice at h_p.
- Precisely,

$$S_p = \{\emptyset\} \cup \{\sigma_h c \mid h \in H_p, c \in C_h\} \tag{2}$$

- The payoff matrix $A = (a_{\sigma,\tau})$ by

$$a_{\sigma,\tau} := \sum_{\text{leaf } t\text{: } \sigma_1(t)=\sigma, \sigma_2(t)=\tau} \beta_0(\sigma_0(t))a(t) \tag{3}$$

Example of sequence-form representation

- $S_1 = [[], [(\{'chance' : ('A',), 'choices' : ('(B,2)', '(F,3)'), 'sigma' : ''\}, '(B,2)')], [(\{'chance' : ('A',), 'choices' : ('(B,2)', '(F,3)'), 'sigma' : ''\}, '(F,3)')], [(\{'chance' : ('K',), 'choices' : ('(B,2)', '(F,3)'), 'sigma' : ''\}, '(B,2)')], [(\{'chance' : ('K',), 'choices' : ('(B,2)', '(F,3)'), 'sigma' : ''\}, '(F,3)')]]$
- $I_1 = \{(('A',), '', ('(B,2)', '(F,3)')) : ['(/,0).(w,1)', '(/,0).(x,1)'], (('K',), '', ('(B,2)', '(F,3)')) : ['(/,0).(u,1)', '(/,0).(v,1)']\}$
- $E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \end{pmatrix}$, $e = (1, 0, 0)$,

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Example of sequence-form representation

- $S_1 = [[], [(\{'chance' : ('A',), 'choices' : ('(B, 2)', '(F, 3)'), 'sigma' : ''\}, '(B, 2)')], [(\{'chance' : ('A',), 'choices' : ('(B, 2)', '(F, 3)'), 'sigma' : ''\}, '(F, 3)')], [(\{'chance' : ('K',), 'choices' : ('(B, 2)', '(F, 3)'), 'sigma' : ''\}, '(B, 2)')], [(\{'chance' : ('K',), 'choices' : ('(B, 2)', '(F, 3)'), 'sigma' : ''\}, '(F, 3)')]]$
- $I_1 = \{(('A',), '', ('(B, 2)', '(F, 3)')) : ['(/, 0).(w, 1)', '(/, 0).(x, 1)'], (('K',), '', ('(B, 2)', '(F, 3)')) : ['(/, 0).(u, 1)', '(/, 0).(v, 1)']\}$
- $E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \end{pmatrix}, e = (1, 0, 0),$

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- $S_1 = [[],[(\{'chance':('A',),'choices':('(B,2)','(F,3)'),'sigma':''\},'(B,2)')],[(\{'chance':('A',),'choices':('(B,2)','(F,3)'),'sigma':''\},'(F,3)')],[(\{'chance':('K',),'choices':('(B,2)','(F,3)'),'sigma':''\},'(B,2)')],[(\{'chance':('K',),'choices':('(B,2)','(F,3)'),'sigma':''\},'(F,3)')]]$
- $I_1 = \{(('A',),'',('(B,2)','(F,3)')):['(/,0).(w,1)','(/,0).(x,1)'],(('K',),'',('(B,2)','(F,3)')):['(/,0).(u,1)','(/,0).(v,1)']\}$
- $E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \end{pmatrix}$, $e = (1,0,0)$,

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Kuhn's Poker sequence-form representation

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \end{pmatrix},$$

$$e = (1, 0, 0, 0, 0, 0, 0)$$

NE minimax and best-response in sequential-games

Thanks to the sequence-form representation, we have:

- NE problem for Alice and Bob is the the *saddle-point problem*

$$\underset{x \in Q_1}{maximize}\ \underset{y \in Q_2}{minimize}\ y^T A x, \tag{4}$$

where:

$$\left.\begin{array}{l} Q_1 := \{x \in \mathbb{R}^n_+ \mid Ex = e\} \text{ is Alice's complex, and} \\ Q_2 := \{y \in \mathbb{R}^m_+ \mid Fy = f\} \text{ is Bob's complex.} \end{array}\right\} \tag{5}$$

A saddle-point (x^*, y^*) corresponds to a NE for the game.

- Given a fixed behavioural strategy $y_0 \in Q_2$ for Bob, a *best-response* behavioural strategy x_0 for Alice [1] is a solution to the LCP

$$\underset{x \in Q_1}{maximize}\ y_0^T A x \tag{6}$$

[1] Of course, there is an analogous concept for Bob.

NE minimax and best-response in sequential-games

Thanks to the sequence-form representation, we have:

- NE problem for Alice and Bob is the the *saddle-point problem*

$$\underset{x \in Q_1}{\text{maximize}}\ \underset{y \in Q_2}{\text{minimize}}\ y^T A x, \tag{4}$$

  where:

$$\left.\begin{array}{l} Q_1 := \{x \in \mathbb{R}^n_+ \mid Ex = e\} \text{ is Alice's complex, and} \\ Q_2 := \{y \in \mathbb{R}^m_+ \mid Fy = f\} \text{ is Bob's complex.} \end{array}\right\} \tag{5}$$

  A saddle-point (x^*, y^*) corresponds to a NE for the game.

- Given a fixed behavioural strategy $y_0 \in Q_2$ for Bob, a *best-response* behavioural strategy x_0 for Alice [1] is a solution to the LCP

$$\underset{x \in Q_1}{\text{maximize}}\ y_0^T A x \tag{6}$$

[1] Of course, there is an analogous concept for Bob.

NE computation Kuhn's Poker

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \end{pmatrix},$$

$$e = (1, 0, 0, 0, 0, 0, 0)$$

Practical considerations

In practice, the payoff matrix A will be very huge (for example $18x10^9$ rows by $18x10^9$ columns for Texas Hold'em!) but sparse, with a nice block structure

$$A = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix} \tag{7}$$

where (for some "small" (worst case $\sim$ 100 by 100) sparse matrices F_1, ..., B_2, ..., S, W):

$$\left.\begin{aligned} A_1 &:= F_1 \otimes B_1 \\ A_2 &:= F_2 \otimes B_2 \\ A_3 &:= F_3 \otimes B_3 + S \otimes W, \end{aligned}\right\} \tag{8}$$