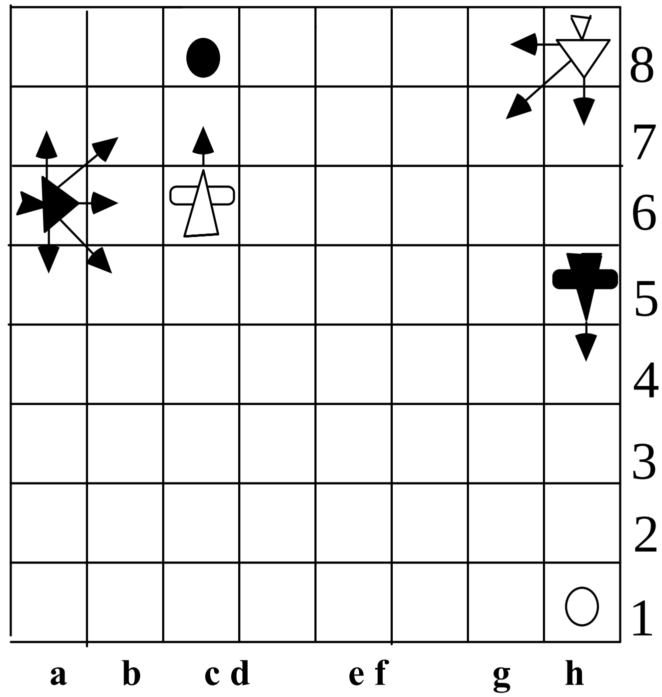
From Search to Construction
Part II
Construction of Strategies

(more details can be found in Chapter 13 of the book on LG)

Optimization problem for autonomous Aerospace robotic vehicles with serial alternating motions

2D/4A Problem



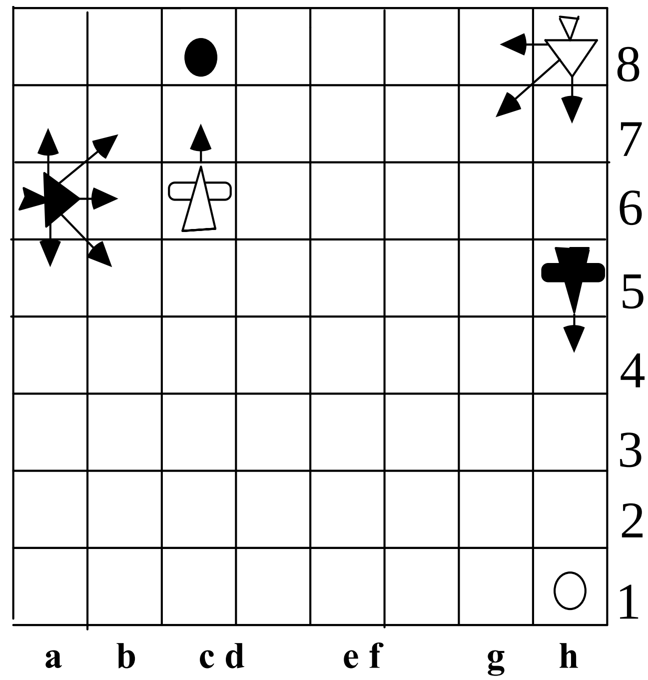
Is there a strategy for the White to make a draw?

The specific question is as follows.
Is there an optimal strategy that provides one of the following?

1. **Both BOMBERs hit their targets on subsequent time increments and stay safe for at least one time increment.**

2. **Both BOMBERs are destroyed before they hit their targets or immediately after that.**

2D/4A Problem: Terminal Sets



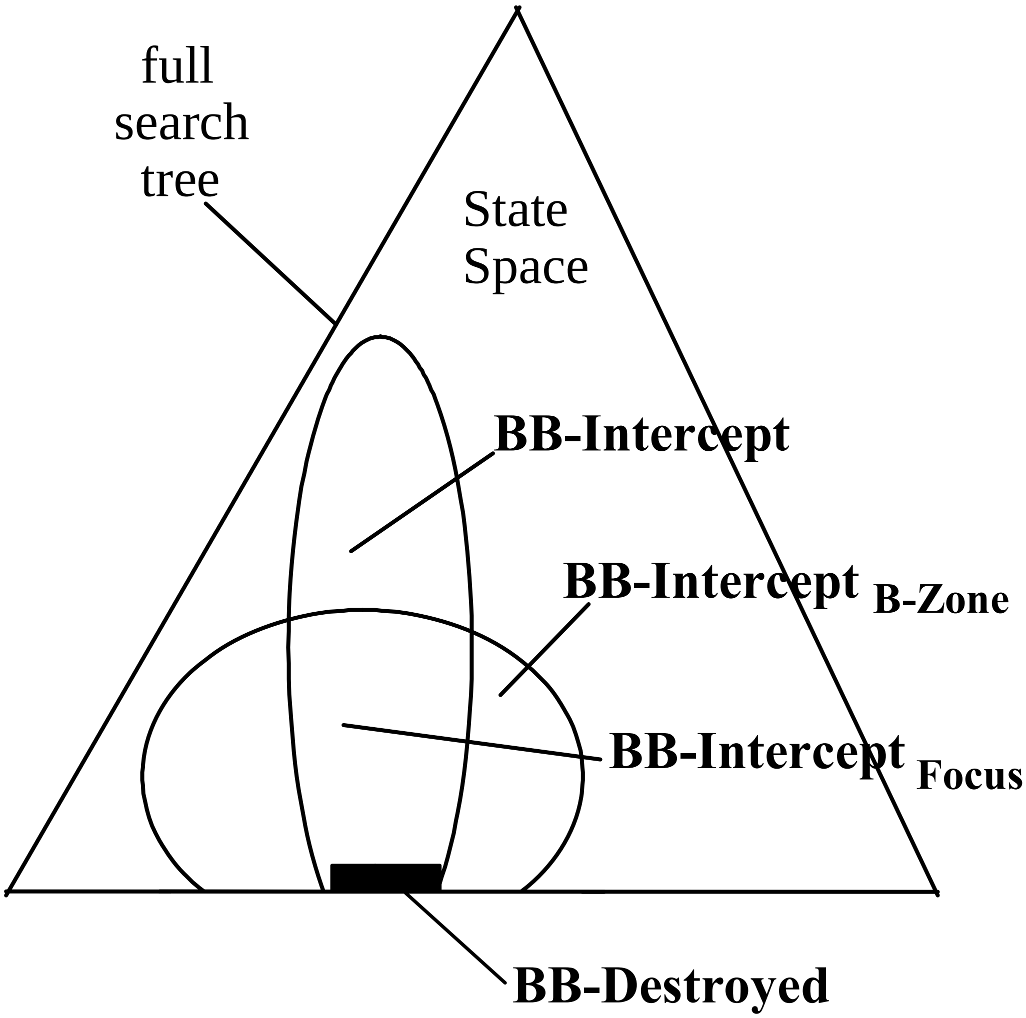
1. **W-Win** = **BB-Destroyed** ∩ **WB-Safe**

2. **B-Win** = **WB-Destroyed** ∩ **BB-Safe**

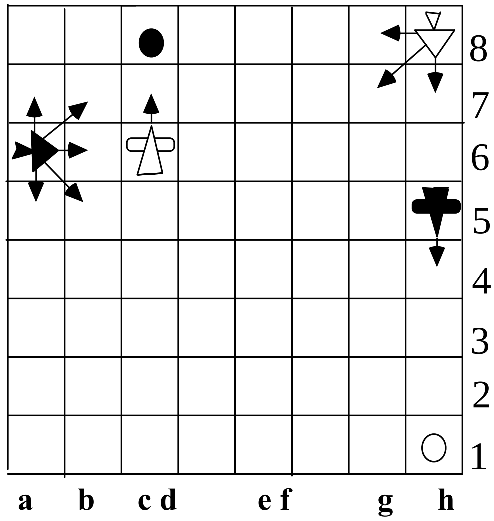
3. **Draw** = **Safe** ∪ **Destroyed**, where
Destroyed = **BB-Destroyed** ∩ **WB-Destroyed**,
Safe = **BB-Safe** ∩ **WB-Safe**

Let A be a set of states.
The strategy is called an A strategy if it is represented by the optimal subtree with the terminal nodes which represent states from A, only.

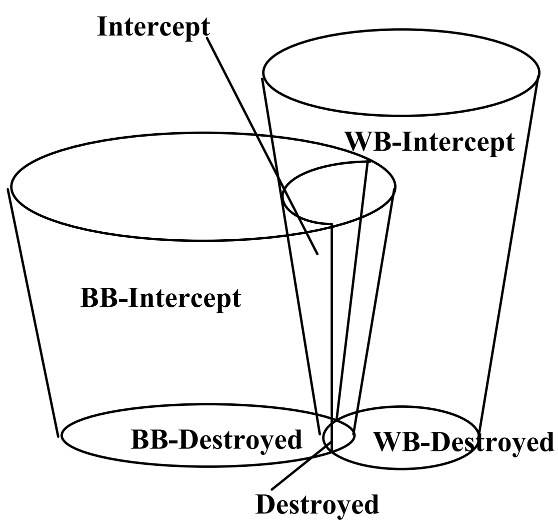
Why do we need the Terminal States Expansion?



Terminal Sets Expansion

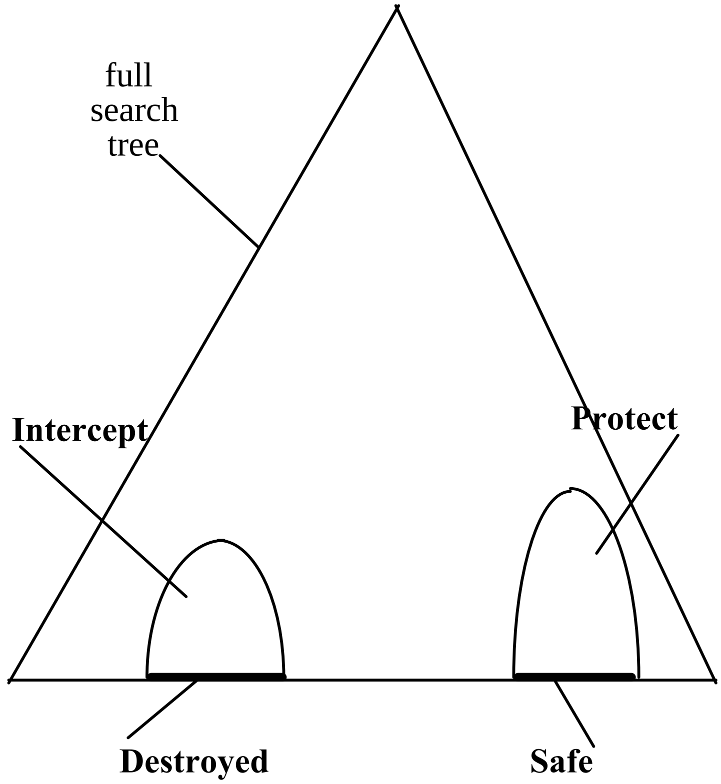


BB-Intercept is the set of states where BB-Destroyed strategy exists



Intercept = BB-Intercept $\cap$WB-Intercept
Destroyed = BB-Destroyed $\cap$ WB-Destroyed.
Destroyed $\subset$ DrawExpand.

Expanded Terminal States



Terminal Sets Expansion

Intercept = **BB-Intercept** $\cap$ **WB-Intercept**

is the set of states where the **Destroyed strategy** exists
Destroyed = **BB-Destroyed** $\cap$ **WB-Destroyed**.
Intercept $\subset$ **DrawExpand**.

Protect = **BB-Protect** $\cap$ **WB-Protect**

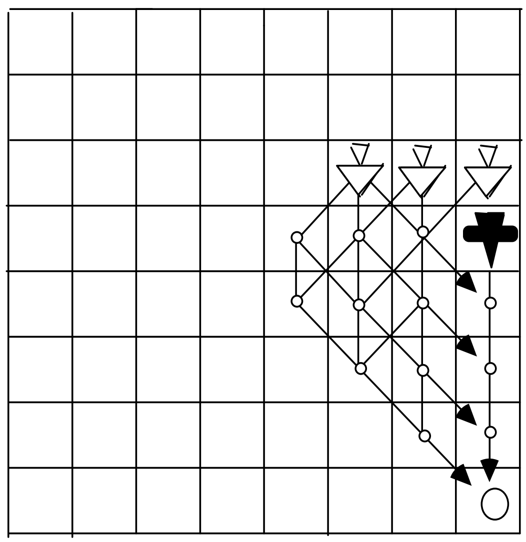
is the set of states where the **Safe strategy** exists
Safe = **BB-Safe** $\cap$ **WB-Safe**.
Protect $\subset$ **DrawExpand**

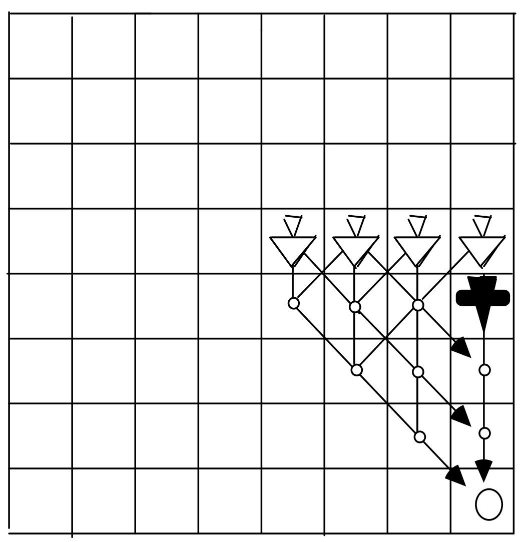
Intercept $\cup$ **Protect** $\subset$ **DrawExpand**.

Intercept $\cup$ **Protect** $\neq$ **DrawExpand**.

Structure of Expanded Terminal Sets

BB-Intercept$_{\text{B-Zone}}$

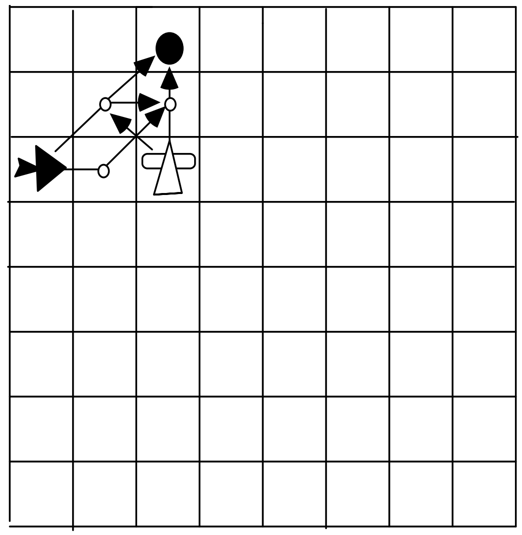


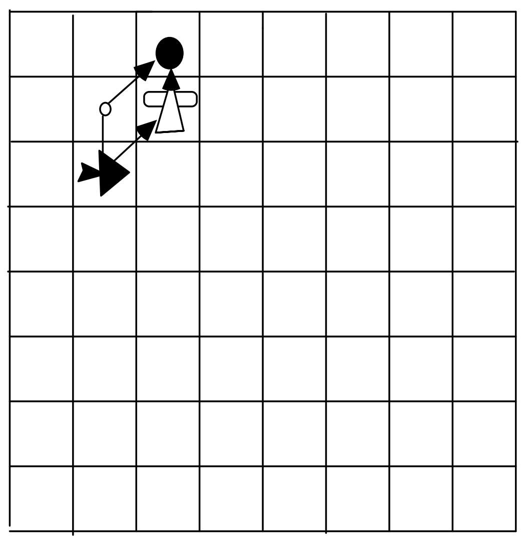


$$Z_{BB} = t(\mathrm{BB}, t_{BB}, l_1)\, t(\mathrm{WF}, t_{WF}, l_2)$$

Structure of Expanded Terminal Sets

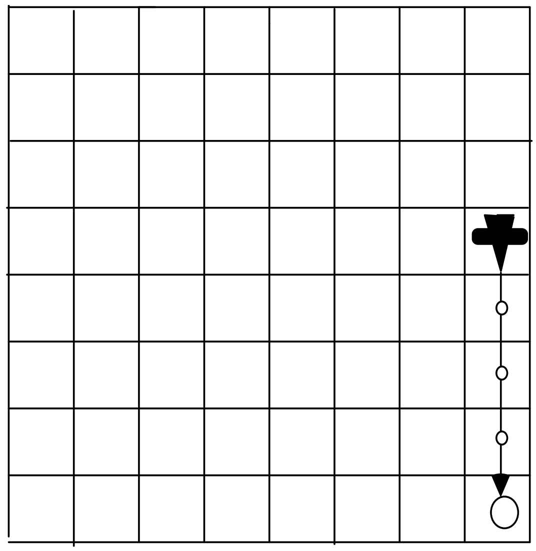
WB-Intercept$_{\text{W-Zone}}$

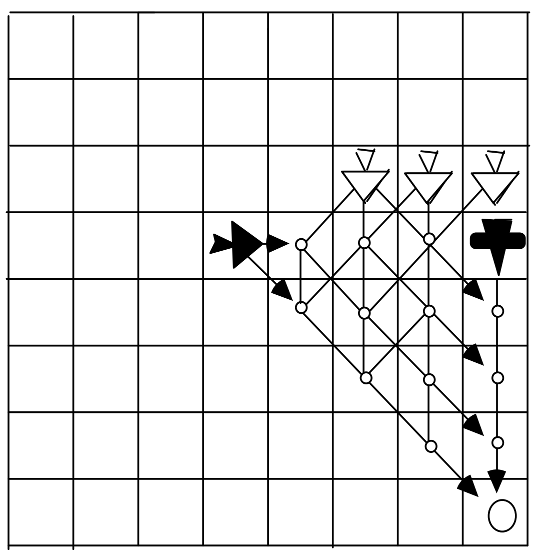




Structure of Expanded Terminal Sets

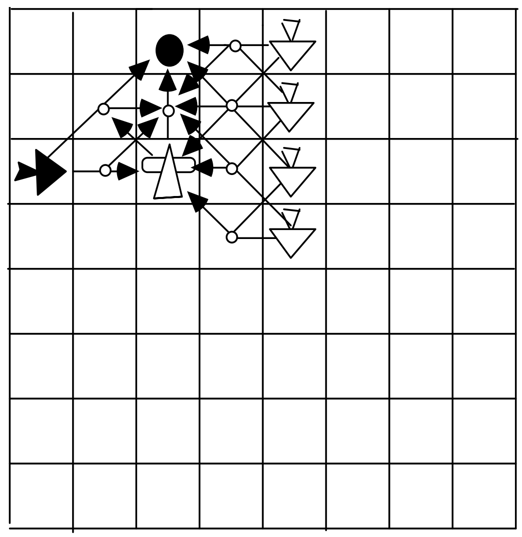
BB-Protect$_{B\text{-}Zone}$

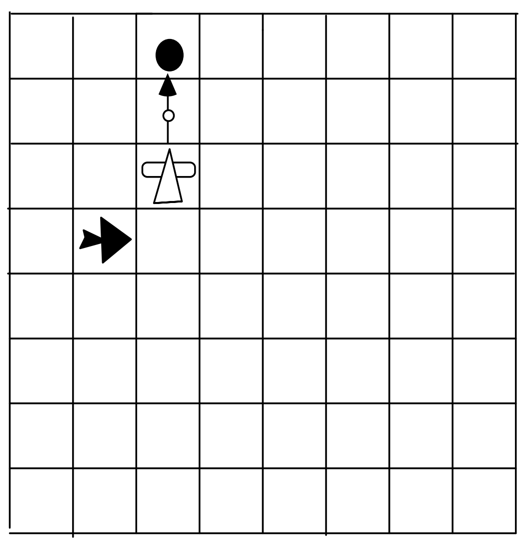




A Structure of Expanded Terminal Sets

WB-Protect$_{\text{W-Zone}}$

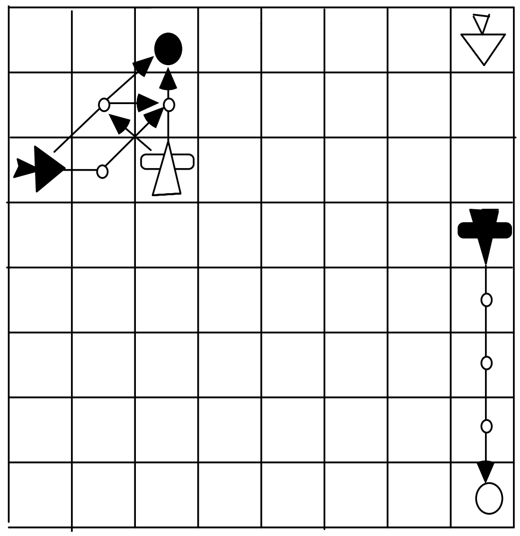


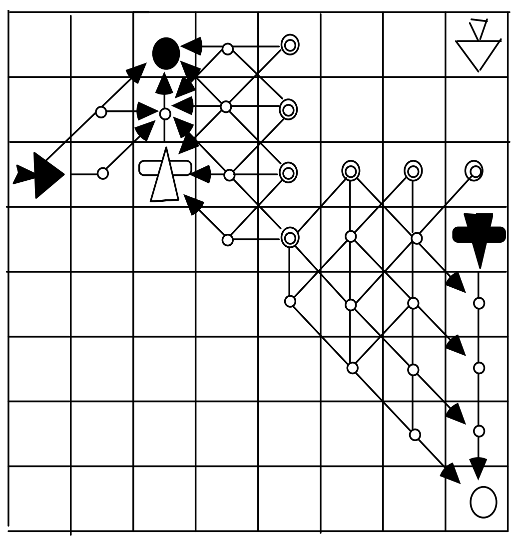


State Space Decomposition

$$\textbf{SPACE} = \textbf{BB-Intercept}_{\textbf{B-Zone}} \cup \textbf{BB-Protect}_{\textbf{B-Zone}}$$

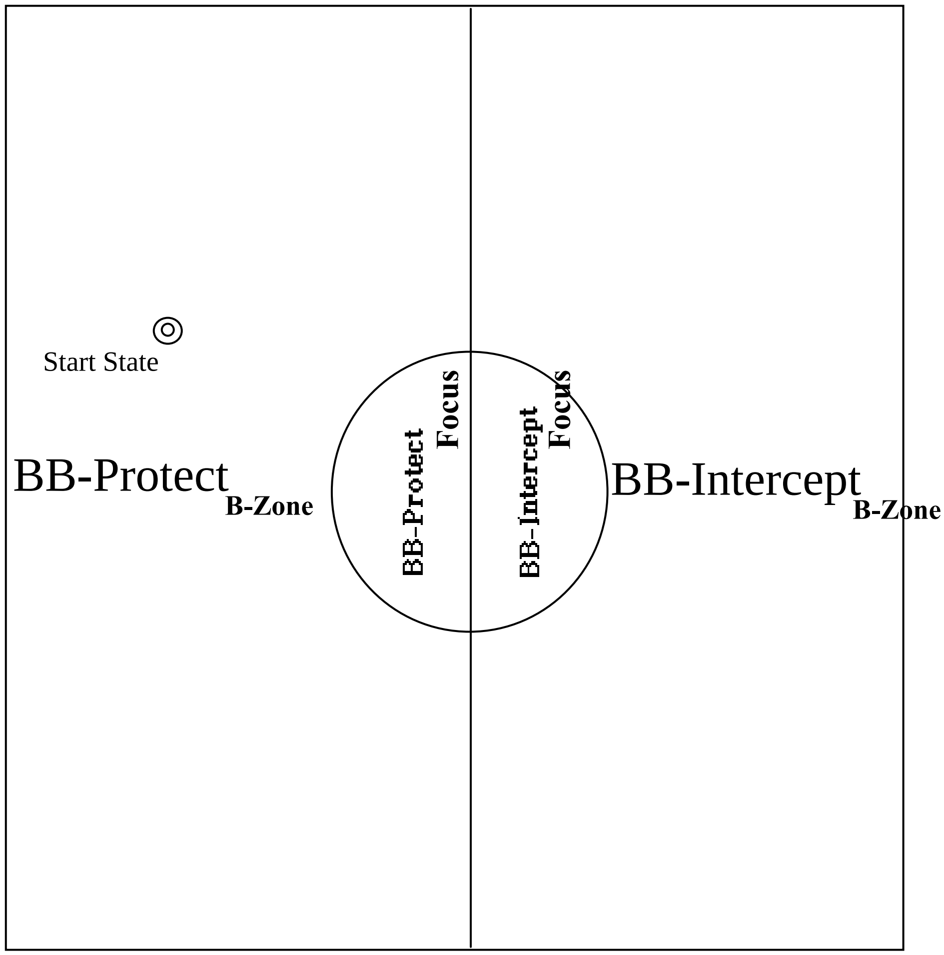
$$\textbf{SPACE} = \textbf{WB-Intercept}_{\textbf{W-Zone}} \cup \textbf{WB-Protect}_{\textbf{W-Zone}}$$





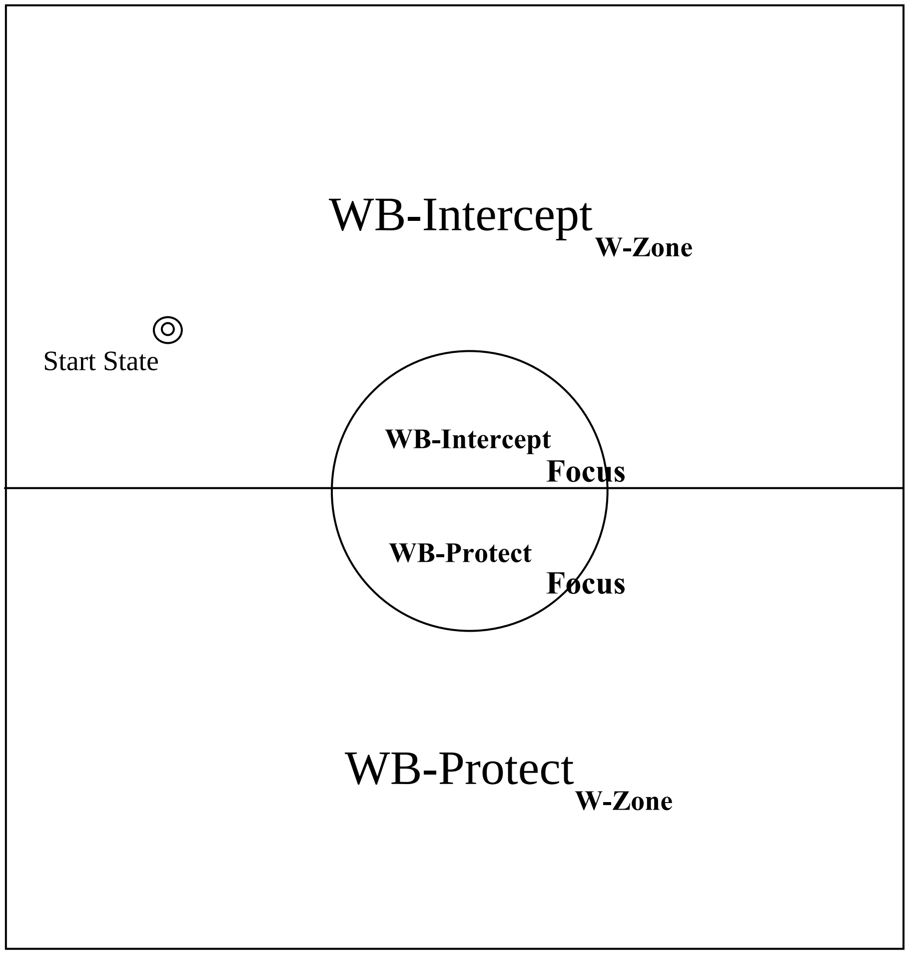
State Space Decomposition

$$\textbf{SPACE} = \textbf{BB-Protect}_{\textbf{B-Zone}} \cup \textbf{BB-Intercept}_{\textbf{B-Zone}}$$

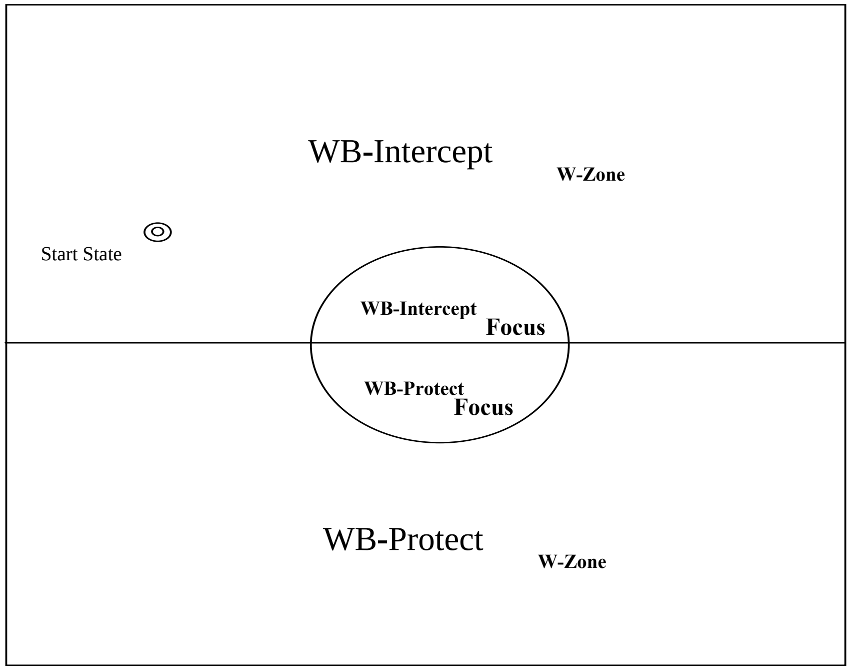


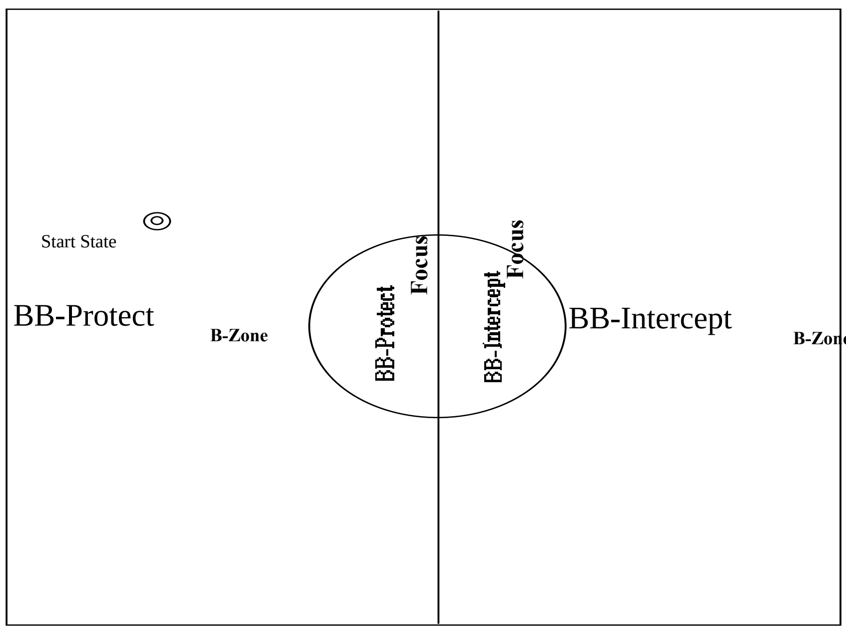
State Space Decomposition

$$\textbf{SPACE} = \textbf{WB-Protect}_{\textbf{W-Zone}} \cup \textbf{WB-Intercept}_{\textbf{W-Zone}}$$

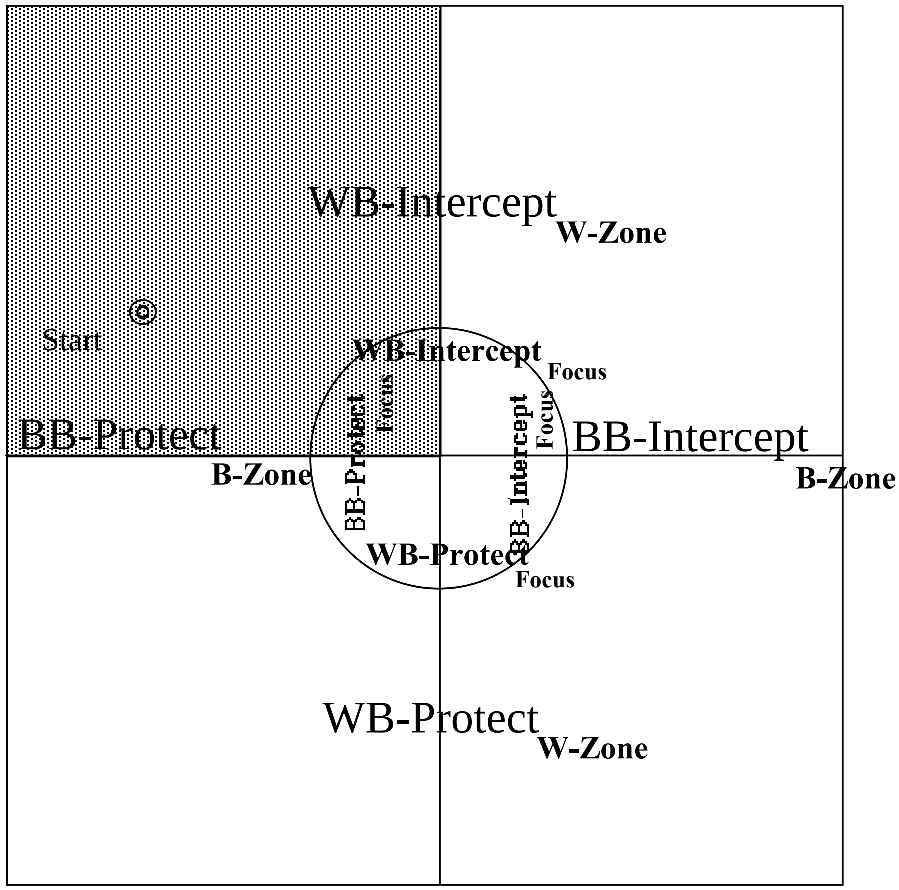


State Space Decomposition

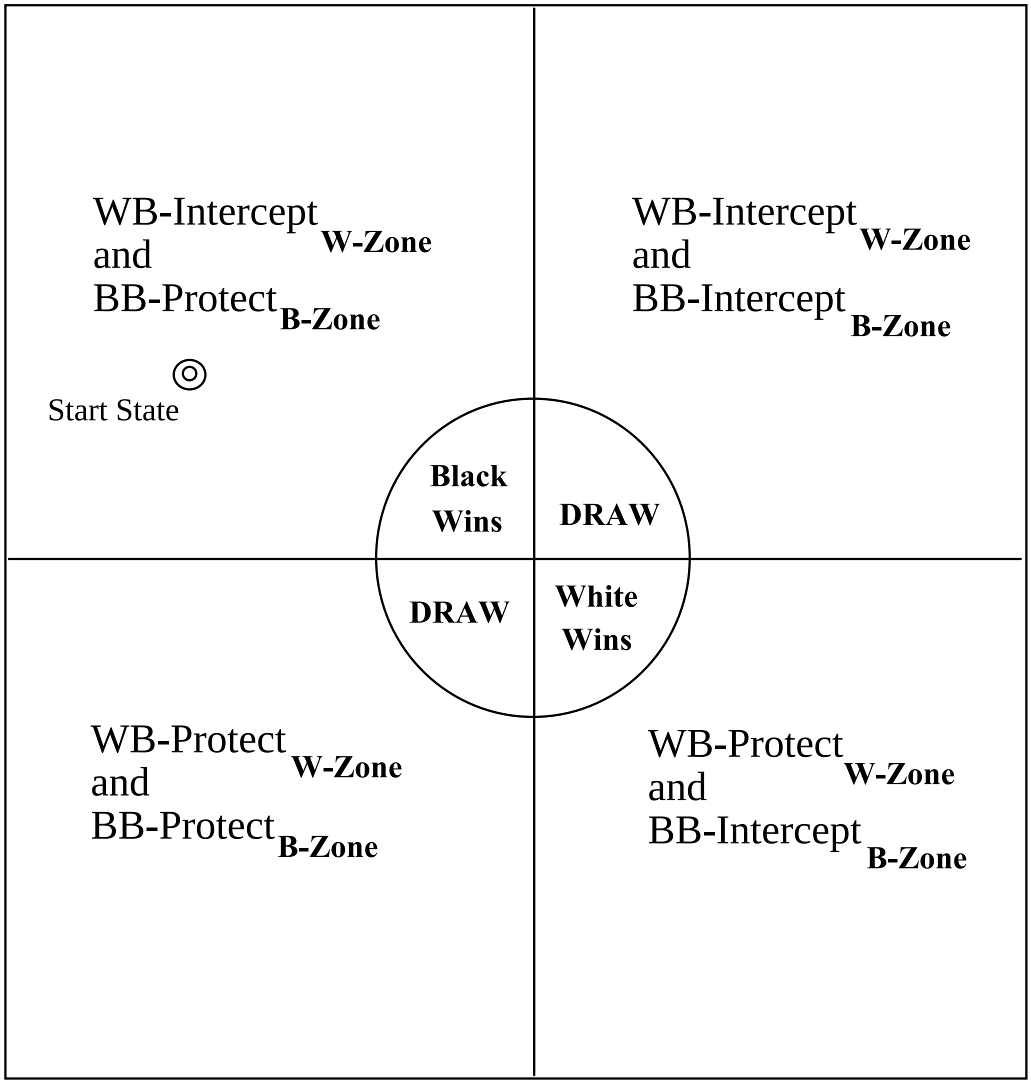




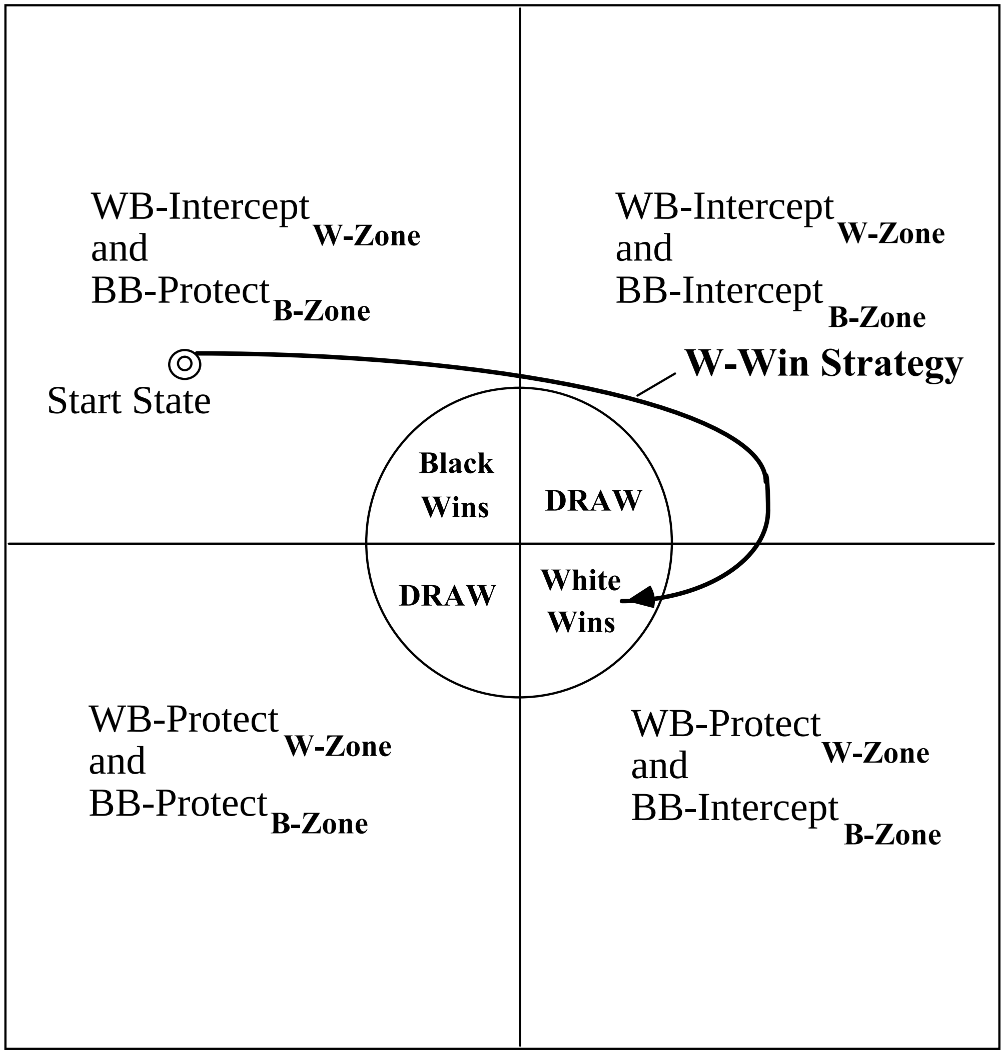
State Space Decomposition
Apply both decompositions simultaneously



State Space Chart



Intend-to-Win Strategies



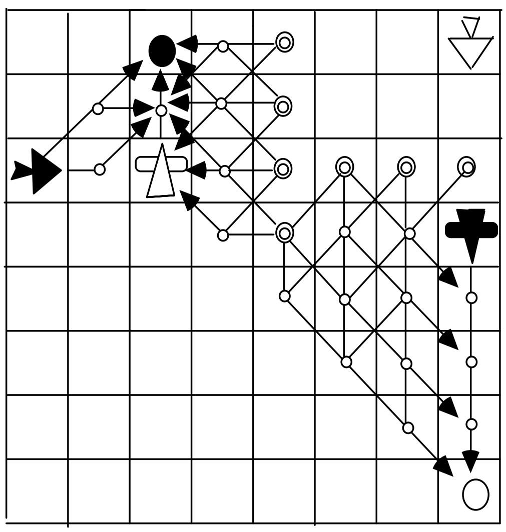
Intend-to-Win Strategies

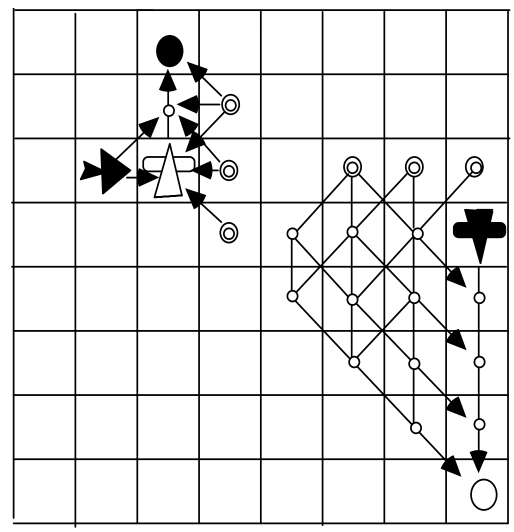
In reality, ***only one of them takes place.***

1. W-Win strategy: W-Win = WB-Safe ∩ BB-Destroyed

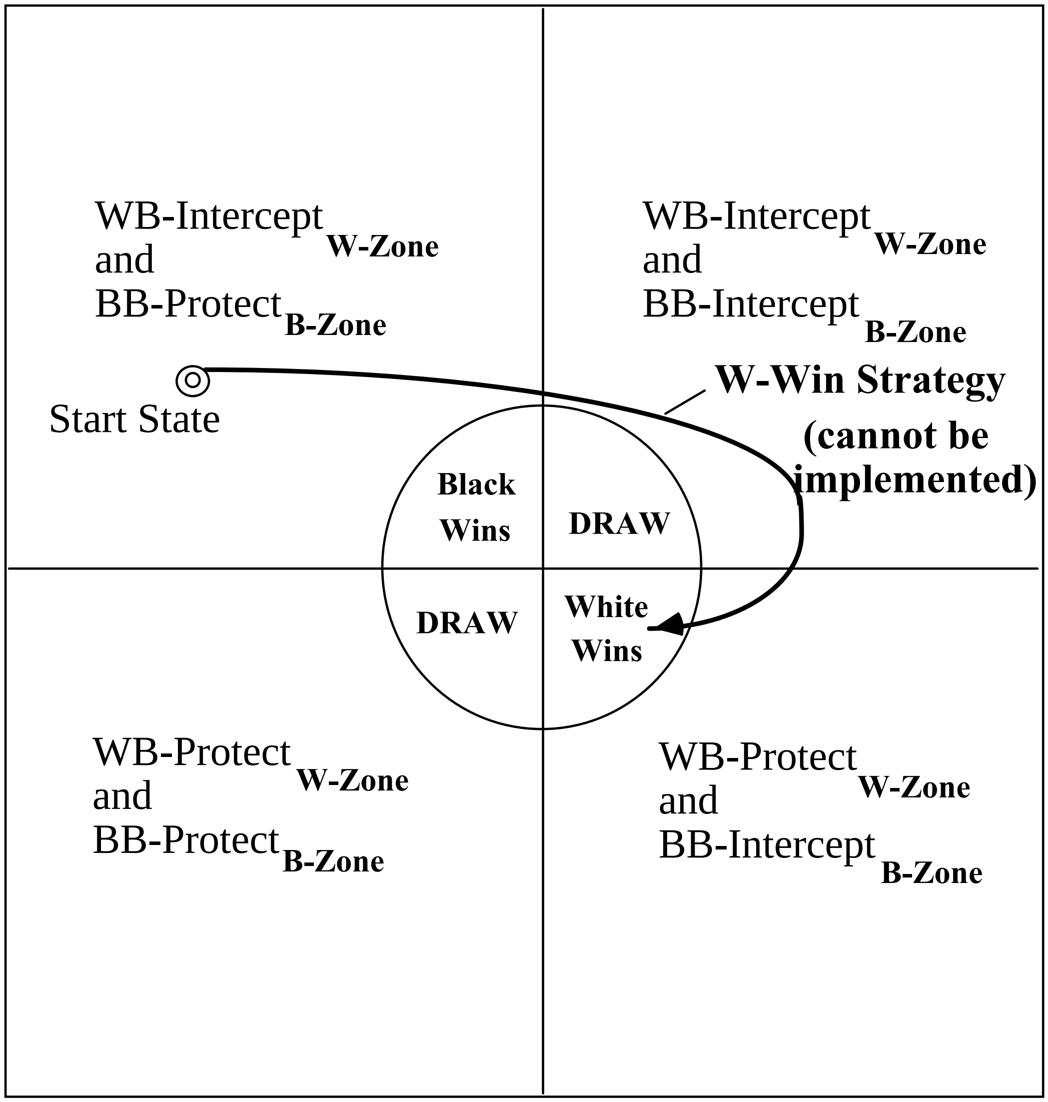
The W-Win strategy, if it exists, is to change the status of <u>both</u> W-Zone and B-Zone,

$$\textbf{WB-Intercept}_{\textbf{W-Zone}} \cap \textbf{BB-Protect}_{\textbf{B-Zone}}$$
$$\Downarrow$$
$$\textbf{WB-Protect}_{\textbf{Focus}} \cap \textbf{BB-Intercept}_{\textbf{Focus}}$$

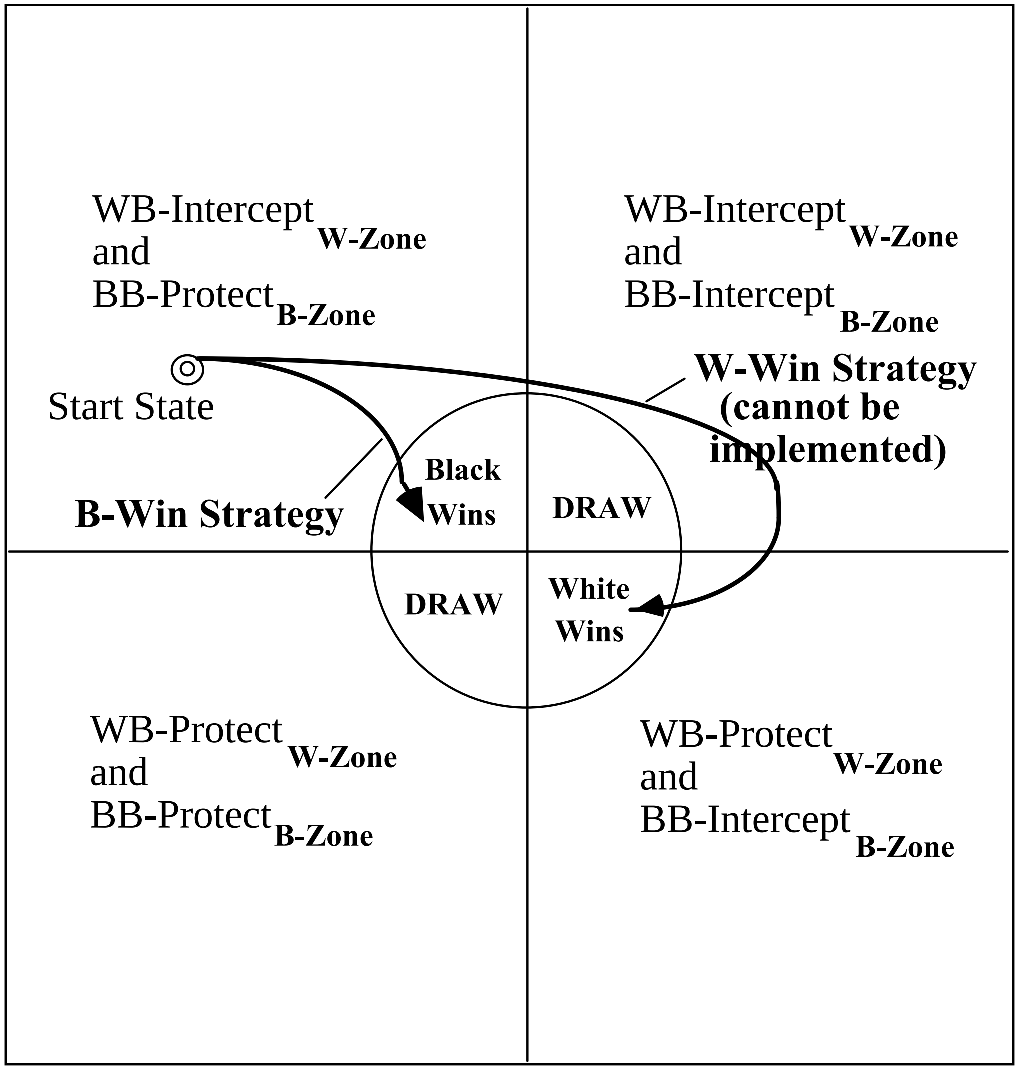




Intend-to-Win Strategies



Intend-to-Win Strategies



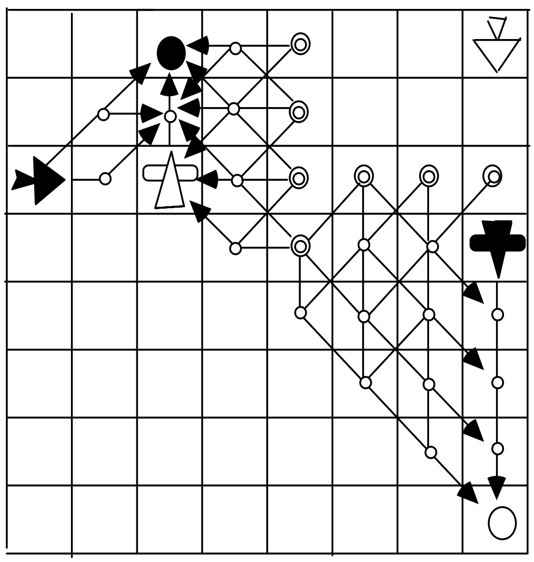
Intend-to-Win Strategies

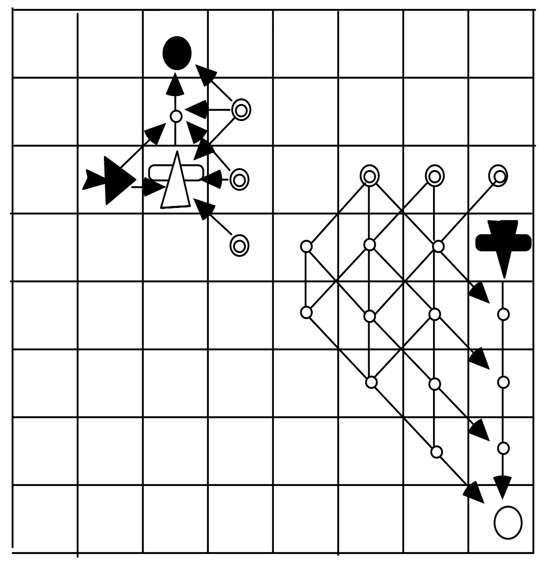
In reality, ***only one of them takes place.***

2. B-Win strategy: **B-Win = WB-Destroyed ∩ BB-Safe**

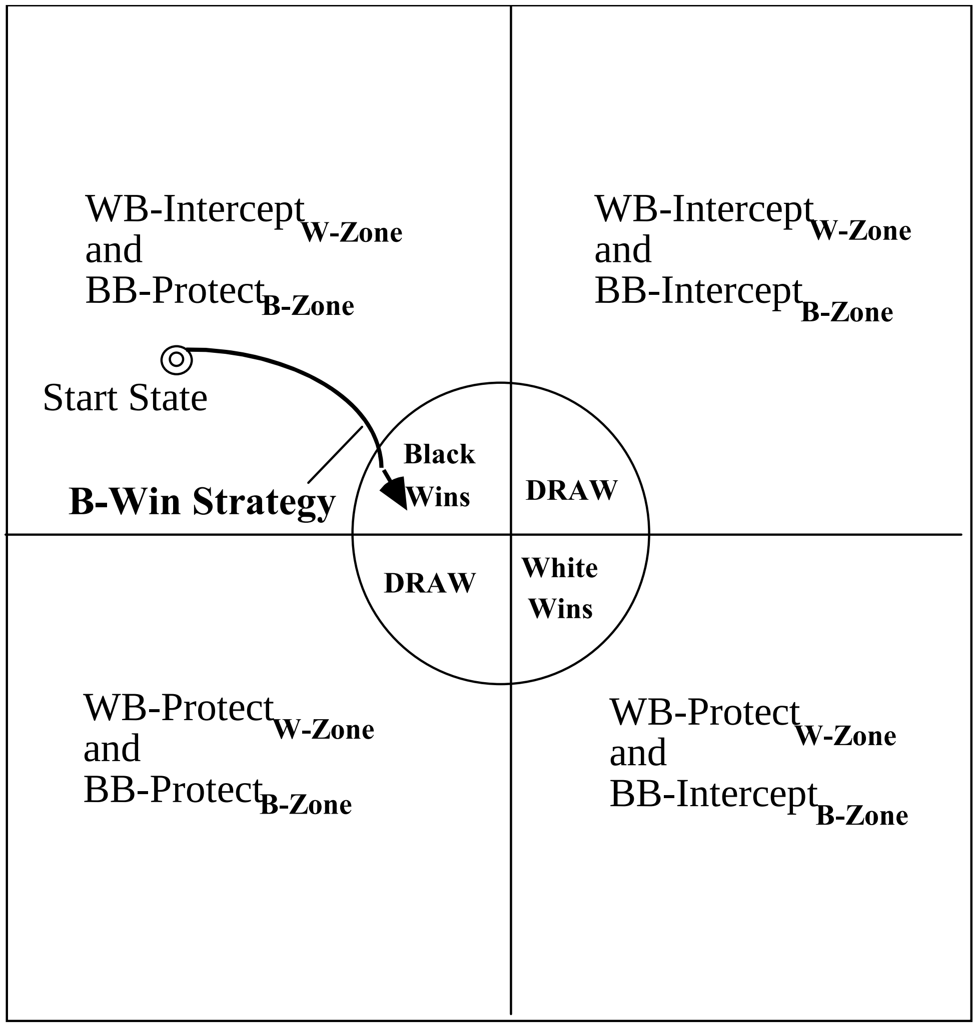
The B-Win strategy, if it exists, is to keep the status of <u>both</u>
W-Zone and B-Zone
unchanged as they are in the Start State.

$$\text{WB-Intercept}_{\text{W-Zone}} \cap \text{BB-Protect}_{\text{W-Zone}}$$
$$\Downarrow$$
$$\text{WB-Intercept}_{\text{Focus}} \cap \text{BB-Protect}_{\text{Focus}}$$

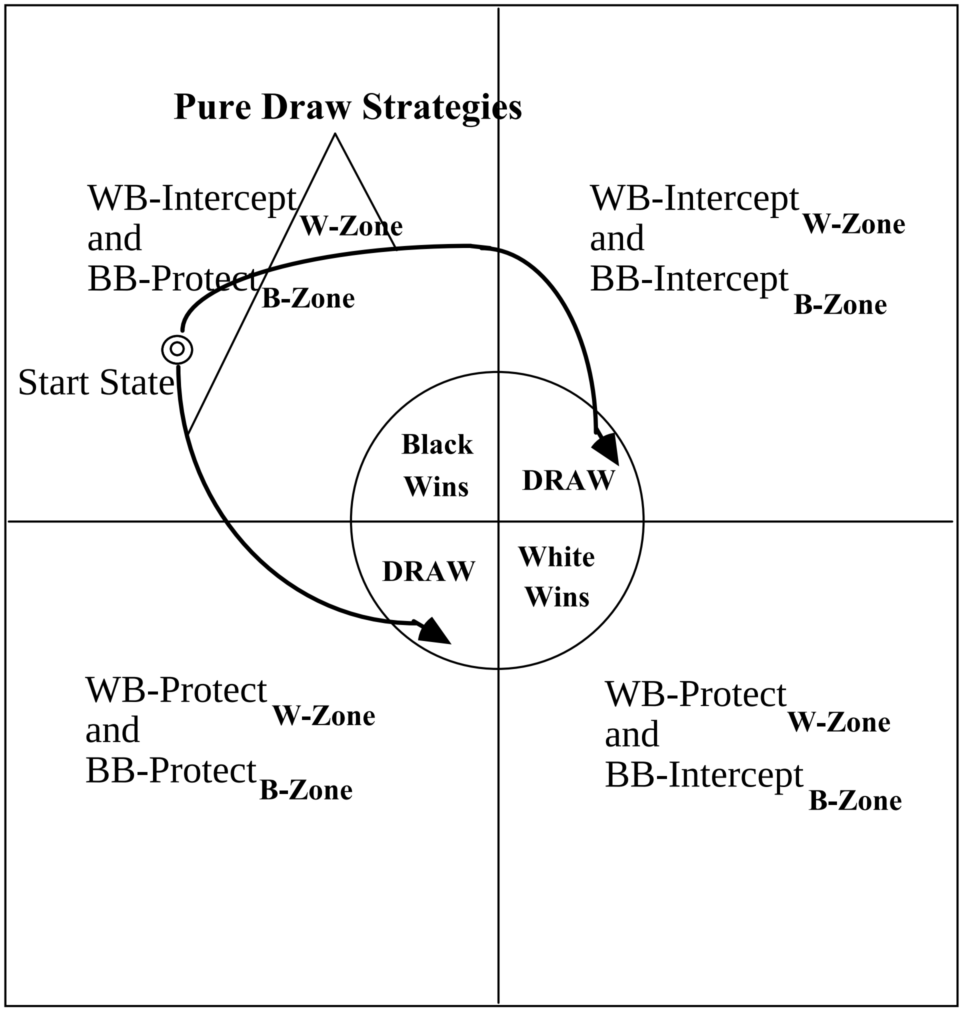




Intend-to-Win Strategies



Intend-to-Draw Strategies



Intend-to-Draw Strategy

In reality, ***only one of them takes place.***

3. Draw strategy:

Draw = Safe $\cup$Destroyed, where

Destroyed = BB-Destroyed $\cap$ WB-Destroyed,

Safe = BB-Safe $\cap$ WB-Safe

The Draw strategy, if it exists, is to change the status of <u>one</u> of the Zones, W-Zone or B-Zone,

i.e., to move from the Start State which belongs to

$$\text{WB-Intercept}_{\text{W-Zone}} \cap \text{BB-Protect}_{\text{B-Zone}}$$

to the state from

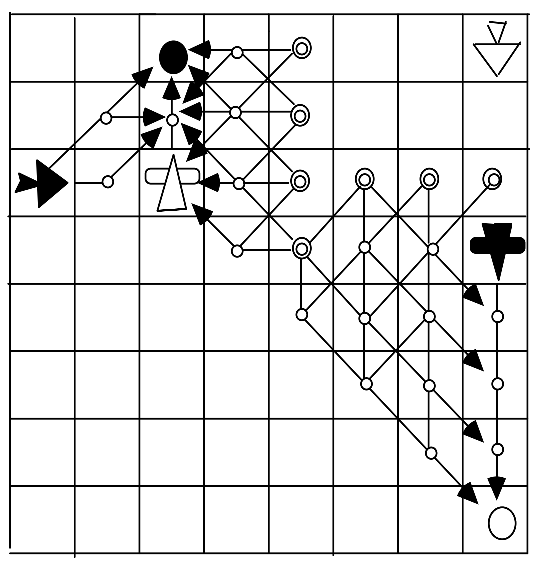
$$\text{WB-Protect}_{\text{W-Zone}} \cap \text{BB-Protect}_{\text{B-Zone}} \text{ or}$$

$$\text{WB-Intercept}_{\text{W-Zone}} \cap \text{BB-Intercept}_{\text{B-Zone}},$$

and then to the state from

$$\text{WB-Protect}_{\text{Focus}} \cap \text{BB-Protect}_{\text{Focus}}$$

$$\text{or WB-Intercept}_{\text{Focus}} \cap \text{BB-Intercept}_{\text{Focus}}.$$



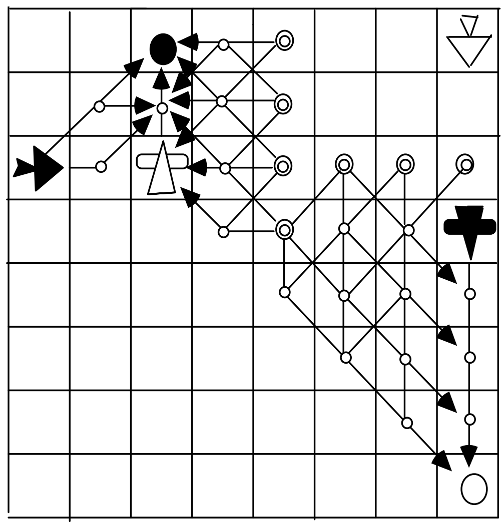
Pure Draw Strategy

StateDist1 = sd(Current State, WB-Protect$_{\text{W-Zone}}$)

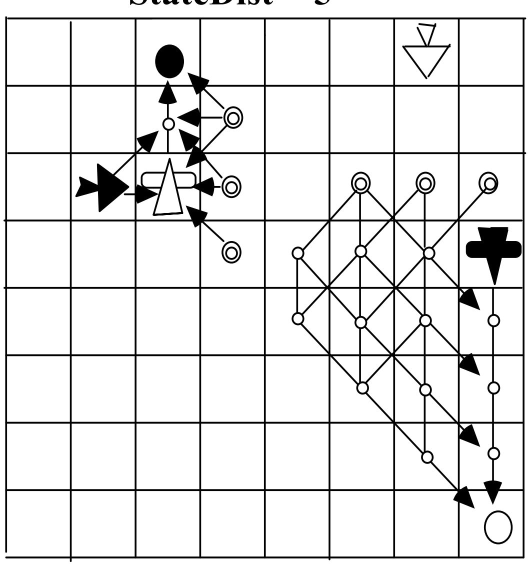
StateDist2 = sd(Current State, BB-Intercept$_{\text{B-Zone}}$)

StateDist = StateDist1 + StateDist2

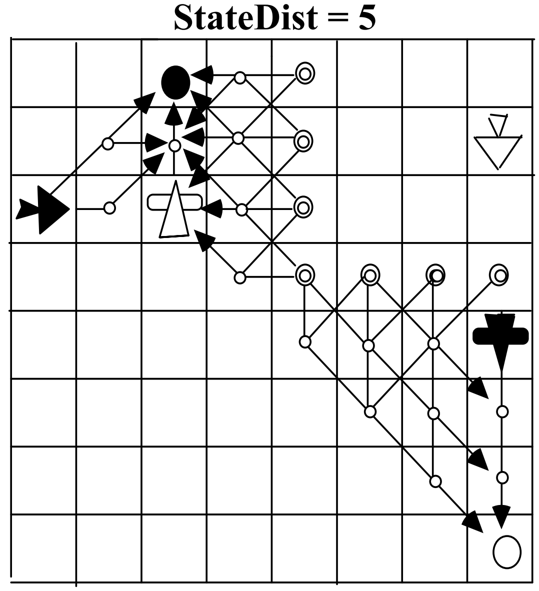
StateDist1 = 3, StateDist2 = 2
StateDist = 5



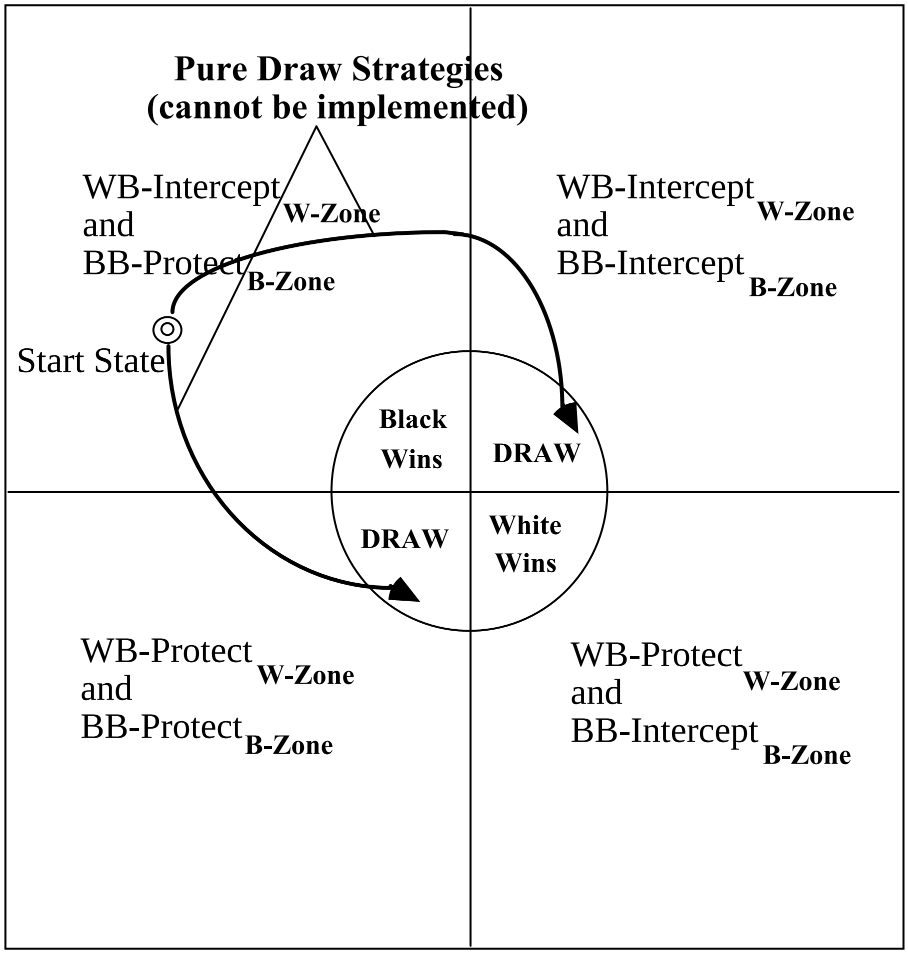
StateDist1=3, StateDist2= 2
StateDist = 5



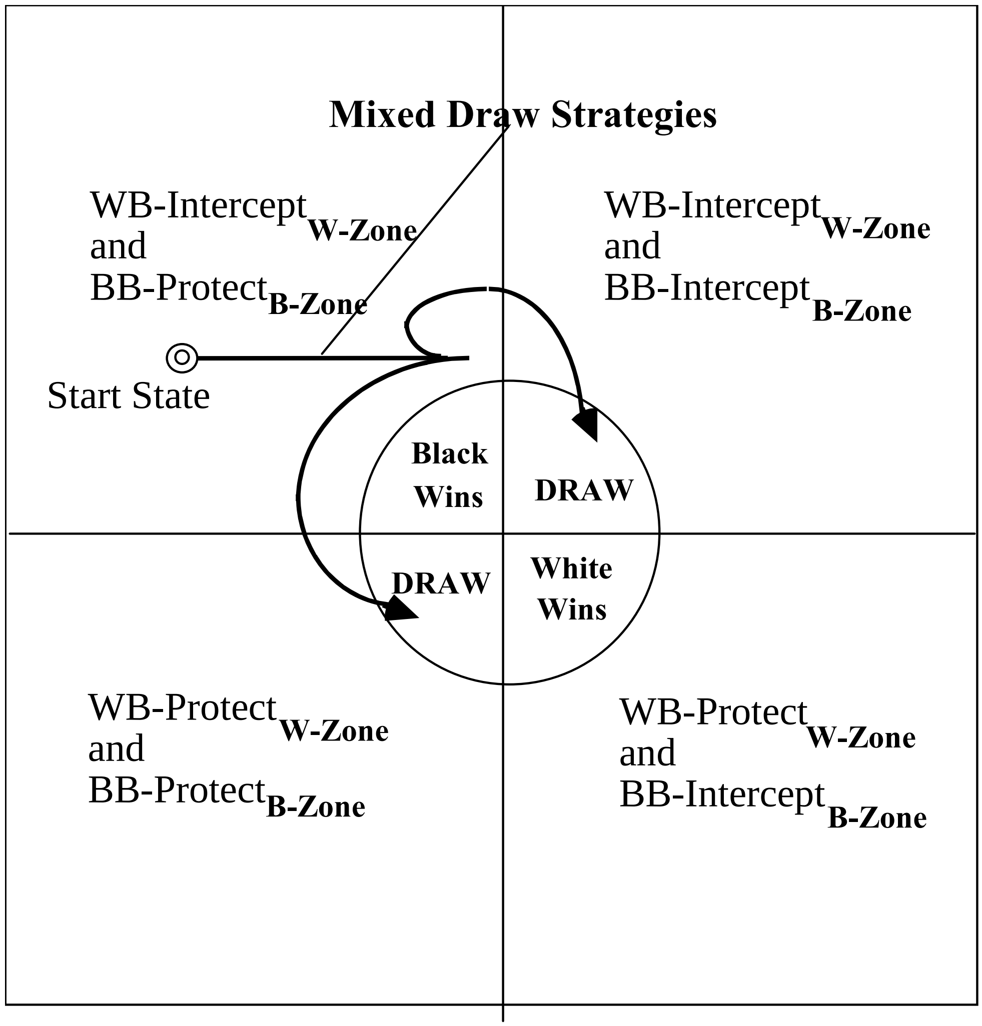
StateDist1=3, StateDist2= 2
StateDist = 5



Intend-to-Draw Strategies



Intend-to-Draw Strategies

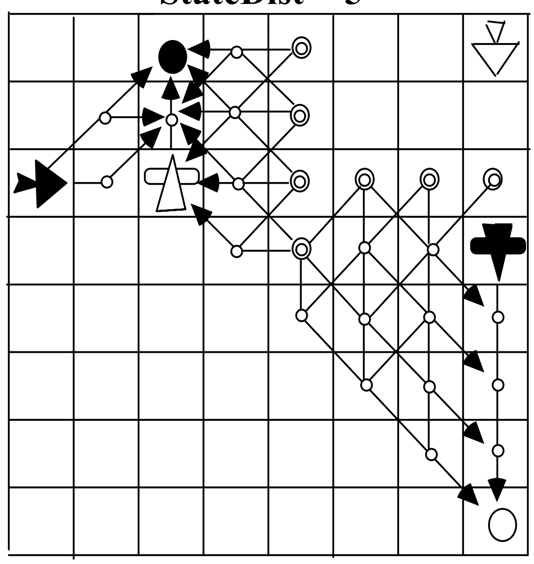


Mixed Draw Strategy

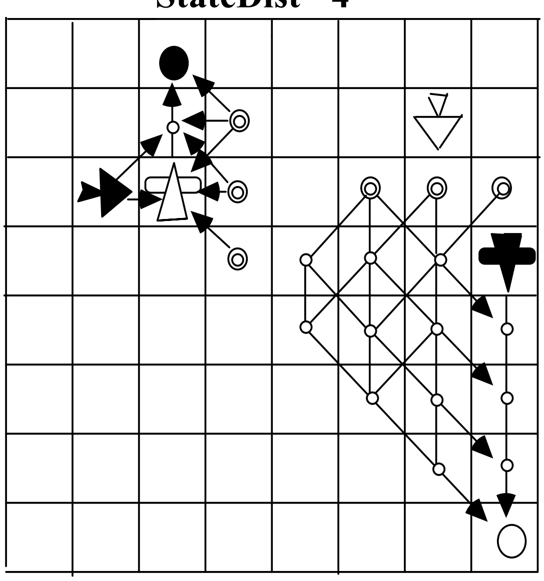
$$\text{StateDist1} = \text{sd}(\text{Current State}, \text{WB-Protect}_{\text{W-Zone}})$$
$$\text{StateDist2} = \text{sd}(\text{Current State}, \text{BB-Intercept}_{\text{B-Zone}})$$
$$\text{StateDist} = \text{StateDist1} + \text{StateDist2}$$

$$\text{StateDist} \geq \text{sd}(\text{Current State}, \text{WB-Protect}_{\text{W-Zone}} \cup \text{BB-Intercept}_{\text{B-Zone}})$$

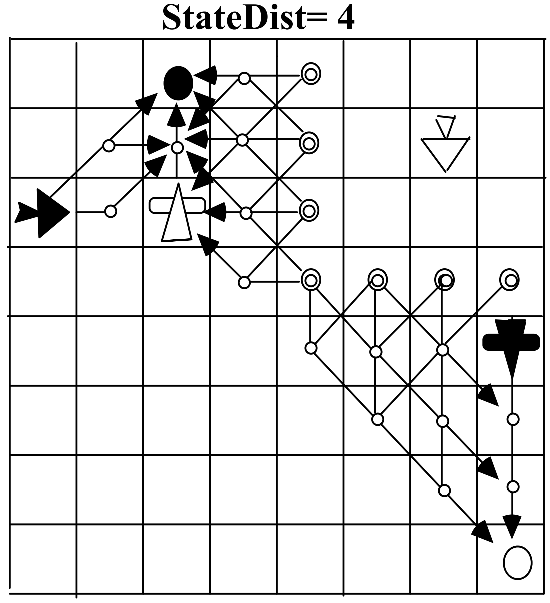
StateDist = 5



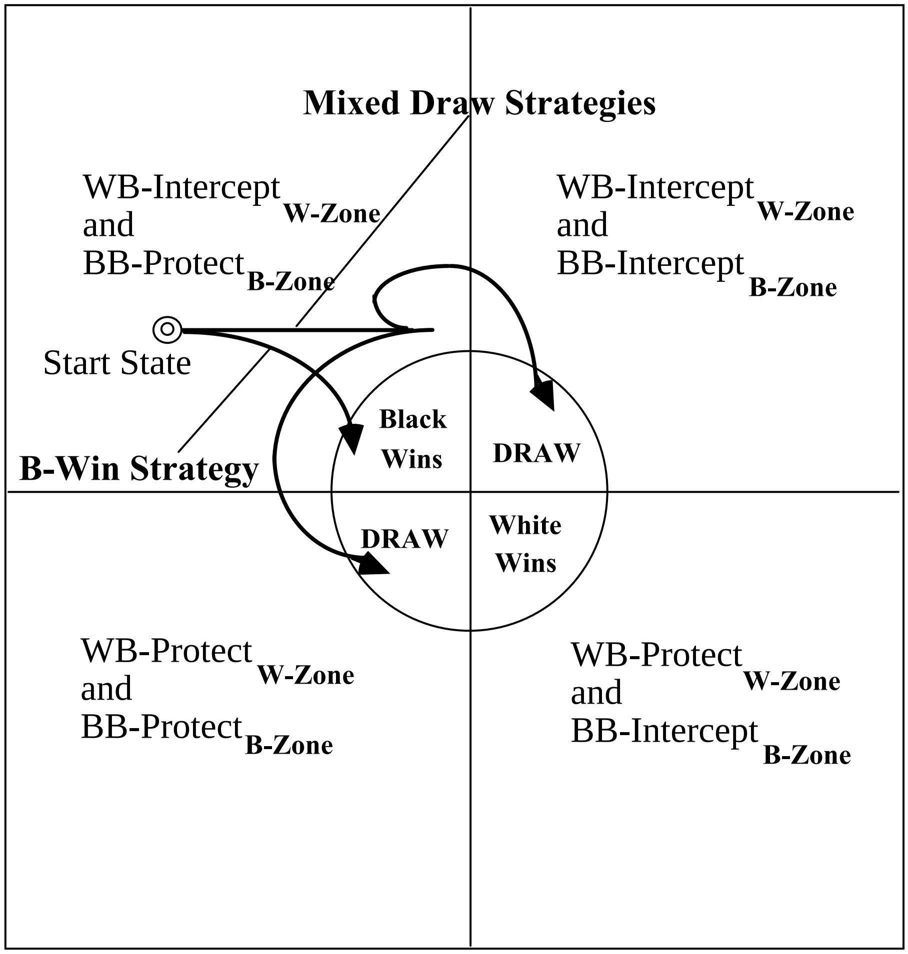
StateDist= 4



StateDist= 4

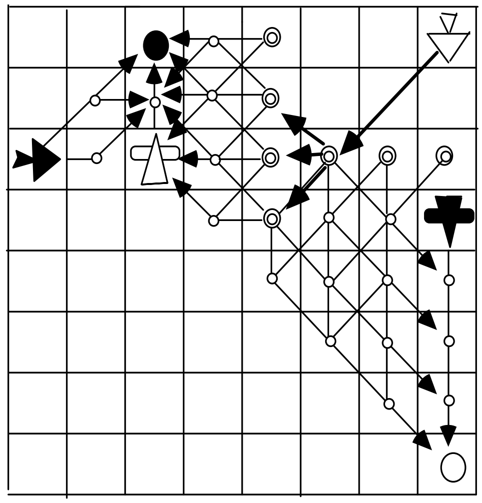


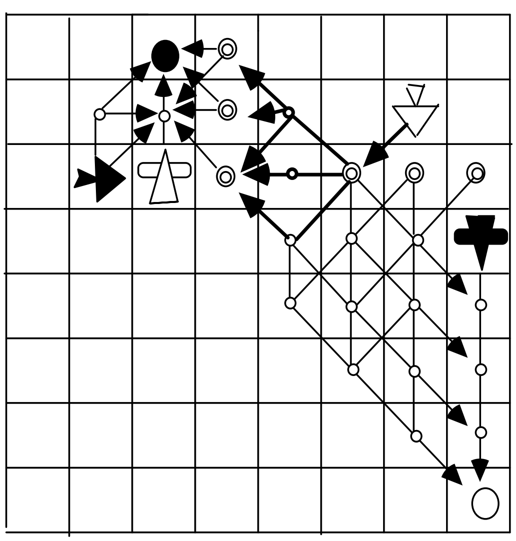
Intend-to-Draw Strategies

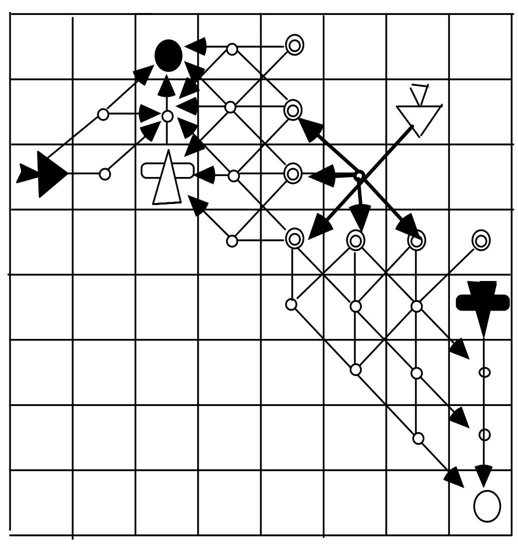


Strategy at the Start State

White follows **Mixed Draw strategy**
while Black follows **B-Win strategy**

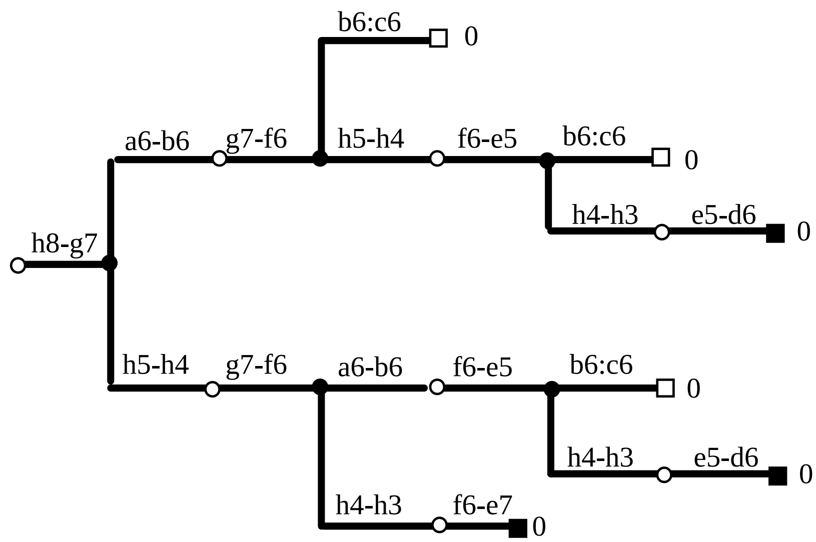


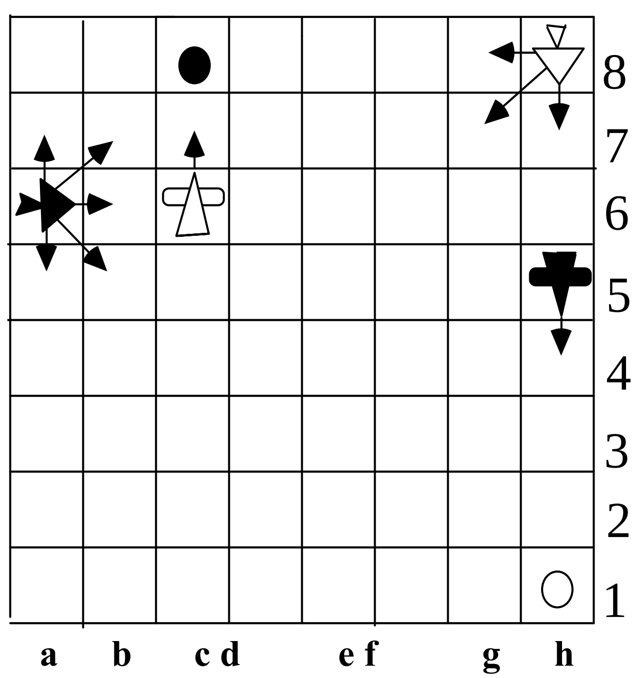




Strategy at the Start State

White follows **Mixed Draw strategy** while Black follows **B-Win strategy.**
This resulted in a **Draw strategy.**





No-Search Approach

The general algorithm of construction can be described as follows. This algorithm is an iterative construction of the solution tree, which is the optimal strategy for a certain problem. Iterations begin with the decomposition of the state space.

On the first iteration the LG tools create State Space Chart. The Chart shows all the areas of the state space that must be reached (or avoided) in order to achieve local goals. For each of the opposing sides a number of areas is identified. The description of areas is given employing LG hierarchy of formal languages, Language of Trajectories and Language of Networks (Zones). The main tool for describing local “skirmishes”, a network of paths within operational district, is represented as a string of the Language of Zones.

A strict identification of the area of the State Space Chart reflecting the outcome of the local skirmish (Zone) might require a limited search within this Zone (or the group of connected Zones). A location of the start state on the Chart is identified.

No-Search Approach (continued)

Once the Chart is created, LG tools are able to construct intend-to-win strategies for each of the opposing sides. Basically, these strategies show classes of possible paths in the state space from the start state to the desired area on the Chart. Moreover, every strategy on the Chart is accompanied by the constructive description of how to follow this strategy. The description includes an algorithm that shows which Zones and in which order must be activated, and which variants within the Zones must be followed. During the next step some of the strategies are eliminated as non-implementable. One of the reasons for elimination can be just a simple prediction of the opponent’s reaction to a certain strategy. This reaction can make it impossible to reach the desired area on the Chart by following this strategy. If opponent’s obvious responses cannot prevent us from following the strategy, this strategy cannot be eliminated.

Then the LG system makes an attempt to implement a non-eliminated intend-to-win strategy. It makes a move following this strategy. This move begins the construction of the tree, the sought *optimal strategy* of the problem. If this is a concurrent system, both sides make their concurrent moves following their respective intend-to-win strategies. This will be the first move of the optimal strategy to be constructed. This completes the first iteration.

Due to simplicity of the problem only one iteration was demonstrated. For different problems a number of iterations would be required.

No-Search Approach (continued)

After the first move, the LG system is in the new state from which new local goals might be seen and a new decomposition of the state space might be justified. On the next iteration, a new State Space Chart is constructed. The rest of the steps are repeated and a new move is selected for the optimal strategy construction. Iterations continue until local goals have been reached. Note that failure to reach the goal by one side always means that the other side did reach its goal (a win, but possibly, a draw). The number of such iterations is limited by the number of nodes in the optimal strategy (the solution tree).