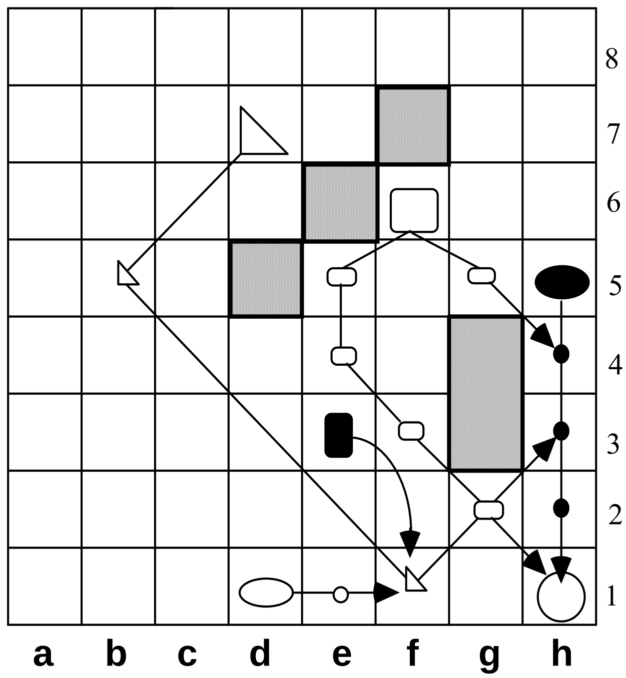
Searches for Robot Control Model



An ***Abstract Board Game***
is the following eight-tuple

$$< \mathbf{X, P, R_p, \{ON\}, v, S_i, S_t, TR} >$$

X = {x_i} is a finite set of ***points***;

P= {p_i} is a finite set of ***elements***; $\mathbf{P=P_1 \cup P_2, P_1 \cap P_2} = \varnothing$;

R_p(x, y) is a family of binary relations of ***reachability*** in X
($x \in X$, $y \in X$, $p \in P$); y is ***reachable*** from x for p;

ON(p) = x is a partial function of ***placement*** of elements P into X;

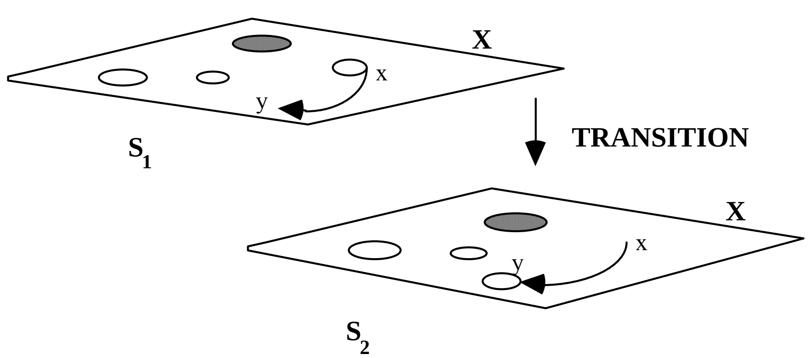
val > 0 is a real function, ***val***(p_i) are the ***values*** of elements;

S_i is a set of ***initial*** states of the system,
a certain set of formulas {ON(p_i) = x_i};

S_t is a set ***target*** states of the system (as S_i);

TR is a set of operators **TRANSITION(p, x, y)** for transition of the system from one state to another described as follows

precondition: (ON(p) = x) ^ R_p(x, y)
delete: ON(p) = x, ON(q) = y
add: ON(p) = y



Family of Grammars of Reduced Searches G_{rs}

L	*Q*	Kernel, π_k	π_n	F_T	F_F
1	Q_1	***S***(i) –> ***A***(i)	**M**(i):= $\mathbf{m_S}$(STATE) formulas for $\mathbf{G_{rs}}$	2	ø

2	$Q_2^{G_{rs}}$	***A***(i) –> ***A***(End)
		π(End)***A***(i)
			Parent(End) := i
			If ***Child***(i) ≠ 0
			then
			Sibling(***Child***(i)) := End
			else
			Sibling(i) := 0
			Child(i) := End
			Endif
			End := End+1
			d := d+1
			SIGN := – SIGN
			
			STATE := NEWSTATE
			m(End) := $\mathbf{m_S}$(NEWSTATE)
			V(End) := **BIG_NUMBER** ∗ SIGN
			
			WHO(End) := ***Element***(NEWMOVE)
			FROM(End) := ***X***(NEWMOVE)
			TO(End) := ***Y***(NEWMOVE)
			
			formulas for $\mathbf{G_{rs}}$
				2	3

3	Q_3	***A***(i) –> ***e***	**if** d ≠ 0 **then** SIGN := **–** SIGN d:= d **–** 1 **endif** STATE:= TRANSITION^{-1} (WHO(i), FROM(i), TO(i)) (STATE) **V**(***Parent***(i)) := ***MINIMAX***(SIGN, **V**(***Parent***(i)), ***LEAF***(**V**(i), **m**(i))) **V**(i) := ***LEAF***(**V**(i), **m**(i)) formulas for $\mathbf{G_{rs}}$	2	ø

At the beginning: i = 0, End =1, d =0; **m**(i), **V**(i), FROM(i), TO(i),***Child***(i), ***Sibling***(i),***Parent***(i) are equal to 0;WHO(i)∈P ∀i ≥ 0;SIGN∈{-1,1}, STATE∈SPACE

$\boldsymbol{V_T} = \{\boldsymbol{\pi}\}$ $\quad \boldsymbol{V_N} = \{\boldsymbol{A}, \boldsymbol{S}\}$ $\quad \boldsymbol{Pred} = \{\boldsymbol{Q_1}, \boldsymbol{Q_2}, \boldsymbol{Q_3}\}$

$\boldsymbol{Q_1} = \boldsymbol{T}$

$\boldsymbol{Q_2}^{\mathbf{Grs}} = (\exists p\ \exists x\ \exists y\ ((((SIGN = 1) \wedge (p \in P_1)) \vee ((SIGN = -1) \wedge (p \in P_2))) \wedge (ON(p) = x) \wedge R_p(x, y)) \wedge (d < d_{max}) \wedge \neg \mathbf{CUT}^{\mathbf{Grs}}$,

where $\mathbf{CUT}^{\mathbf{Grs}}$ depends on the specific grammar $\mathbf{G_{rs}}$

$\boldsymbol{Q_3} = \boldsymbol{T}$

Fcon = {$\mathbf{MV}^{\mathbf{Grs}}$, ***Element***, ***X***, ***Y, MINIMAX***, ***LEAF***, TRANSITION, TRANSITION^{-1}, $\mathbf{m_S}$}

$\mathbf{MV}^{\mathbf{Grs}}$ depends on the grammar $\mathbf{G_{rs}}$

Fvar = {End, SIGN, **m**, **V**, WHO, FROM, TO, ***Child***, ***Parent***, ***Sibling***,STATE}

Parm: $S \rightarrow \{i\}$, $A \rightarrow \{i\}$, $\boldsymbol{\pi} \rightarrow \{i\}$, $\boldsymbol{L} = \{1, 2, 3\}$

Functions are defined as follows:

$D(\mathbf{m_S}) = SPACE$, $\mathbf{m_S}(STATE) = \sum_{p \in P_1} \boldsymbol{val}(p) - \sum_{p \in P_2} \boldsymbol{val}(p)$,

Functions ***val***, TRANSITION, TRANSITION^{-1} are from the Definition of ABG; $BIG\_NUMBER = \sum_{p \in P} \boldsymbol{val}(p) + 1$

$D(\boldsymbol{LEAF}) = \mathbf{Z+} \times \mathbf{Z+}$, **If** $\boldsymbol{a}$ **= BIG_NUMBER then** $\boldsymbol{LEAF(a, b) := b}$**, else** $\boldsymbol{LEAF(a, b) := a}$

$D(\boldsymbol{MINIMAX}) = \{-1, +1\} \times \mathbf{Z+} \times \mathbf{Z+}$

$$\boldsymbol{MINIMAX}(SN, v_1, v_2) = \begin{cases} \boldsymbol{max}(v_1, v_2), \text{if } SN = 1 \\ \boldsymbol{min}(v_1, v_2), \text{if } SN = -1 \end{cases}$$

MOVE = $\{(p, x, y) \mid x \in X, y \in X, (\exists p\ ((((SIGN = 1) \wedge (p \in P_1)) \vee ((SIGN = -1) \wedge (p \in P_2))) \wedge (ON(p) = x) \wedge R_p(x, y))))\}$

MOVE = $\{(p_1, x_1, y_1), ..., (p_k, x_k, y_k)\}$

NEWMOVE = $\mathbf{MV}^{\mathbf{Grs}}$(d, End, SIGN, **m**, **V**, WHO, FROM, TO, ***Child***, ***Parent***, ***Sibling***, STATE, ...)

MV yields the ordinal of a triple from the list MOVE

Element(NEWMOVE) $= p_{NEWMOVE}$,

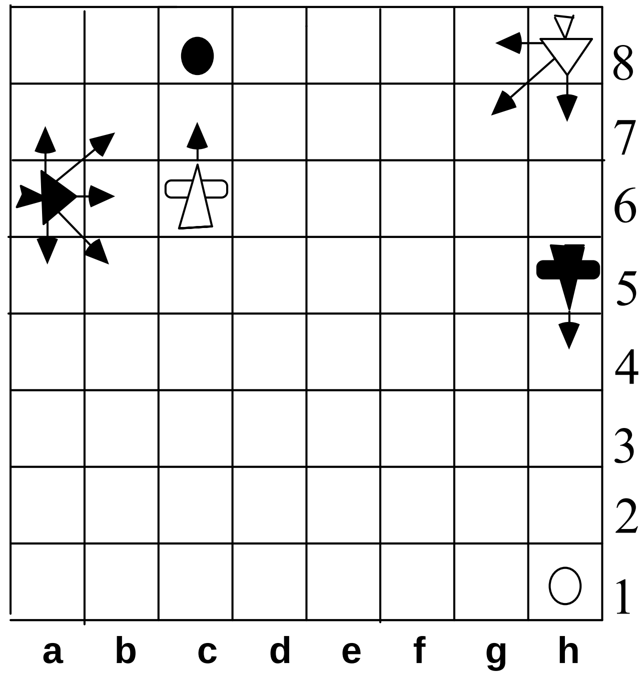
X(NEWMOVE) $= x_{NEWMOVE}$

Y(NEWMOVE) $= y_{NEWMOVE}$

NEWSTATE = TRANSITION(***Element***(NEWMOVE), ***X***(NEWMOVE), ***Y***(NEWMOVE)) (STATE)

Optimization problem for autonomous aerospace robotic vehicles with serial alternating motions

2D/4A Problem



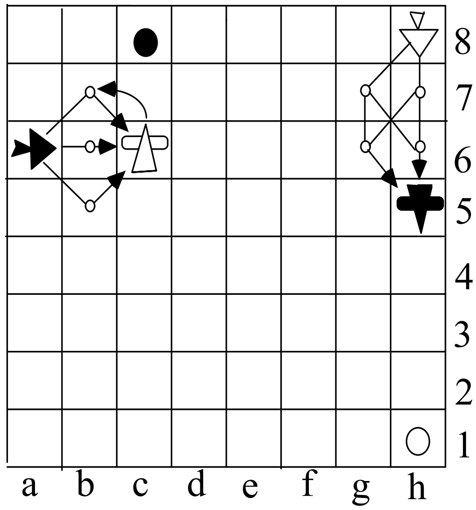
***Is there a strategy for the White to make a draw*?**

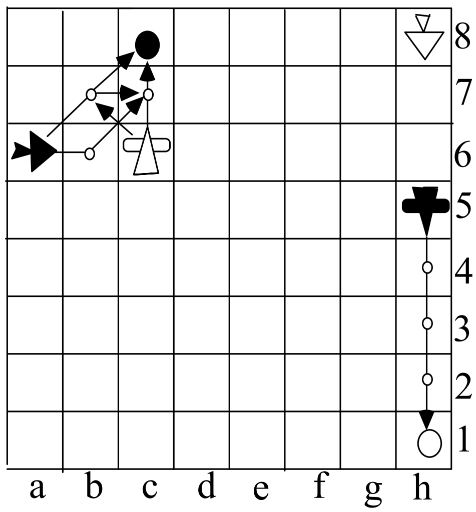
The specific question is as follows.
Is there an optimal strategy that provides one of the following?

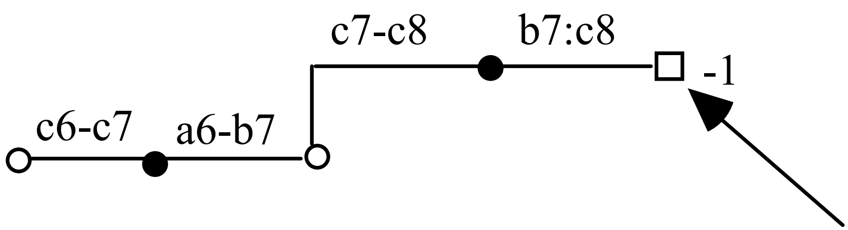
1. **Both BOMBERs hit their targets on subsequent time increments and stay safe for at least one time increment.**

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

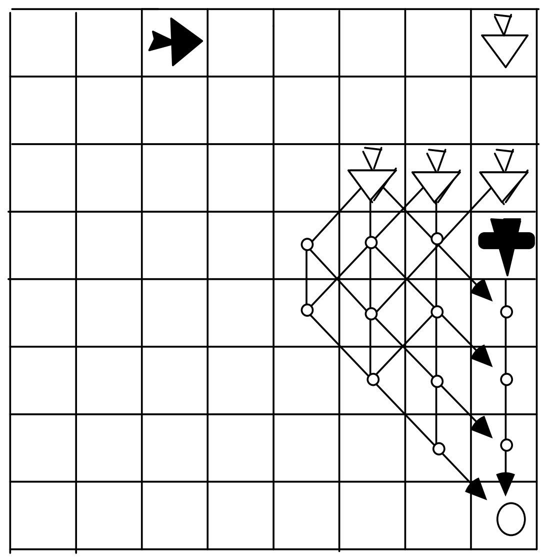
Zones in the Start State of the Robot Control Model.

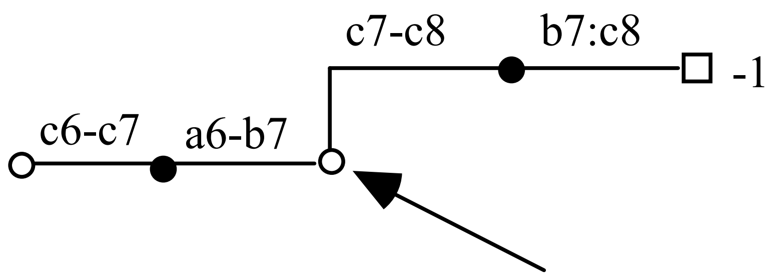




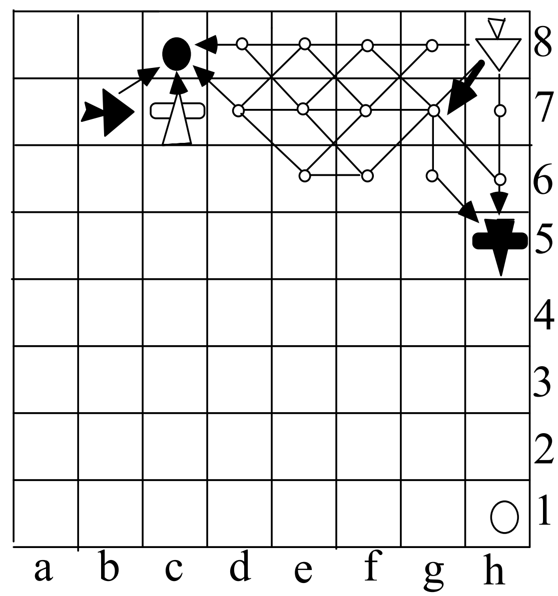


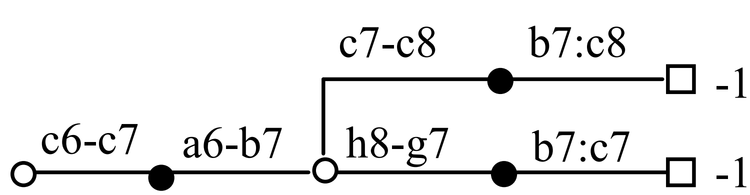
State where the branch was terminated. Zone gateways.

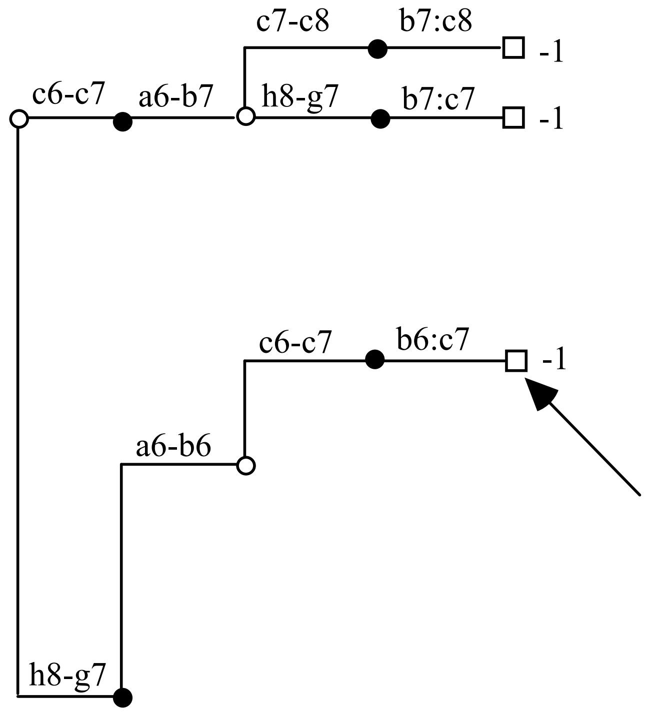




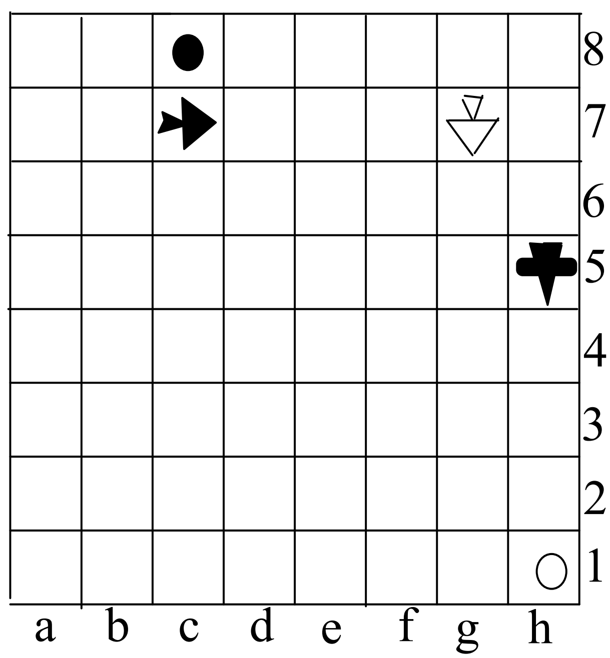
State where the control Zone from h8 to c8 was included into the search

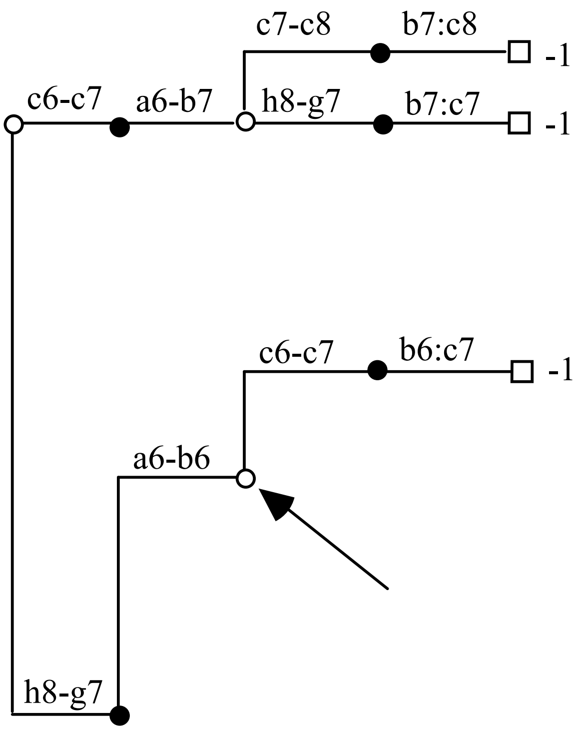




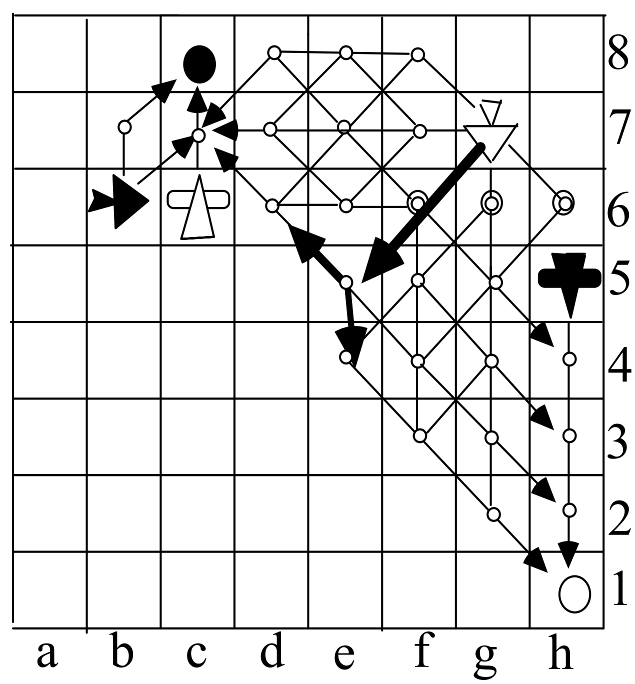


State where the control Zone from g7 to c7 was detected

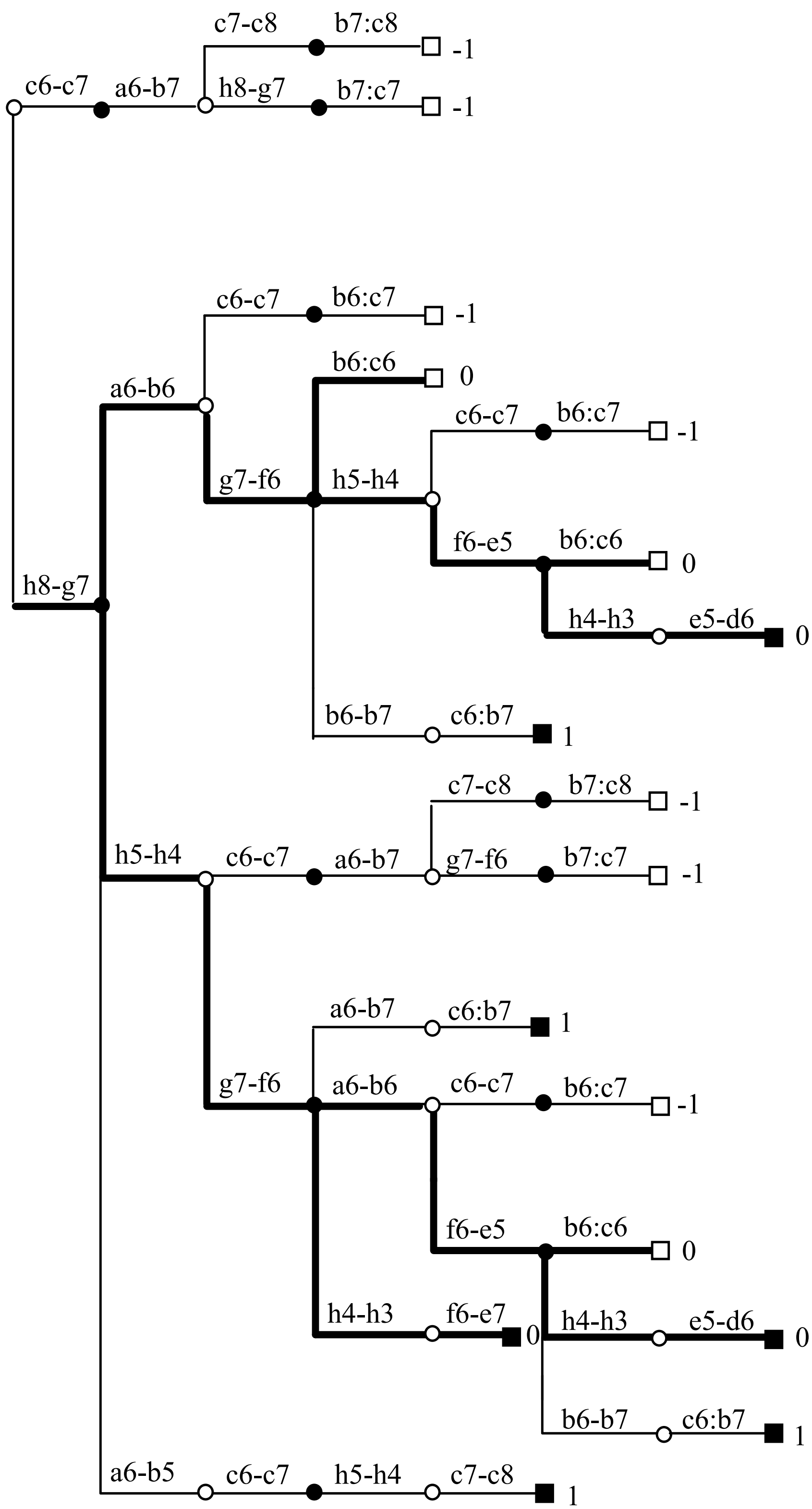




State where the control Zone from g7 to c7 was included into the search



Search tree for the robotic vehicles with alternating serial motions



Brute Force Search with limited depth

Add function NM(i) into ***Fvar***. At the beginning of derivation NM(i) = 1 for all i from **Z+**. Include new functional formula in the production 2 (section $\boldsymbol{\pi_n}$):

NM(i) := NM(i) +1

Function MV in this case is as follows:

MV(r) = r, r is from **Z+**.

NEWMOVE = MV(NM(i)).

Predicate CUT:

CUT = MV(i) > | MOVE|, where |MOVE| is the number of triples in MOVE.

Best-Move Search without backtrack

Add function ***Value***: $P \times X \times X \longrightarrow$ **R** and function **DIRECTION.** The algorithm for computation of ***Value*** depends on the particular heuristic function.

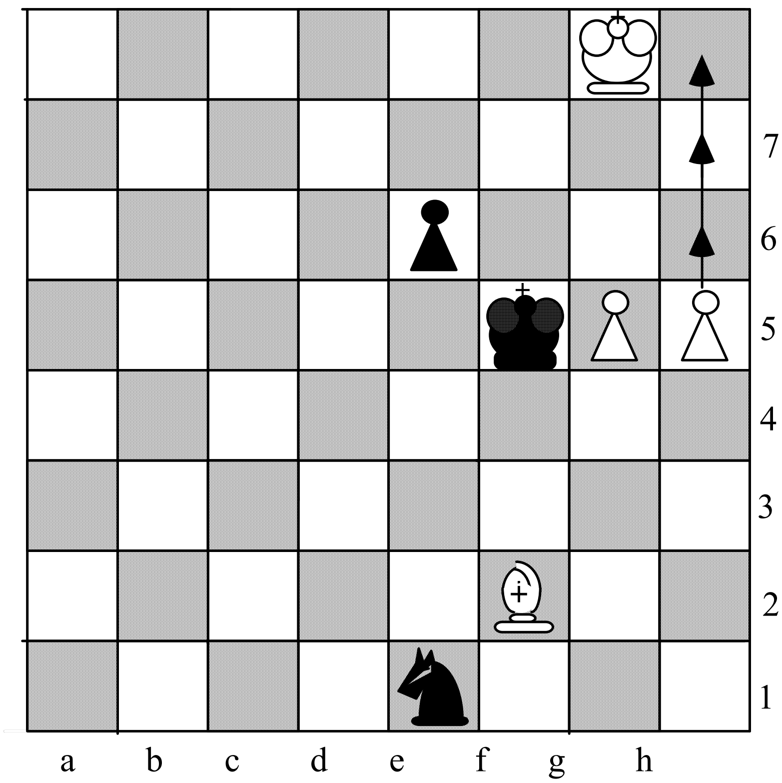
Function MV is defined as follows:

MV(i, STATE) is the number of triple $(p_i, x_i, y_i) \in$ MOVE for which ***Value***((p_i, x_i, y_i)) is maximum.

Add formula **DIRECTION**:= 1 into the section $\boldsymbol{\pi_n}$ of production 1 and **DIRECTION** := -1 in production 3.

CUT = ((MOVE = $\emptyset$) $\vee$ (**DIRECTION**= -1))

Generate Search in this Zone



A Comparison of Searches
for the same processing time