Local Model Checking of Weighted CTL with Upper-Bound Constraints

Jonas Finnemann Jensen, Kim Guldstrand Larsen,
Jiří Srba, and Lars Kaerlund Oestergaard

Department of Computer Science, Aalborg University
Selma Lagerlöfs Vej 300, 9220 Aalborg, Denmark

July 8, 2013

Introduction

- Model checking *both* functional and quantitative properties.
  - Embedded systems - resources are very limited.
  - Resource constraints: cost, memory, bandwidth, power, etc.
- We extend well-known models and temporal logic:
  - *Weighted* CTL & weighted Kripke structures.
- Efficient model checking of WCTL:
  - *Symbolic* dependency graphs
  - Local/on-the-fly fixed-point algorithm

Outline

- Weighted Model Checking
- Dependency graphs
- Symbolic dependency graphs
- Experiments
- Conclusion

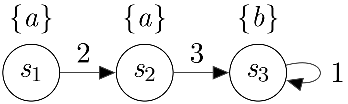
Weighted Kripke Structure

Definition (WKS)

A WKS is a tuple $\mathcal{K} = (S, \mathcal{AP}, L, \rightarrow)$, where

- S is a finite set of states,
- $\mathcal{AP}$ is a set of atomic propositions,
- $L : S \rightarrow \mathcal{P}(\mathcal{AP})$ is a labelling function, and
- $\rightarrow \subseteq S \times \mathbb{N}_0 \times S$ is a transition relation.

Example



Weighted Computation Tree Logic (WCTL)

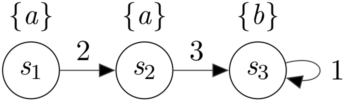
The set of WCTL formulas is given as follows.

$$
\begin{aligned}
\varphi ::= &\ \textit{true} \mid \textit{false} && \text{(Boolean Properties)} \\
&\mid a && \text{(Atomic Proposition)} \\
&\mid \varphi_1 \wedge \varphi_2 && \text{(Conjunction)} \\
&\mid \varphi_1 \vee \varphi_2 && \text{(Disjunction)} \\
&\mid E\ \varphi_1\ U_{\leq k}\ \varphi_2 && \text{(Existential Until)} \\
&\mid A\ \varphi_1\ U_{\leq k}\ \varphi_2 && \text{(Universal Until)} \\
&\mid EX_{\leq k}\ \varphi && \text{(Existential Next)} \\
&\mid AX_{\leq k}\ \varphi && \text{(Universal Next)}
\end{aligned}
$$

where $k \in \mathbb{N}_0$ and $a \in \mathcal{AP}$.

Semantics of the Until Modality

Example



We have that

$$s_1 \models E\ a\ U_{\leq 8}\ b$$
$$s_1 \not\models E\ a\ U_{\leq 4}\ b$$

Consider the only run

$$\sigma = \underbrace{\overbrace{s_1 \xrightarrow{2} s_2}^{a \text{ holds}} \xrightarrow{3} \overset{b \text{ holds}}{s_3}}_{\text{Accumulated weight } 2+3=5} \xrightarrow{1} s_3 \ldots$$

Dependency Graph (1)

Definition (Dependency Graph)

A DG is a pair $G = (V, E)$, where

- V is a set of configurations, and
- $E \subseteq V \times \mathcal{P}(V)$ is a set of hyper-edges.

- An assignment is a mapping $A : V \rightarrow \{1, 0\}$
- A_{min} is the minimum fixed-point assignment.

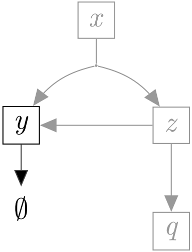
$A_{min}(u) = 1$ if there is $(u, T) \in E$ s.t. for all $v \in T$ we have $A_{min}(v) = 1$.

Functor

$$F(A)(u) = \bigvee_{(u,T)\in E} \left(\bigwedge_{v \in T} A(v) \right)$$

$A_{min} = F(F(\ldots F(A_0)))$ where $A_0(v) = 0$

Example



$A_{min}(y) = 1$ as $(y, \emptyset) \in E$

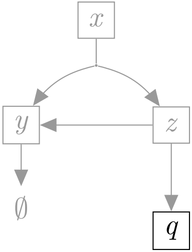
Dependency Graph (2)

Definition (Dependency Graph)

A DG is a pair $G = (V, E)$, where

- V is a set of configurations, and
- $E \subseteq V \times \mathcal{P}(V)$ is a set of hyper-edges.

Example



$$A_{min}(q) = 0$$

- An assignment is a mapping $A : V \to \{1, 0\}$
- A_{min} is the minimum fixed-point assignment.

$A_{min}(u) = 1$ if there is $(u, T) \in E$ s.t.
for all $v \in T$ we have $A_{min}(v) = 1$.

Functor

$$F(A)(u) = \bigvee_{(u,T) \in E} \left(\bigwedge_{v \in T} A(v) \right)$$

$A_{min} = F(F(\ldots F(A_0)))$ where $A_0(v) = 0$

Dependency Graph (3)

Definition (Dependency Graph)

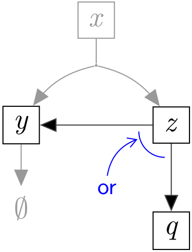
A DG is a pair $G = (V, E)$, where

- V is a set of configurations, and
- $E \subseteq V \times \mathcal{P}(V)$ is a set of hyper-edges.

- An assignment is a mapping $A : V \to \{1, 0\}$
- A_{min} is the minimum fixed-point assignment.

$A_{min}(u) = 1$ if there is $(u, T) \in E$ s.t.
for all $v \in T$ we have $A_{min}(v) = 1$.

Example



$$A_{min}(z) = A_{min}(y) \vee A_{min}(q)$$

Functor

$$F(A)(u) = \bigvee_{(u,T)\in E} \left(\bigwedge_{v\in T} A(v) \right)$$

$A_{min} = F(F(\ldots F(A_0)))$ where $A_0(v) = 0$

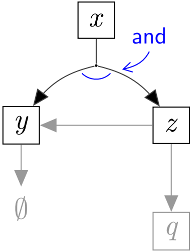
Dependency Graph (4)

Definition (Dependency Graph)

A DG is a pair $G = (V, E)$, where

- V is a set of configurations, and
- $E \subseteq V \times \mathcal{P}(V)$ is a set of hyper-edges.

Example



$$A_{min}(x) = A_{min}(y) \wedge A_{min}(z)$$

- An assignment is a mapping $A : V \to \{1, 0\}$
- A_{min} is the minimum fixed-point assignment.

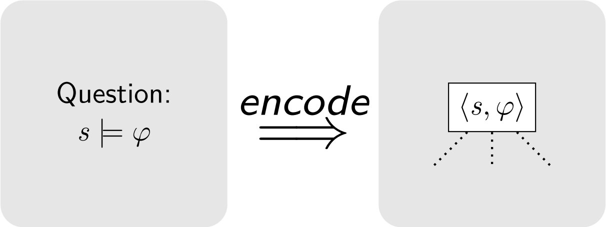
$A_{min}(u) = 1$ if there is $(u, T) \in E$ s.t.
for all $v \in T$ we have $A_{min}(v) = 1$.

Functor

$$F(A)(u) = \bigvee_{(u,T)\in E} \left(\bigwedge_{v \in T} A(v) \right)$$

$A_{min} = F(F(\dots F(A_0)))$ where $A_0(v) = 0$

WCTL Model Checking with Dependency Graphs



Theorem 2

$$s \models \varphi \quad \Leftrightarrow \quad A_{min}(\langle s, \varphi \rangle) = 1$$

Encoding Example ($\varphi =$ ***true***)

Question:
$s \models$ *true*

$\xRightarrow{\textit{encode}}$

$\boxed{\langle s, \textit{true} \rangle}$
$\downarrow$
$\emptyset$

We have the vacuous case, $A_{min}(u) = 1$ *for all* u *in* $\emptyset$*, hence*

$$A_{min}(\langle s, \textbf{true} \rangle) = 1$$

Encoding Example ($\varphi = \textit{\textbf{false}}$)

Question:
$s \models \textit{false}$

$$\xRightarrow{\textit{encode}}$$

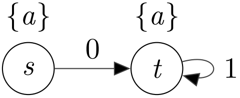
$\boxed{\langle s, \textit{false} \rangle}$

We have the trivial case, as $\langle s, \textit{false} \rangle$ has no hyper-edges, hence

$$A_{min}(\langle s, \textit{\textbf{false}} \rangle) = 0$$

Model Checking Example

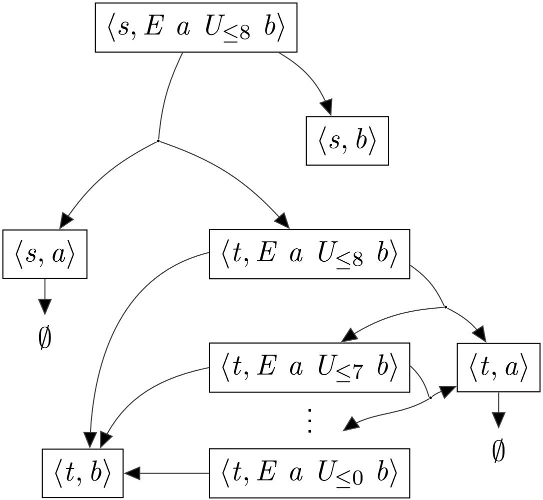
If we take the WKS



and want to determine if

$$s \models E\ a\ U_{\leq 8}\ b$$

we can encode this as:



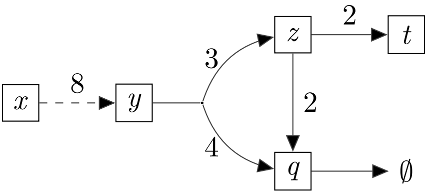
Symbolic Dependency Graphs

Definition (Symbolic Dependency Graphs)

An SDG is a triple $G = (V, H, C)$, where

- V is a finite set of configurations,
- $H \subseteq V \times \mathcal{P}(\mathbb{N}_0 \times V)$ is a finite set of hyper-edges, and
- $C \subseteq V \times \mathbb{N}_0 \times V$ is a finite set of cover-edges.

Example



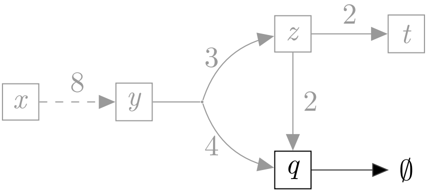
Fixed-Point A_{min} of an SDG (1)

An assignment is a mapping $A : V \rightarrow \mathbb{N}_0 \cup \{\infty\}$
Functor for minimum fixed-point A_{min}

$$F(A)(u) = \begin{cases} 0 & \text{if } \exists(u,k,v) \in C \text{ s.t. } A(v) \leq k \\ \min\limits_{(u,T)\in H}\Big(\max\{w + A(v) \mid (w,v) \in T\}\Big) & \text{otherwise.} \end{cases}$$

$A_{min} = F(\dots F(A_0))$ where $A_0(v) = \infty$.

Example



$$A_{min}(q) = 0 \text{ as } (q, \emptyset) \in E$$

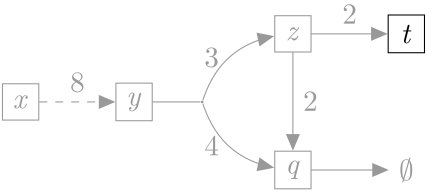
Fixed-Point A_{min} of an SDG (2)

An assignment is a mapping $A : V \to \mathbb{N}_0 \cup \{\infty\}$
Functor for minimum fixed-point A_{min}

$$F(A)(u) = \begin{cases} 0 & \text{if } \exists(u,k,v) \in C \text{ s.t. } A(v) \leq k \\ \min\limits_{(u,T)\in H}\Big(\max\{w + A(v) \mid (w,v) \in T\}\Big) & \text{otherwise.} \end{cases}$$

$A_{min} = F(\dots F(A_0))$ where $A_0(v) = \infty$.

Example



$$A_{min}(t) = \infty$$

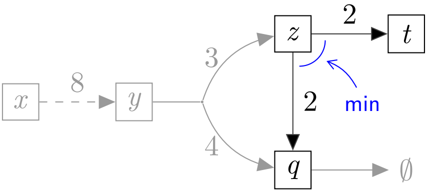
Fixed-Point A_{min} of an SDG (3)

An assignment is a mapping $A : V \to \mathbb{N}_0 \cup \{\infty\}$
Functor for minimum fixed-point A_{min}

$$F(A)(u) = \begin{cases} 0 & \text{if } \exists(u,k,v) \in C \text{ s.t. } A(v) \leq k \\ \min\limits_{(u,T)\in H}\Big(\max\{w + A(v) \mid (w,v) \in T\}\Big) & \text{otherwise.} \end{cases}$$

$A_{min} = F(\dots F(A_0))$ where $A_0(v) = \infty$.

Example



$$A_{min}(z) = \min(2 + A_{min}(q), 2 + A_{min}(t))$$

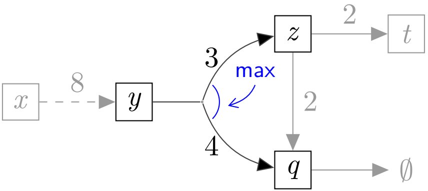
Fixed-Point A_{min} of an SDG (4)

An assignment is a mapping $A : V \to \mathbb{N}_0 \cup \{\infty\}$
Functor for minimum fixed-point A_{min}

$$F(A)(u) = \begin{cases} 0 & \text{if } \exists(u,k,v) \in C \text{ s.t. } A(v) \leq k \\ \min\limits_{(u,T)\in H} \big(\max\{w + A(v) \mid (w,v) \in T\}\big) & \text{otherwise.} \end{cases}$$

$A_{min} = F(\dots F(A_0))$ where $A_0(v) = \infty$.

Example



$$A_{min}(y) = \max(3 + A_{min}(z), 4 + A_{min}(q))$$

Fixed-Point A_{min} of an SDG (5)

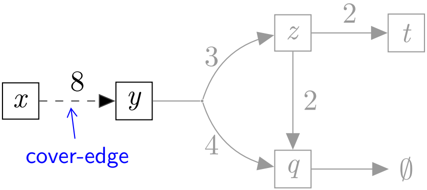
An assignment is a mapping $A : V \to \mathbb{N}_0 \cup \{\infty\}$

Functor for minimum fixed-point A_{min}

$$F(A)(u) = \begin{cases} 0 & \text{if } \exists(u,k,v) \in C \text{ s.t. } A(v) \leq k \\ \min\limits_{(u,T)\in H} \Big(\max\{w + A(v) \mid (w,v) \in T\} \Big) & \text{otherwise.} \end{cases}$$

$A_{min} = F(\dots F(A_0))$ where $A_0(v) = \infty$.

Example

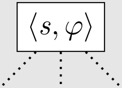


$$A_{min}(x) = \begin{cases} 0 & \text{if } A_{min}(y) \leq 8 \\ \infty & \text{otherwise} \end{cases}$$

WCTL Model Checking with SDGs

Question:
$s \models \varphi$

$\xRightarrow{\textit{encode}}$



Theorem 5

$$s \models \varphi \quad \Leftrightarrow \quad A_{min}(\langle s, \varphi \rangle) = 0$$

Encoding Example ($\varphi =$ ***true***)

Question:
$s \models$ *true*

$$\xRightarrow{\textit{encode}}$$

$$\boxed{\langle s, \textit{true} \rangle} \longrightarrow \emptyset$$

We have the empty target-set and $\max(\emptyset) = 0$, *hence*

$$A_{min}(\langle s, \textbf{true} \rangle) = 0$$

Encoding Example ($\varphi =$ ***false***)

Question:
$s \models$ *false*

$\xRightarrow{\textit{encode}}$

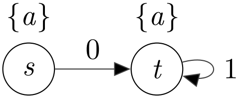
$\boxed{\langle s, \textit{false} \rangle}$

We have the trivial case, as $\langle s, \textit{false} \rangle$ has no hyper-edges, hence

$$A_{min}(\langle s, \textit{false} \rangle) = \infty$$

Model Checking with SDG Example

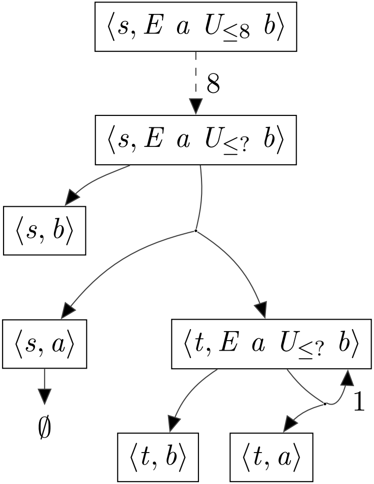
If we take the WKS



and want to determine if

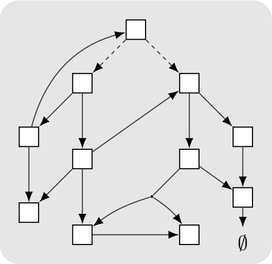
$$s \models E\ a\ U_{\leq 8}\ b$$

then we can encode this as:



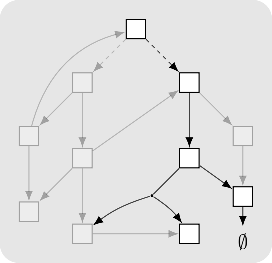
Fixed-Point Algorithms

Global



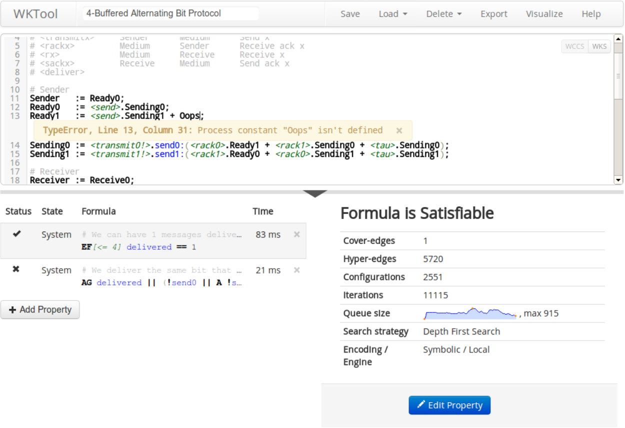
- Up-front construction of SDG.
- Repeated application of F.
- Terminates with A_{min} for all configurations.

Local



- On-the-fly construction of SDG.
- Top-down w. backwards propagation.
- Terminates with A_{min} for the initial configuration.

Model Checking with WKTool



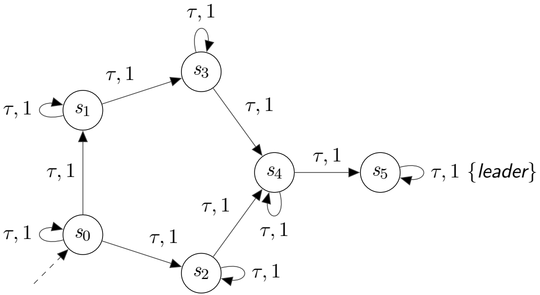
http://wktool.jonasfj.dk/

Experiments

Evaluation of DG vs. SDG and local vs. global for SDG.

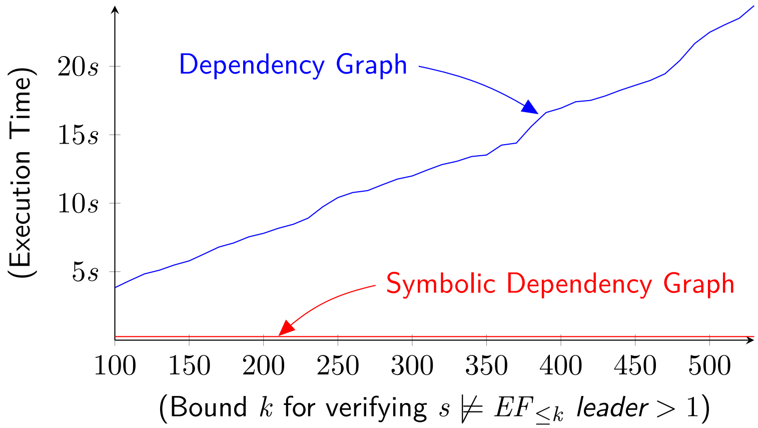
Models:

- Leader Election
- Alternating Bit Protocol
- Task Graph Scheduling problems for 2 processors



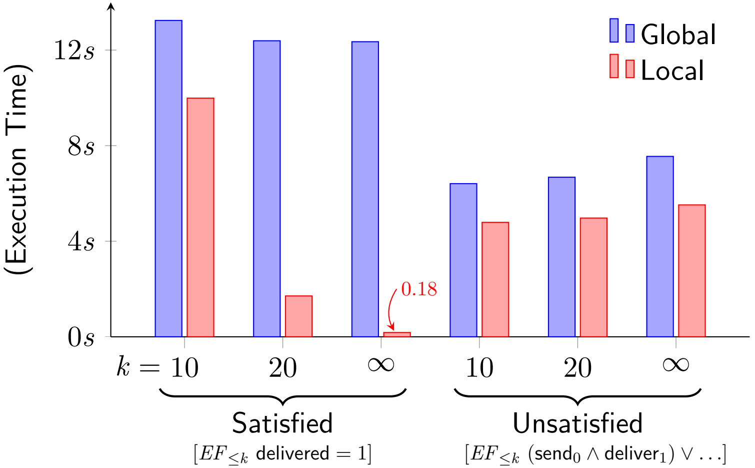
Direct vs. Symbolic (Scaling Bound)

Leader election with DG and SDG encodings using global algorithms.

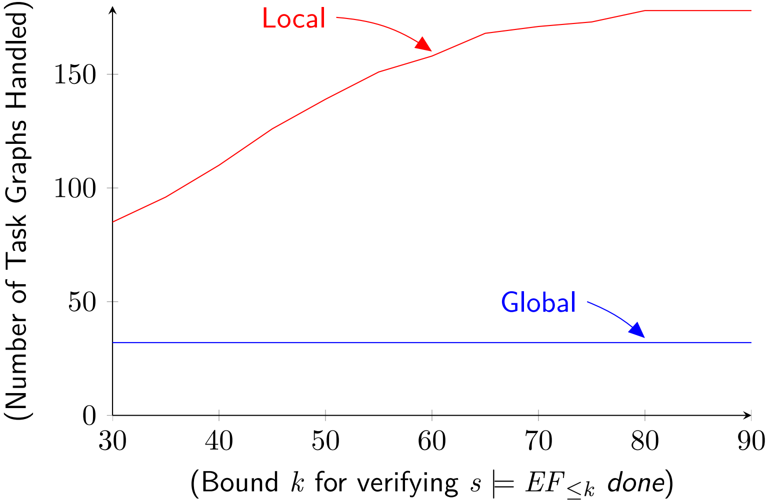


Comparing Global and Local for SDGs

Alternating bit protocol with buffer size 9 (satisfied) and 8 (unsatisfied).

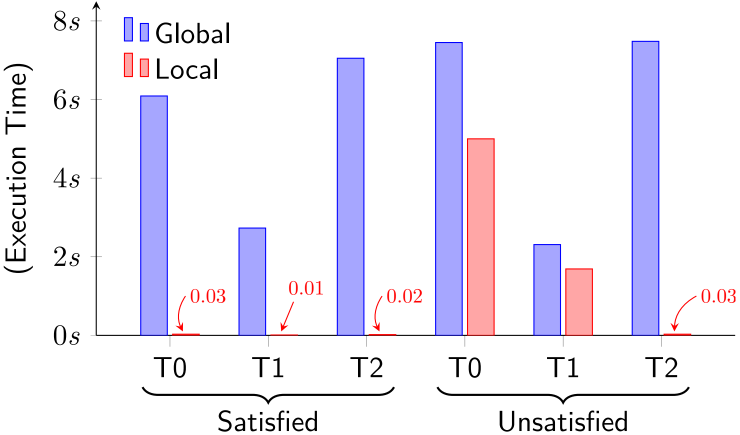


Global vs. Local on 180 Task Graphs



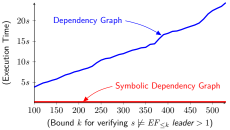
Comparing Global and Local for SDGs

Task graphs T0, T1 and T2 with 5 tasks and nested WCTL properties.



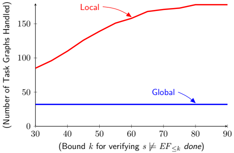
$\left[EF_{\leq 90}\ \left(t_3^{ready} \wedge AF_{\leq 80}\ \textit{done}\right)\right]$ $\left[EF_{\leq 5}\ \left(t_3^{ready} \wedge AF_{\leq 10}\ \textit{done}\right)\right]$

Conclusion



Symbolic Dependency Graphs are advantageous for weighted model checking

Local algorithm can handle larger problems



Future work:

- Alternating fixed-points for *full* WCTL logic.
- Lower-bound constraints on temporal operators.
- Heuristics for search strategy.