KYBERNETIKA — VOLUME 13 (1977), NUMBER 1

# A New Possibility in Bi-Directional Search

JOSEF KOLÁŘ

A new variant of bi-directional heuristic search algorithm is given, which provides a very small number of nodes generated during the search process.

## 1. INTRODUCTION

A great number of problems solved in the field of Artificial Intelligence can be characterized by two distinguished sets: the *set of states* and the *set of operators*. It is known that the process of solution of such problems can be formulated as a search for a certain subgraph in some oriented graph which corresponds to the stated problem. The nodes of this graph correspond to different states, the edges represent the operators. By the *solution of the problem* we mean any sequence of operators, which transforms the given initial state into one of the desired goal states. This sequence defines path from the initial node to some goal node in the corresponding graph.

There are well known algorithms for path-finding in a graph, so the main difficulty lies in the fact that the graph being searched is so large that it is practically impossible to express it explicitly. The graph must be specified by some procedure capable of generating all successors of any given node. This makes it possible to generate only the close neighbourhood of the path being searched for. The successive selection of nodes for expansion (i.e. for successor generation) is controlled by some evaluation function which assigns a nonnegative value to every node. This value is computed using some information about the problem state which corresponds to given node.

The aim of this article is to propose a new variant of bi-directional searching algorithm, which seems to have some good properties with respect to the search effort.

Suppose that a problem $\Lambda$ is formally given by the ordered quadruple $\langle \Xi, \Omega, Q, \xi_v \rangle$, where

$$\Xi = \{\xi_1, \xi_2, \ldots\} \quad \text{is the set of states,}$$

$$\Omega = \{\omega_1, \omega_2, \ldots\} \quad \text{is the set of operators,}$$

where every $\omega \in \Omega$ is a (partial) mapping $\omega : \Xi \to \Xi$, $Q$ is the goal predicate defining the set of goal states of problem $\Lambda$ and $\xi_v \in \Xi$ is the selected initial state of problem $\Lambda$.

By the solution of problem $\Lambda$ we mean the sequence $(\omega_{i1}, \omega_{i2}, \ldots, \omega_{ik})$, which satisfies

$$Q(\omega_{ik}(\omega_{ik-1}(\ldots \omega_{i1}(\xi_v) \ldots))) .$$

If there exists more than one such sequence, it is possible to state an additional requirement that the number $k$, giving the length of this sequence, be as small as possible.

For a given problem $\Lambda = \langle \Xi, \Omega, Q, \xi_v \rangle$ we define the corresponding problem graph as an oriented (multi-) graph $\vec{G} = [H, U, \sigma]$ with a bijective mapping $\varphi : U \leftrightarrow \leftrightarrow \Xi$ and a mapping $\eta : H \to \Omega$, whose edge set $H$ and incidence relation $\sigma \subset H \times \times (U \times U)$ satisfy the following condition:

Let us denote by $\Omega_\xi \subset \Omega$ the set of operators applicable to the state $\xi \in \Xi$ and by $H_u^+ \subset H$ the set of edges leaving the node $u$; then for every $\xi = \varphi(u) \in \Xi$ the mapping $\eta|_{H_u^+}$ is a bijection of the set $H_u^+$ on the set $\Omega_\xi$ and for every $h \in H_u^+$ we have

$$h\,\sigma[u, v] \wedge \eta(h) = \omega \Rightarrow \varphi(v) = \omega(\xi) . \tag{1}$$

We see that the relation of immediate succession $\Gamma$ defined for the nodes of the problem graph $\vec{G}$ will satisfy

$$u \Gamma v \Leftrightarrow \exists \omega \in \Omega(\varphi(v) = \omega(\varphi(u))) . \tag{2}$$

With respect to the one-one correspondence between the set of solutions of problem $\Lambda$ and the set of paths starting in the node $s = \varphi^{-1}(\xi_v)$ and ending in some node $t$ satisfying $Q(\varphi(t))$, it is not necessary to deal with the graph $\vec{G}$ during the search process in its full complexity. We can restrict ourselves to the subgraph induced by the set $\Gamma^*(s)$ of all successors of the initial node $s$ ($\Gamma^*$ is the reflexive-transitive closure of $\Gamma$). For every node in this subgraph we are further interested in the shortest path connecting the initial node $s$ with the given node. From this it follows that for an implicitly stated graph $\langle s, \Gamma \rangle$ we construct explicitly only some part of its minimal tree rooted in $s$.

Pohl studied the possibilities of searching in graphs with just one goal node and in [3] he has formulated the bi-directional search algorithm. His algorithm has good properties with respect to the optimality of the solution path, but it leads to a great number of generated nodes. Pohl states that there is no theoretical guarantee for obtaining good results in bi-directional heuristic search.

We shall describe our version of bi-directional heuristic search (called **SYBIS** as acronym for Symmetric BI-directional Search) and we shall present the main theoretical results derived for this search algorithm. The main difference between **SYBIS** and the other search algorithms is in the fact that the search is directed symmetrically both from the initial and from the goal node. The nodes are expanded in pairs and the evaluation function of the node pairs approximates the length of the solution path containing this node pair.

Let $\vec{G} = [H, U, \sigma]$ be a problem graph of some problem $A = \langle \Xi, \Omega, Q, \xi_v \rangle$. We suppose that every edge $h \in H$ is associated with some real $l(h) > 0$ defining the length of the edge $h$ (the trivial case is $l(h) = 1$ for all $h \in H$). Without loss of generality we can further suppose that $\vec{G}$ has no parallel edges, for every group of parallel edges could be regarded for pur purposes as one edge having the minimal length (if $h\sigma[u, v]$ we can express the length $l(h)$ as $l(u, v)$, too). The edge length induces in the straightforward manner some (partial) distance function $d(u, v)$ defined on those node pairs $[u, v]$ for which $v \in \Gamma^*(u)$. We start now by introducing some symbols which will be used further.

$s, t$ – initial and goal node of the problem graph, resp.

$\tilde{S}, \tilde{T}$ – two sets of nodes called open in the forward and backward direction, resp.

$S, T$ – two sets of nodes called closed in the forward and backward direction, resp.

$g_s(u) = d(s, u)$ – oriented distance from the node $s$ to the node $u$

$g_t(u) = d(u, t)$ – oriented distance from the node $u$ to the node $t$

$h(u, v) = d(u, v)$ – oriented distance between the nodes $u, v$

$f(u, v) = g_s(u) + h(u, v) + g_t(v)$ – the length of solution path containing the nodes $u, v$

$\hat{g}_s(u), \hat{g}_t(u)$ – approximation of the distance $g_s(u)$ and $g_t(u)$, resp. measured on generated part of minimal rooted tree of the problem graph

$\hat{h}(u, v)$ – approximation of the distance $h(u, v)$ (we call $\hat{h}$ the *heuristic function* and $h$ the *perfect heuristic function*)

$\hat{f}(u, v) = \hat{g}_s(u) + \hat{h}(u, v) + \hat{g}_t(v)$ – evaluation function defined for ordered pairs of nodes $[u, v]$

$k_{\min}$ – the shortest solution path length

The **SYBIS** algorithm demands the following input informations:

- the nodes $s$ and $t$ (represented by corresponding states of problem $\Lambda$)
- the procedure representing the immediate succession relation $\Gamma$ (and its inverse $\Gamma^{-1}$)
- the edge length (giving the value $l(u, v)$ for every $v \in \Gamma(u)$ or $u \in \Gamma^{-1}(v)$)
- the procedure representing the heuristic function $\hat{h}$ defined for all pairs of $U \times U$.

The goal of the algorithm is to find the shortest path between the nodes $s$ and $t$ (this path will be represented by the sequence of nodes as we suppose no parallel edges).

The **SYBIS** algorithm proceeds by constructing 4 distinct sets of nodes denoted $S$, $\tilde{S}$, $T$ and $\tilde{T}$. The sets $S$ and $\tilde{S}$ are constructed from the initial node $s$ in the forward direction using the relation $\Gamma$, the sets $T$ and $\tilde{T}$ are constructed from the goal node $t$ in the backward direction using the relation $\Gamma^{-1}$ (the application process of $\Gamma$ or $\Gamma^{-1}$ to some node is called *expansion* of this node). The nodes of the sets $S$ and $T$ are *closed* – all their successors and predecessors, resp. have been generated. The sets $\tilde{S}$ and $\tilde{T}$ contain boundary nodes which are called *open*.

Every node $u \in S \cup \tilde{S}$ is associated with the triplet $[\xi'_u, \hat{g}_s(u), u_p]$, where $\xi'_u$ denotes the representation of the state of the problem $\Lambda$ corresponding to the node $u$, $\hat{g}_s(u)$ is the length of so far obtained minimal path from $s$ to $u$ and $u_p$ is the predecessor of the node $u$ on this minimal path. Analogously, every node $u \in T \cup \tilde{T}$ is associated with the triplet $[\xi'_u, \hat{g}_t(u), u_s]$, where $\xi'_u$ denotes the same as above, $\hat{g}_t(u)$ is the length of so far obtained minimal path from $u$ to $t$ and $u_s$ is the successor of the node $u$ on this minimal path. Both the information $\hat{g}_s(u)$ ($\hat{g}_t(u)$) and $u_p(u_s)$ can be updated during the search, the first is used in computation of the evaluation function values, the second permits the reconstruction of the solution path when the search is finished. Now we give the formal description of the **SYBIS** algorithm.

The **SYBIS** algorithm:

1) We set $S := \{s\}$, $\tilde{S} := \Gamma(s)$, $T := \{t\}$, $\tilde{T} := \Gamma^{-1}(t)$, $k_{\min} := +\infty$ and we set the information associated with the nodes of $S \cup \tilde{S} \cup T \cup \tilde{T}$ appropriately.
2) If $\tilde{S} = \emptyset$ or $\tilde{T} = \emptyset$ the search fails, otherwise we continue.
3) We choose the pair $[u, v] \in \tilde{S} \times \tilde{T}$ such that $\hat{f}(u, v)$ is minimal over the set $\tilde{S} \times \tilde{T}$; we set $S := S \cup \{u\}$, $\tilde{S} := \tilde{S} - \{u\}$, $T := T \cup \{v\}$, $\tilde{T} := T - \{v\}$.
4) For every node $x \in \Gamma(u)$ we perform the following steps:
   a) if $x \in S \cup \tilde{S}$ and $\hat{g}_s(x) > \hat{g}_s(u) + l(u, x)$, then we associate the node $x$ with the new value $\hat{g}_s(x) = \hat{g}_s(u) + l(u, x)$ and with predecessor $u$ and if $x \in S$ we set $S := S - \{x\}$, $\tilde{S} := \tilde{S} \cup \{x\}$;
   b) if $x \notin S \cup \tilde{S}$, then we set $\hat{g}_s(x) = \hat{g}_s(u) + l(u, x)$, associate the predecessor $u$ with it and place the node $x$ into the set $\tilde{S}$,
   c) in remaining cases the node $x$ is left ignored.

5) For every node $x \in \Gamma^{-1}(v)$ we perform the following steps:

a) if $x \in T \cup \tilde{T}$ and $\hat{g}_t(x) > \hat{g}_t(v) + l(x, v)$, then we associate the node $x$ with the new value $\hat{g}_t(x) = \hat{g}_t(v) + l(x, v)$ and with successor $v$ and if $x \in T$ we set $T := T - \{x\}$, $\tilde{T} := \tilde{T} \cup \{x\}$;

b) if $x \notin T \cup \tilde{T}$, then we set $\hat{g}_t(x) = \hat{g}_t(v) + l(x. v)$, associate the successor $v$ with it and place the node $x$ into the set $\tilde{T}$,

c) in remaining cases the node $x$ is left ignored.

6) If $C = (S \cup \tilde{S}) \cap (T \cup \tilde{T}) \neq \emptyset$, we find such node $z \in C$ for which the value $k(z) = \hat{g}_s(z) + \hat{g}_t(z)$ is minimal. We set $k_{\min} := \min(k_{\min}, k(z))$ and if $k_{\min} \leqq \hat{f}(u, v)$ for every $[u, v] \in \tilde{S} \times \tilde{T}$, the search suceeds by finding the solution path of length $k_{\min}$. Otherwise we return to the step 2).

We see that the **SYBIS** search starts by the expansion of the pair $[s, t]$ (we suppose $s \neq t$, for it is trivial to test the case $s = t$ and then the use of any search algorithm does not make any sense). In the step 3) we choose such node pair from the open nodes in opposite directions for which the approximated length of the solution path containing this node pair is minimal. This step, based on originally formulated form of the evaluation function $\hat{f}(u, v)$ plays decisive role in obtaining advantageous properties of the **SYBIS** algorithm.

The step 4) guarantees dynamic updating of the distance approximations of newly generated nodes in the forward direction and forms in this way better approximation of the partial minimal tree rooted in $s$. In the step 5) the same is done in the backward direction. The step 6) contains a little elaborated terminating test of the **SYBIS** algorithm, which guarantees (provided that some assumptions about the heuristic function $\hat{h}(u, v)$ are valid), that the shortest solution path will be found. This step can be simplified to the test $C \neq \emptyset$, if we wish to accelerate the search process by accepting arbitrary (possibly slightly suboptimal) solution path.

## 4. BASIC PROPERTIES OF THE **SYBIS** ALGORITHM

Theoretical analysis of the basic unidirectional heuristic search algorithm **A*** presented by Hart, Nilsson and Raphael in [1] can be adapted for the **SYBIS** algorithm, too. This analysis shows that with respect to admissibility and consistency our algorithm has the same properties as algorithm **A*** mentioned above (for details see [4]). However, **A*** benefits from this comparison because it is more simple and effective than **SYBIS**. The main advantage of **SYBIS** is demonstrated by the "worst case" analysis introduced by Pohl in [2], which gives surprisingly good upper bounds for the number of nodes closed by **SYBIS** during the search process with respect to the **A*** algorithm.

We shall further suppose for simplicity that our problem graph is symmetric (i.e. $\Gamma = \Gamma^{-1}$) and that it is represented by an undirected binary tree (every edge of this tree represents the pair of oppositely oriented edges of problem graph which are incident with the same pair of nodes). Let the length of all edges of this tree be equal to 1. We shall further suppose a slightly modified version of the **SYBIS** algorithm called **SYBIS***, which terminates whenever $(S \cup \tilde{S}) \cap (T \cup \tilde{T}) \neq \emptyset$.

**Theorem 1.** If the heuristic function used by **SYBIS*** is perfect, then **SYBIS*** is optimal, i.e. the sets $S$ and $T$ contain always only nodes of the solution path.

Proof. Let $P = \{x_0, x_1. x_2, \ldots, x_{k-1}, x_k\}$, $x_0 = s$, $x_k = t$ be the solution path. We prove our statement by induction.

1) We have $x_1 \in \Gamma(s)$ and $x_{k-1} \in \Gamma^{-1}(t)$, so

$$\hat{f}(x_1, x_{k-1}) = 1 + k - 2 + 1 = k$$

For every $u \in \Gamma(s)$, $v \in \Gamma^{-1}(t)$, $[u, v] \neq [x_1, x_{k-1}]$ we have

$$\hat{f}(u, v) = 1 + h(u, v) + 1 = h(u, v) + 2 \geqq k + 2 > k\,,$$

for at least one of the nodes $u$, $v$ is not on $P$. Consequently, in the first step the pair $[x_1, x_{k-1}]$ will be selected for expansion.

2) Suppose in the $i$-th step the pair $[x_i, x_{k-i}]$ was selected. We prove that the following pair selected will be $[x_{i+1}, x_{k-i-1}]$ (unless the algorithm is finished). We have

$$\hat{f}(x_i, x_{k+i}) = \hat{f}(_{i+1}, x_{k-i-1}) = k$$

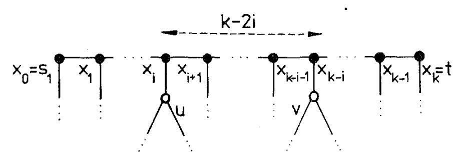


**Fig. 1.** A search with perfect heuristic function.

and we can test only the pairs of $\Gamma(x_i) \times \Gamma^{-1}(x_{k-i})$. Let $[u, v] \in \Gamma(x_i) \times \Gamma^{-1}(x_{k-i})$, $[u, v] \neq [x_{i+1}, x_{k-i-1}]$, then we have

$$\hat{f}(u, v) = \hat{g}_s(u) + h(u, v) + \hat{g}_t(v) = i + 1 + h(u, v) + i + 1 \geqq$$
$$\geqq 2i + 2 + k - 2i = k + 2 > k$$

(the value of $h(u, v)$ is obvious from Fig. 1). □

Remark 1. We can generalize the evaluation function $\hat{f}(u, v)$ to the form 

$$\hat{f}(u, v) = \alpha\, \hat{g}_s(u) + \beta\, \hat{g}_t(v) + (1 - \alpha - \beta)\, \hat{h}(u, v),$$

where $0 < \alpha < 1$, $0 < \beta < 1$, $0 < \alpha + \beta < 1$. It can be shown that under appropriate conditions for the values of $\alpha$ and $\beta$ Theorem 1 remains valid for the **SYBIS*** algorithm using the generalized evaluation function.

As a rule, we do not know the perfect heuristic function $h(u, v)$. In the best case we can only give some estimate of error of our function $\hat{h}(u, v)$ in the form

$$h(u, v) - \varepsilon \leqq \hat{h}(u, v) \leqq h(u, v) + \varepsilon . \tag{3}$$

We can state the question, as to how the magnitudes $|S|$ and $|T|$ at end of **SYBIS*** depend on the value of $\varepsilon$. We shall try to give some upper bounds for these values.

**Theorem 2.** Let $\hat{h}(u, v)$ be any heuristic function satisfying (3) for all pairs $[u, v]$ and some fixed $\varepsilon \geqq 0$. Let the distance of the node $u$ and $v$ from the solution path be $w_1$ and $w_2$, respectively. Then

$$w_1 + w_2 > \varepsilon \tag{4}$$

implies that neither the pair $[u, v]$ nor the pair $[v, u]$ will be selected for expansion by **SYBIS***.

Proof. In every step of **SYBIS*** the sets $\tilde{S}$ and $\tilde{T}$ contain just one node of the solution path $P$. Let this be the nodes $x_i$ and $x_{k-j}$, respectively. Then $0 < i < k - j < k$ and we have

$$\hat{f}(x_i, x_{k-j}) = i + \hat{h}(x_i, x_{k-j}) + j \leqq i + (k - i - j + \varepsilon) + j = k + \varepsilon$$

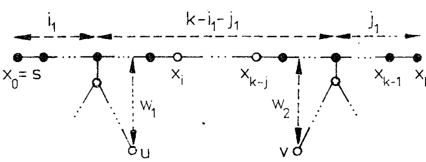


**Fig. 2.** Expansion of nodes situated off the solution path.

On the other hand, for the pair $[u, v]$ (see Fig. 2) we have

$$\begin{aligned}\hat{f}(u, v) &= (i_1 + w_1) + \hat{h}(u, v) + (j_1 + w_2) \geqq \\ &\geqq i_1 + w_1 + (k - i_1 - j_1 + w_1 + w_2 - \varepsilon) + j_1 + w_2 = \\ &= k + 2(w_1 + w_2) - \varepsilon > k + \varepsilon .\end{aligned}$$

Consequently, the pair $[x_i, x_{k-j}]$ is available with better value of evaluation function and the pair $[u, v]$ will not be selected. □

**Remark 2.** The previous theorem excludes from the selection only the ordered pairs $[u, v]$ and $[v, u]$ and not the individual nodes $u$, $v$.

Let us suppose now that we have some node $u$ of the solution path $P$. From Theorem 2 it follows that this node could be selected for expansion with some node $v$ situated in the distance less than, or in the worst case equal to, $\varepsilon$ from the path $P$. This leads immediately to a conclusion which seems to be rather pessimistic:

**SYBIS*** *algorithm can expand nodes in distance up to* $\varepsilon$ *from the solution path.*

The same is true for the **A*** algorithm, but we shall show that **SYBIS*** – in contrast with **A*** – cannot expand all nodes up to this distance.

**Theorem 3.** Let $\hat{h}(u, v)$ be any heuristic function satisfying (3) for all pairs $[u, v]$ and for some fixed even $\varepsilon \geqq 0$. Suppose the set $S$ and $T$, just before termination of **SYBIS***, contains $k_1$ first nodes and $k_2$ last nodes of the solution path, respectively. Then we have

(5) $$|S| \leqq k_1\, 2^{\varepsilon/2} + k_2\, 2^{(\varepsilon/2)-1}$$

(6) $$|T| \leqq k_1\, 2^{(\varepsilon/2)-1} + k_2\, 2^{\varepsilon/2}$$

**Table 1.**

| Distance from the solution path | Maximum number of expanded nodes in given distance | |
|---|---|---|
| | belonging to $S$ | belonging to $T$ |
| 0 | $k_1$ | $k_2$ |
| 1 | $k_1$ | $k_2$ |
| 2 | $2k_1$ | $2k_2$ |
| 3 | $4k_1$ | $4k_2$ |
| . | . | . |
| . | . | . |
| . | . | . |
| $\varepsilon/2 - 1$ | $2^{\varepsilon/2-2}k_1$ | $2^{\varepsilon/2-2}k_2$ |
| $\varepsilon/2$ | $2^{\varepsilon/2-1}k_1$ | $2^{\varepsilon/2-1}k_2$ |
| $\varepsilon/2 + 1$ | $2^{\varepsilon/2-2}k_2$ (!) | $2^{\varepsilon/2-2}k_1$ (!) |
| . | . | . |
| . | . | . |
| . | . | . |
| $\varepsilon - 1$ | $k_2$ | $k_1$ |
| $\varepsilon$ | $k_2$ | $k_1$ |

Proof. Theorem 2 states which pairs $[u, v]$ have the possibility to be ever expanded by **SYBIS***. Let us imagine the worst case, when expanded nodes are as far from the solution path as possible. The $k_1$ nodes of $S \cap P$ could have been expanded (in the worst case) in pairs with $k_1$ nodes of the set $T$, situated in the distance $\varepsilon$ from the solution path. In the same way, $k_1$ nodes of $S$ in the distance 1 from $P$ could form the pairs with $k_1$ nodes of $T$ in the distance $\varepsilon - 1$ from $P$, etc. Numbers obtained in this way can be ordered in the preceeding Table 1. By computing the sums of the second and the third columns we obtainthe upper bounds presented in (5) and (6), respectively. □

**Remark 3.** In the previous theorem we suppose that the problem graph is a binary tree with root $s$ and goal node $t$ situated in some distance $k$ from $s$. This means that the goal node $t$ should have its degree equal to 3 and not to 2 as we have used above. This difference is in fact negligible, for it influences only the factor $k_2$ in Table 1 changing it to $k_2 + 1$ in all but the first and the last rows. The binary tree serves us only as a conveniently simplified model of actual problem graph and we can consequently suppose that the search by **SYBIS*** advances from two roots of opposite binary trees.

**Remark 4.** Since the problem graph is a tree, the nodes are never reopened using the steps 4a) or 5a) of **SYBIS*** and the sets $S$ and $T$ will always contain the same number of nodes. Consequently, the right-hand sides of (5) and (6) could be interchanged. The worst case is obviously obtained when $k_1 = k_2$ or $|k_1 - k_2| = 1$.

**Theorem 4.** Let $\hat{h}(u, v)$ be any heuristic function satisfying (3) for all pairs $[u, v]$ and for some fixed odd $\varepsilon > 0$. Suppose the set $S$ and $T$, just before termination of **SYBIS***, contains $k_1$ first nodes and $k_2$ last nodes of the solution path, respectively. Then we have

$$|S| = |T| \leqq (k_1 + k_2)\, 2^{(\varepsilon + 1)/2}\,. \tag{7}$$

Proof. Using Table 1 from the proof of Theorem 3 divided for odd $\varepsilon$ in two equal parts between the $((\varepsilon - 1)/2)$-th and the $((\varepsilon + 1)/2)$-th row we obtain the desired result. □

**Theorem 5.** Let $\hat{h}(u, v)$ be any heuristic function satisfying (3) for all $[u, v]$ and for some fixed $\varepsilon \geqq 0$. Then the upper bound $N$ for the number of nodes closed by **SYBIS*** during the solution search given by the expression

$$N = 2 + (k - 1)\, 2^{\varepsilon_1}\,, \tag{8}$$

where $\varepsilon_1$ is the integer part of $(\varepsilon + 3)/2$ and $k \geqq 2$ is the solution path length.

Proof. We can suppose that the solution path is found at the moment, when the set $S$ contains its first $k_1$ nodes and the set $T$ contains its last $k_2$ nodes. The last expanded pair will contain one or both end nodes of available parts of the solution path. Obviously, we have $k - 2 \leqq k_1 + k_2 \leqq k - 1$. The worst case is obtained (see Remark 4), when

$$k_1 = k_2 = \frac{k-1}{2} \quad \text{for odd } k$$

and

$$k_1 = k_2 + 1 = \frac{k}{2} \quad \text{for even } k\,.$$

Using (5) and (7) we obtain

$$|S| = |T| \leqq \tfrac{3}{2}(k-1)\, 2^{(\varepsilon/2)-1} \quad \text{for even } \varepsilon, \text{ odd } k\,,$$

$$|S| = |T| \leqq (\tfrac{3}{2}k - 1)\, 2^{(\varepsilon/2)-1} \quad \text{for even } \varepsilon, \text{ even } k\,,$$

$$|S| = |T| \leqq (k-1)\, 2^{(\varepsilon+1)/2} \quad \text{for odd } \varepsilon\,.$$

Since $|S \cup T| = 2|S|$, the upper bound $N$ is given by the maximum of previous values (represented by the last expression) multiplied by 2. The leading term 2 expresses the possible concluding expansion of both the nearest nodes. □

It is well known that for the **A*** algorithm the upper bound is given by

$$N' = k\, 2^{\varepsilon} \tag{9}$$

and for other algorithms this value remains roughly the same. Using (8) we obtain

$$\lim_{\varepsilon \to +\infty} \frac{N}{N'} = 0\,.$$

This implies that we can expect the number of nodes closed by **SYBIS*** to be much less than the number of nodes closed by **A***.

## 5. EXPERIMENTAL RESULTS

The **SYBIS*** algorithm was tested by searching for a solution of the well known "15-puzzle". The results are given in Table 2.

From Table 2 we obtain summary results presented in Table 3.

Although the statistical importance of the presented pattern is relatively small, we see that our theoretical results are in close correspondence with practical experiments.

Table 2.

| Problem No. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | *k* | 27 | 33 | | 27 | | 37 | 31 | 24 | 42 | |
| | *A* | 26 | 83 | | 226 | | 207 | 27 | 38 | 168 | |
| A* | *B* | 34 | 88 | | 238 | | 224 | 38 | 42 | 170 | |
| | *C* | 60 | 171 | 500 | 464 | 500 | 431 | 65 | 80 | 338 | 500 |
| Algorithm | *k* | 27 | 27 | 24 | 27 | 28 | 35 | 31 | 24 | | 29 |
| | *A* | 33 | 47 | 39 | 34 | 38 | 60 | 36 | 43 | | 38 |
| **SYBIS*** | *B* | 28 | 44 | 35 | 27 | 37 | 52 | 32 | 35 | *) | 37 |
| | *C* | 61 | 91 | 74 | 61 | 75 | 112 | 68 | 78 | | 75 |

$k$ = path length, $A$ = number of open nodes, $B$ = number of closed nodes, $C$ = total number of generated nodes.

*) time exhausted.

Table 3.

| Algorithm | Number of problems | Solved (%) | Total of nodes | Closed nodes per problem | Pene- trance |
|---|---|---|---|---|---|
| A* | 10 | 70 | 3109 | 119 | 0·122 |
| SYBIS* | 10 | 90 | 995 | 36 (!) | 0·364 |

## 6. CONCLUSION

We have introduced a new bi-directional search algorithm and its basic theoretical properties together with some experimental results. Our algorithm is superior in comparison to the **A*** algorithm of Hart, Nilsson and Raphael with respect to the number of nodes closed during the search. On the other hand, the computation time comparicon shows that the **A*** algorithm is shorter due to the relative complexity of the **SYBIS*** algorithm. Our further effort will be devoted to possible improvements leading to less complexity and shorter computation time of **SYBIS***.

(Received March 3, 1976.)

REFERENCES

[1] P. E. Hart, N. J. Nilsson, B. Raphael: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. on SSC *4* (1968), 2, 100—107.

[2] I. Pohl: First results on the effect of error in heuristic search. Machine Intelligence *5* (eds. B. Meltzer, D. Michie), Edinburgh Univ. Press, Edinburgh 1970.

[3] I. Pohl: Bi-directional search. Machine Intelligence *6*, Edinburgh Univ. Press, Edinburgh 1971.

[4] J. Kolář: K možnostem zvýšení efektivnosti heuristického řešení úloh (Some possibilities of optimization of the heuristic search). Technical University, Prague 1975 (dissertation).

*RNDr. Josef Kolář, katedra počítačů elektrotechnické fakulty ČVUT (Department of Computers — Electroengineering Faculty — Czech Technical University), Karlovo nám. 13, 120 00 Praha 2. Czechoslovakia.*