

PEMECAHAN MASALAH DENGAN MENCARI

Pemecahan masalah
AGEN

algorithms

interna

2.4.7—that is,

10.

solutions

a *fixed*

case—where

$O()$

3.1 AGEN pemecahan masalah

RO-

Manian, menikmati pemandangan, menikmati kehidupan malam (seperti itu

tujuan

TUJUAN FORMULASI

formulasi tujuan,

left.”

banyak,

PROBLEM
FORMULATION

Formulasi masalah

agen akan mempertimbangkan tindakan pada tingkat mengemudi dari satu kota utama

particular

adalah mengingat

which

2.3—then

berikut

agen dengan beberapa

Pilihan segera nilai yang tidak diketahui dapat memutuskan apa yang harus dilakukan dengan terlebih dahulu memeriksa tindakan yang akhirnya menyebabkan negara nilai dikenal.

Sebuah

—

one

a fixed

on.

actions.

PENCARIAN
SOLUTION
EXECUTION

OPEN-LOOP

MASALAH
AWAL STATE

In(A

```

SIMPLE-PROBLEM-SOLVING-AGENT(percept)
  seq.

  ATE(state)

  ← FORMULATE-GOAL(state)
  ←
  SEARCH(
    ← FIRST(seq)
    ← REST(seq)
    action
  )

```

•

S,

APPLICABLE

TRANSITION MODE:

SUCCESSOR

S.

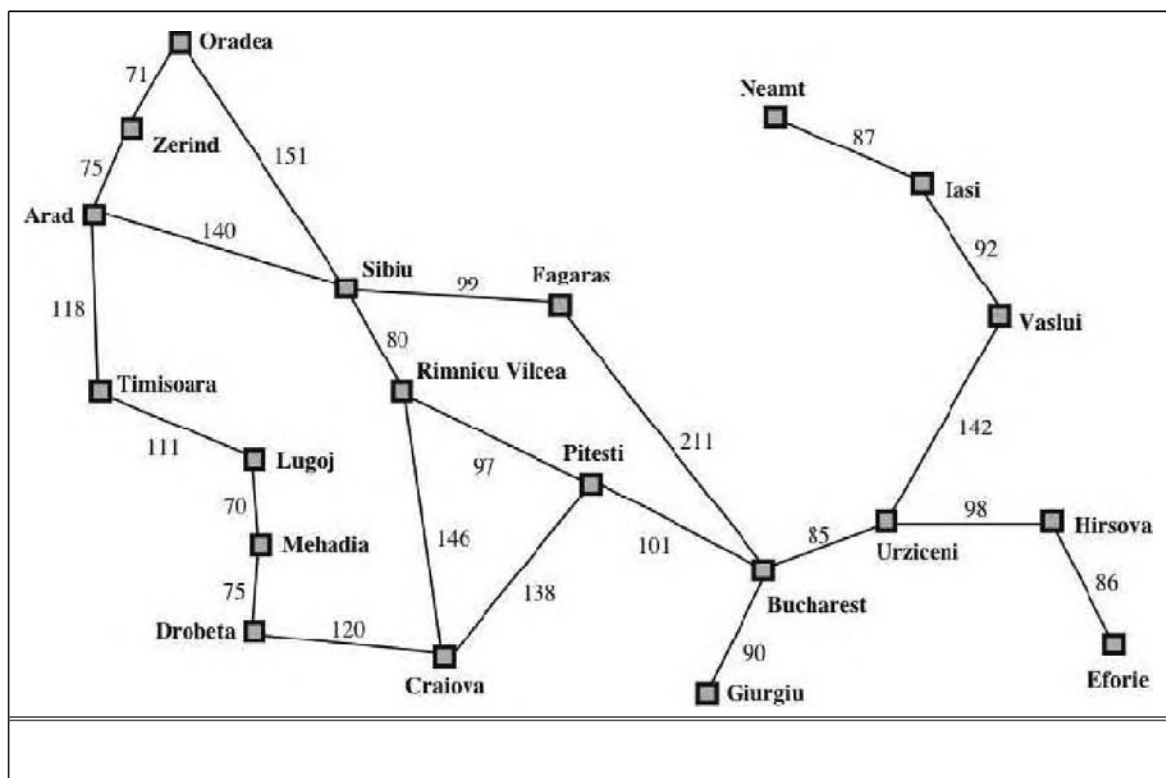
sct

•

$$\text{ULT}(a, s)$$

RESL

Da



PAIN BIAYA

LANGKAH BIAYA

sum

s'

$c(s, s')$

nonnegative.

SOLUSI OPTIMAL

model—an

In(Arad),

ABSTRAKSI

enforcement

Rimnicu

sah

berguna

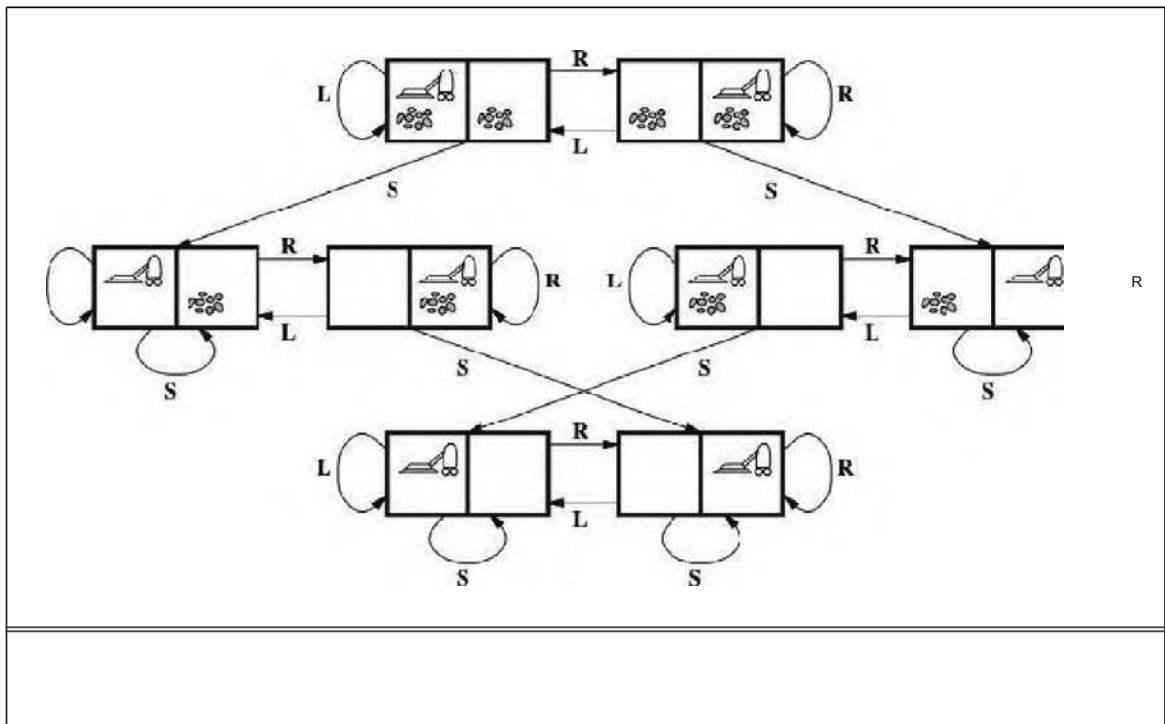
3.2 CONTOH

oleh dunia nyata

TOY MASALAH

REAL-WORLD
MASALAH

complete



2

$n \cdot 2^n$

Mengisap.

Naik

Turun.

Kiri kanan,

Kiri

Kanan

mengisap

-		
	3	1

instance 8-puzzle.

from 4)

are

ruled out

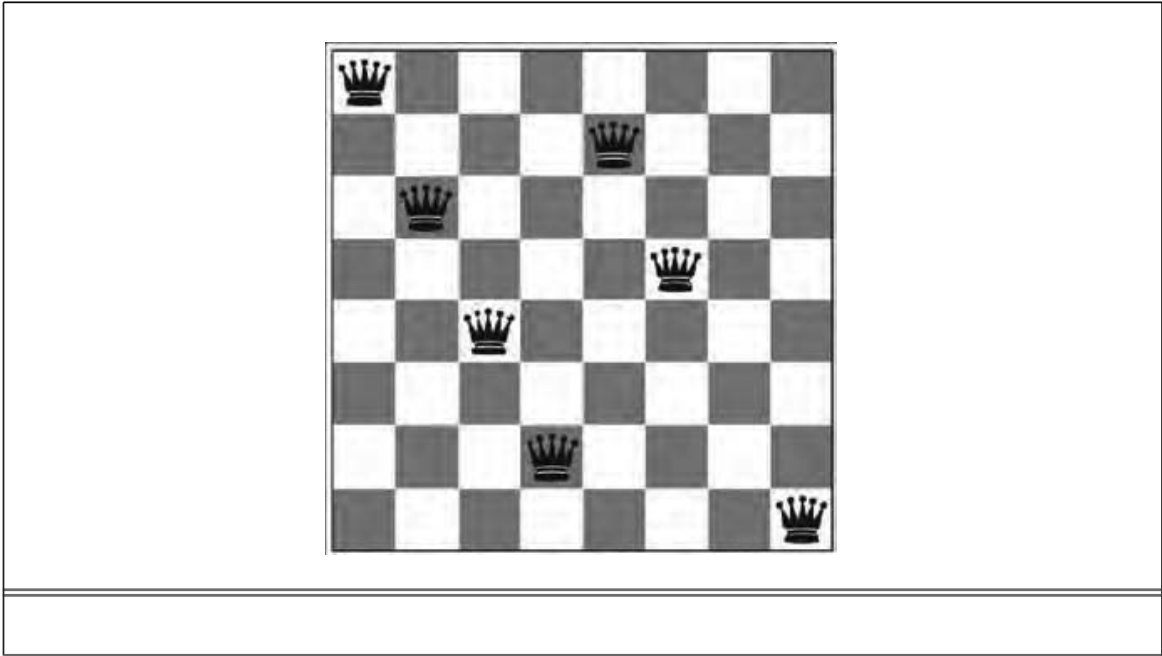
SLIDING-BLOCK
puzzle

10

algorithms.

8-QUEEN S MASALAH

8-queens



TAMBAHAN
FORMULATION

menambah

COMPLETE STATE
FORMALIZATION

either

64

71

71

10^{400}

4.1

4

the

5—50

ROUTE FINDING

them.

within-airport

TOURING MASALAH

In *ed*(*{Bucharest}*);
In(*Vaslui*), *Visited*(*{Bucharest, Urziceni, Vaslui}*),
visited.

TRAVELING
SALESPERSON
PROBLEM

VLSI LAYOUT

solving.

SCOOT NAVIGASI

AUTOMATIC
MAJELIS
SEQUENCING

searchSebuah

PROTEIN DESIGN

3.3 Mencari Solusi

search tree
node

In(Arad).

menembangkan
pembangkit induk
node
anak node

Dalam (Arad)

In(Sibiu), In(Timisoura), In(Zerind).

In(Arad).

leaf node

Dalam (Fagaras), di (Oradea), In(Rimnicu Vilcea), Zerind.
set

frontier
daftar open

cari strategi

so-called strategy.

In(Arad)

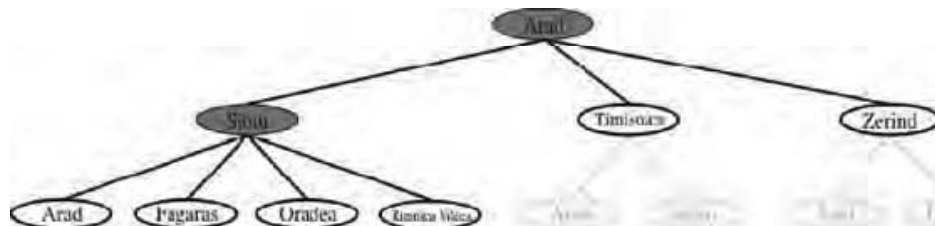
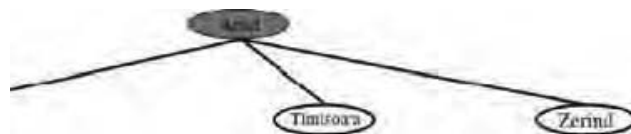
repeated negara
path loopy

tak terbatas

limitother

Arad-Zerind-Oradea-Sibiu

keadaan awal



TREE-SEARCH(

GRAPH-SEARCH(*problem*)

node.

GRAPH-SEARCH

GRID RECTANGULAR

7
4



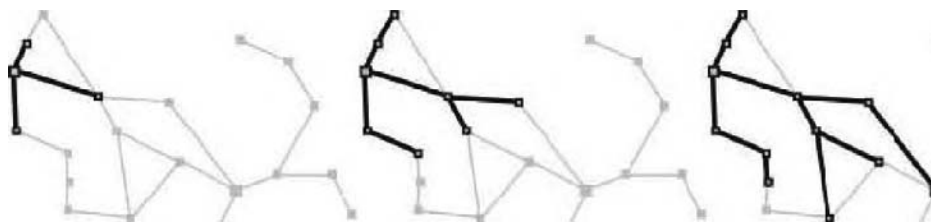
As

algoritma itu, oget sejarah mereka ditakdirkan untuk mengulanginya.

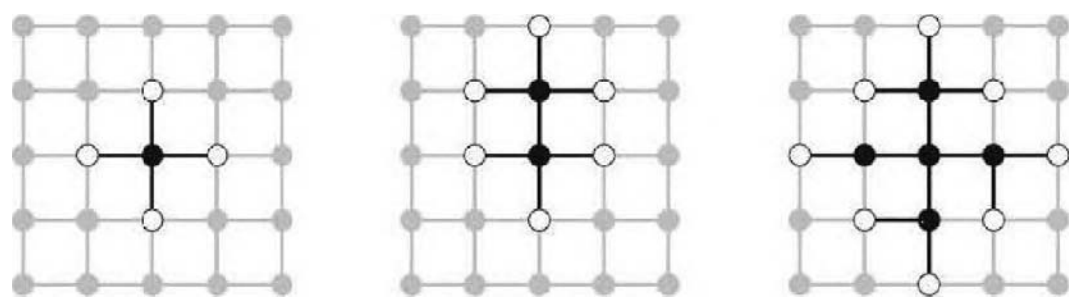
EXPLORED set

TERTUTUP UST

SEPARATOR



3.2.



separation

rectangular-grid

In

seara sistematizata

n

■ n .

$n.PARENT$:

di

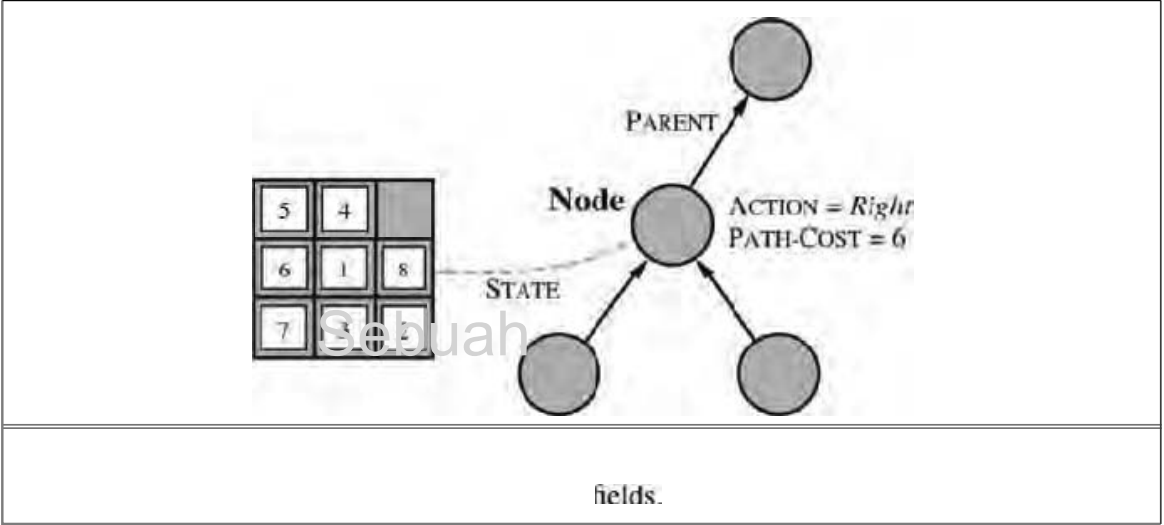
$n.ACTION$:

$n.PATH-COST$:

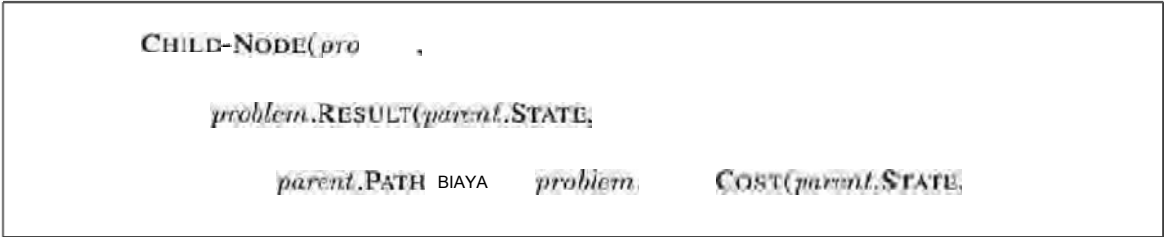
traditionally

$g(n)$,

from



CHILD-NODE



Figure

```

EMPTY?( antre)
•P OP(queue)
INSERT(element,

```


FIFO QUEUE

sulung

LIFO QUEUE

PRIORITY ANTRE

74).

Urziceni,Vaslui,Bucharest }.

CANONICAL BENTUK

KELENGKAPAN

OPTIMALITY

TIME KOMPLEKSITAS

SPACE KOMPLEKSITAS

(links). $|E$

structure

implisit

Bercabang FAKTOR

KEDALAMAN

CARI COAT

TOTAL BIAYA

usage—or

from Arad

untuk

16.

3.4 STRATEGI CARI

Kurang informasi CARI
CARI BLIND

memesan

INFORMASI CARI
MENDENGAR ISM CARI

BREADTH-F HST
PENCARIAN

mereka

dangkal

tweak

dihasilkan

menyelesaikan-jika

d,

b

dangkal

optimal

problem.GOAL-TEST(*simul.* **SOLUTION(*node*)**)

EMPTY?(*frontier*)

$$\text{simpul} \leftarrow \text{POPI}(\text{frontier}) f^*$$

menamibankan. to explored

.ACTIONS masalah tindakan (n ode.

 $\leftarrow \text{CHILE-NODE}(\text{problem},$

anak. dieksplorasi perbatasan

anak)

perbatasan

anak, perbatasan)

 \mathbb{Z}

b

ini

b d

b b^2 b^3 $b^a = O(b^d)$

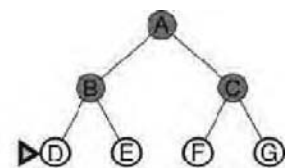
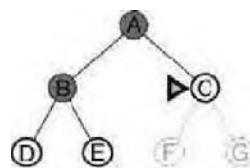
was

 $O(t^{1/2+\epsilon}).)$

dieksplorasi

$$b$$
 $O(b^2)$

dieksplorasi

 $O(t^2)$ 

sehingga kompleksitas ruang $O(b)$, yaitu, didominasi oleh ukuran perbatasan. Switching untuk

beralih bisa biaya banyak waktu.

Kompleksitas eksponensial terikat untuk

berbagai nilai kedalaman solusi

pencarian pertama dengan branching factor

dihasilkan per detik dan bahwa node membutuhkan

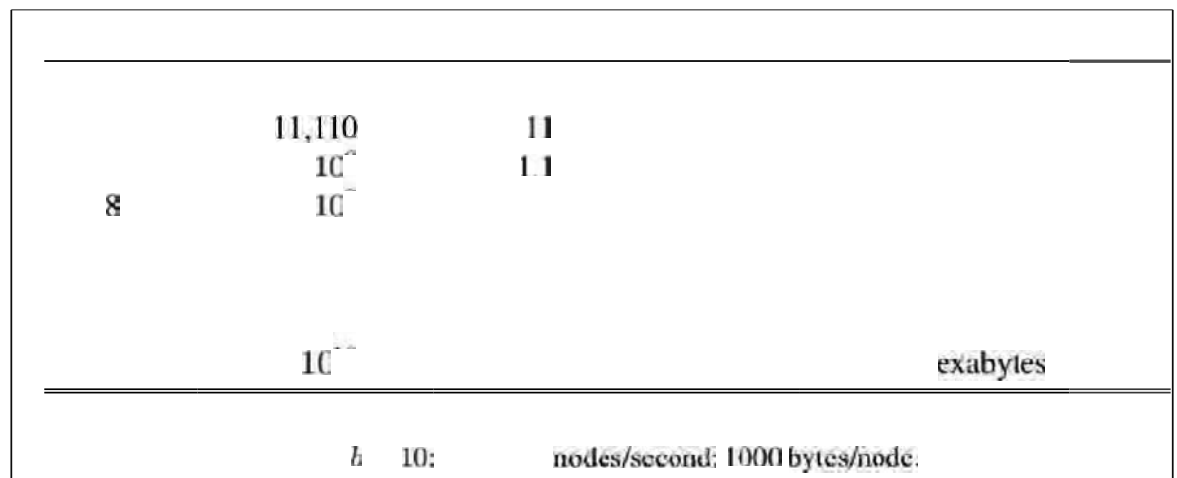
muat kira-kira dalam asumsi ini (memberi atau mengambil faktor 100) ketika dijalankan pada komputer pribadi modem.

 $O(b^{\frac{1}{2}})$ $O(t)$

d. waktu dan memori yang dibutuhkan untuk breadth- sebuah

10. Tabel mengasumsikan bahwa juta node dapat

1000



Two pelajaran yang dapat dipelajari dari Gambar 3.13. Pertama,

memory requirements

first

execution time

untuk solusi

take.

breadth-first

(or

memang

kurang informasi pencarian) untuk menemukan itu. Secara umum, *exponential-complexity*

m, *exponential-complexity*

Ketika semua biaya langkah yang sama, pencarian luas-pertama adalah optimal karena selalu memperluas

simpul tidak dikembangkan. Dengan perpanjangan sederhana, kita dapat menemukan sebuah algoritma yang optimal

dengan fungsi langkah-biaya. Alih-alih memperluas simpul dangkal, memperluas node dengan

TE $g(n).$

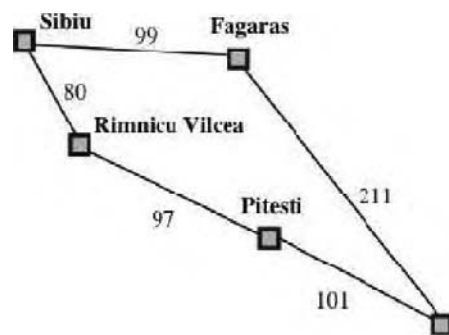
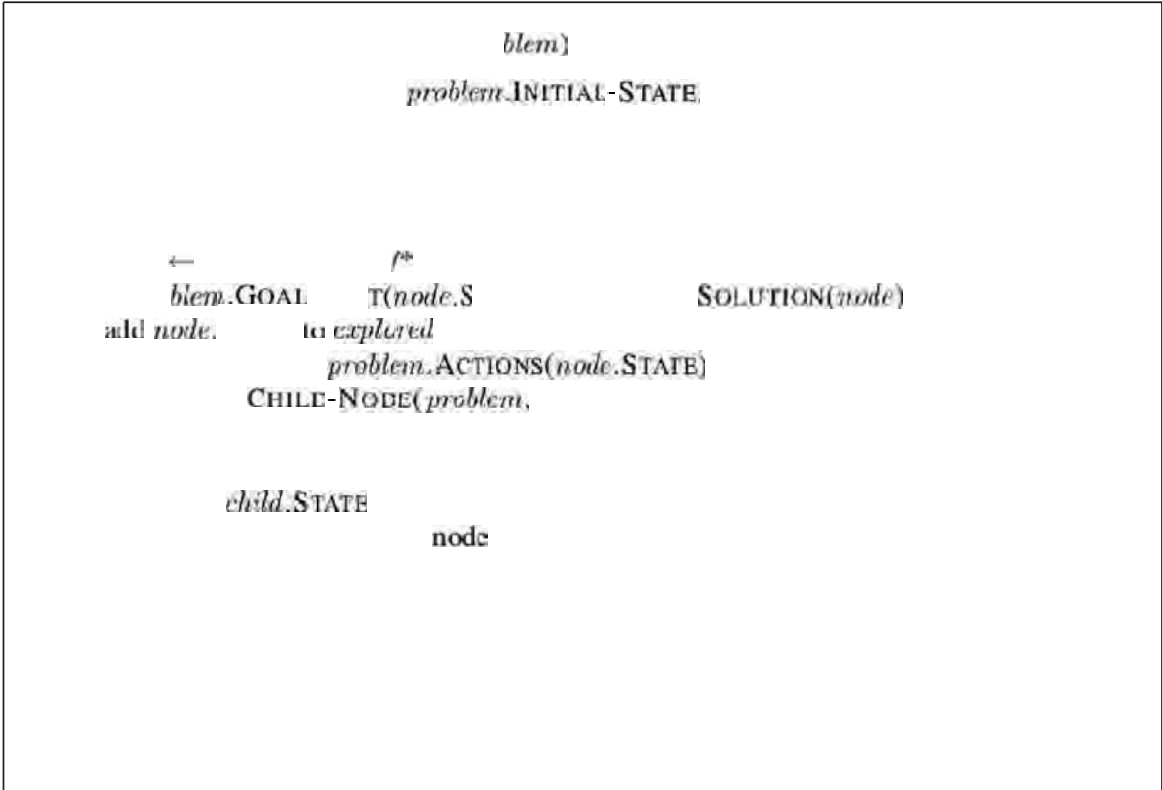
diperintahkan oleh q. Algoritma ini ditunjukkan pada Gambar 3.14.

perbedaan dari pencarian luas-pertama. Yang pertama adalah bahwa tes tujuan diterapkan ke node ketika

expansion

graph-search

daripada ketika pertama kali dihasilkan. Alasannya adalah bahwa node tujuan pertama yang



Rimnicu

Bucharest

310.

$$80+97+101=$$

g-cost

n'
3.9:

n' *n.*
g-cost *n*

uniform-cost pencarian memperluas node dalam urutan jalur optimal biaya.

jumlah

noop

e.

E *d.* *C** *solution,*
O(l² +) *ε.* *b².*

$$b^{1+\lceil C^*/\epsilon \rceil} \text{ atau } b^{d+1}.$$

d

3.4.3

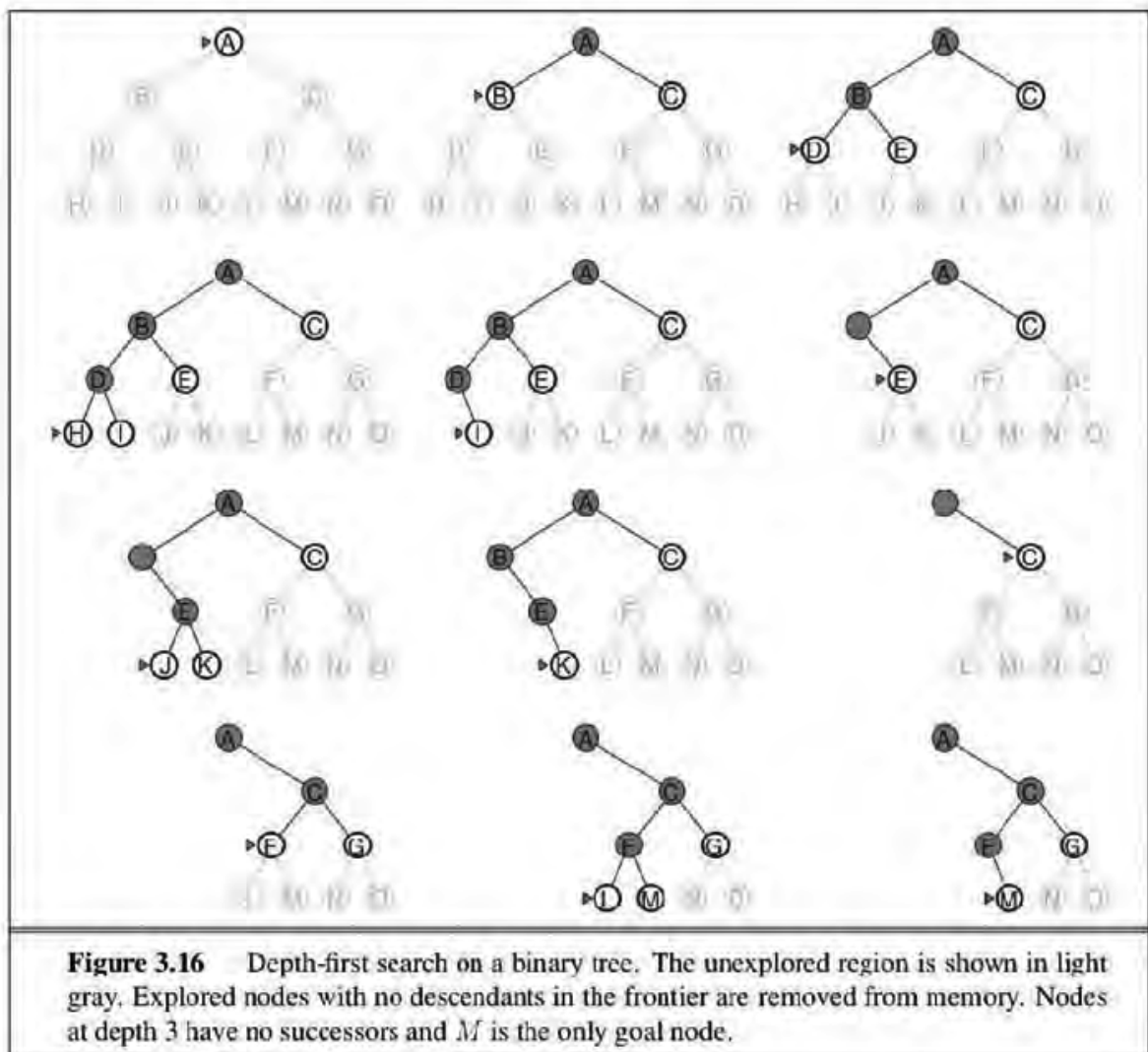
DEPTH-FIRST
SEARCH

terdalam

3.17.]

NoOp.

*C**



properties

depth-first nonoptimal.
subtree return

$$O(b^m)$$

$$m$$

$$3.16.)$$

$$m.$$

$$O(bm)$$

$$9).$$

backtracking
SEARCH

$$O(m)$$

$$O(bm),$$

$$O(m)$$

DEPTH-LIMITED
PENCARIAN

$$\ell.$$

$$\ell$$

$$\ell$$

$$>$$

$$O(b^{\hat{\ell}})$$

$$O(b\ell);$$

$$\ell = \infty.$$

$$20$$

$$\ell$$

$$it$$

DEPTH-LIMITED-SEARCH

membatasi)

RECURSIVE-DLS(MAKE-NODE(*problem*.INITIAL-STATE), *masalah*, *batasan*)

(Node, masalah, batas) node.

problem.GOAL-TEST(*node*)

SOLUTION(*node*)

limit =

cutoff

cutoff_occurred?

←

anak

problem.ACTIONS(*node*.STATE)

tindakan

CHILD-NODE(*masalah*, *simpul*, *tindakan*)

hasil

RECURSIVE-DLS(*anak*, *masalah*, *Memancarkan* - 1)

Hasilnya = *cutoff*

cutoff terjadi?

←

hasil

≠

kegagalan

hasil

cutoff_occurred?

memotong

kegagalan

o IAVETFF

ITERATIVE

Memperdalam CARI

d,

ITERATIVE-DEEPENING-SEARCH

```

function ITERATIVE-DEEPENING-SEARCH(problem) returns a solution, or failure
  for depth = 0 to  $\infty$  do
    result  $\leftarrow$  DEPTH-LIMITED-SEARCH(problem, depth)
    if result  $\neq$  cutoff then return result

```

Figure 3.18 The iterative deepening search algorithm, which repeatedly applies depth-limited search with increasing limits. It terminates when a solution is found or if the depth-limited search returns *failure*, meaning that no solution exists.

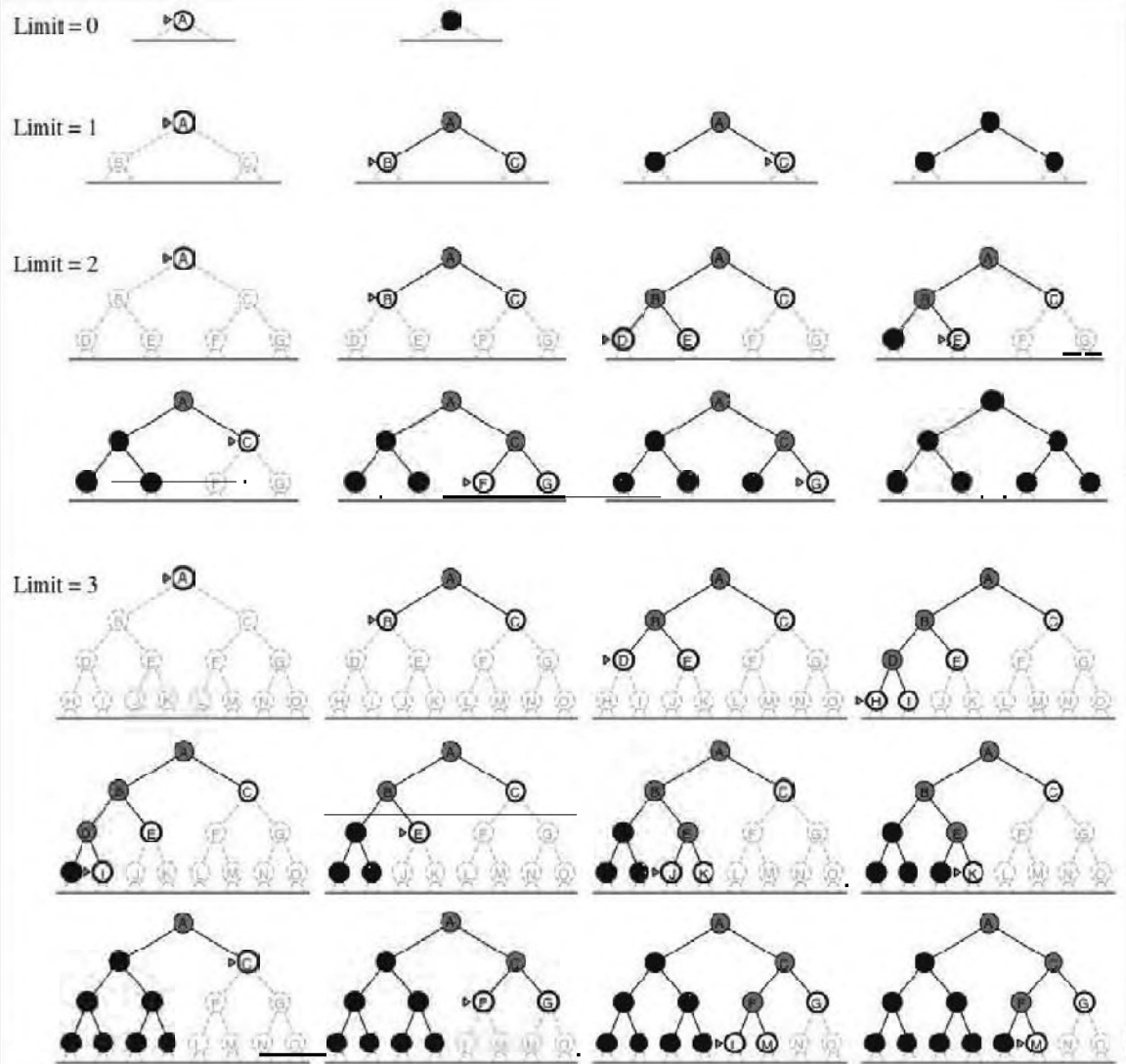


Figure 3.19 Four iterations of iterative deepening search on a binary tree.

d times.

$$N(\text{IDS}) = (D)b + (d-1)b^2 + \dots + (1)b^d$$

$O(b^d)$ = asymptotically
generating

$$d =$$

+

.

runs

Secara umum, pendalaman berulang akan

metode pencarian uninformed yang disukai ketika ruang pencarian besar dan kedalaman solusi

is not dikenal.

ITERATIF
LENGTHENING
SEARCH

$$b^{d/2} + b^{(d-1)/2} + \dots + b^{1/2} + b^{0/2}$$

$$d =$$

$$3. \quad b =$$

$$O(b^{d/2}).$$

$$O(b^{d/2}).$$

the

3,5 INFORMASI (HEURISTIK) STRATEGI CAR!

INFORMASI CAR!

BEST-FIRST PENCARIAN

EVALUASI
FUNGSI

HEURISTIC
FUNGSI

CARI **BEST-FIRST**
SERAKAH

STRAIGHT-LINE
JARAK

search.

shows.

$h(n)$; n

$h(n)$
 $h(n)$

constraint; nonnegative;
 (n)

$h(n)$, Romania;
route-finding h_{SLD} .

$h_{SLD}(In(Arail))$

h_{SLD}

h_{SLD}

h_{SLD}

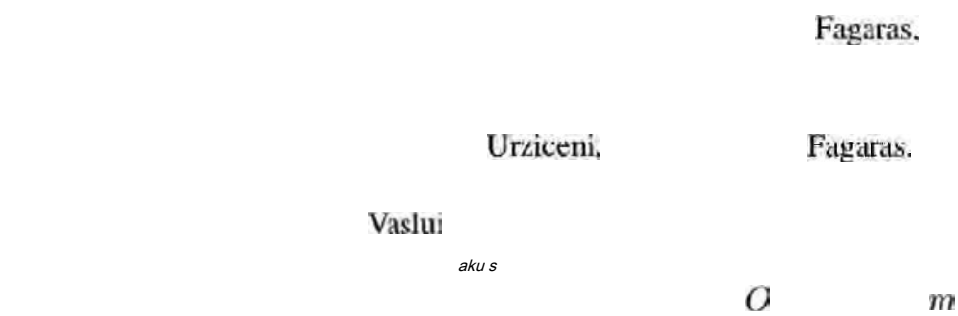
Zerind

best-first

(1984).

				Mehadia
				Neamt
	Drobeta			Rimnicu Vilcea
	Fagaras			
	Hirsova	151		Urziceni
Lngo	j	244	Zerind	374
<i>h_{SLN}</i> —straight-line				

saya t



3.5.2 A * search: Meminimalkan total perkiraan biaya solusi



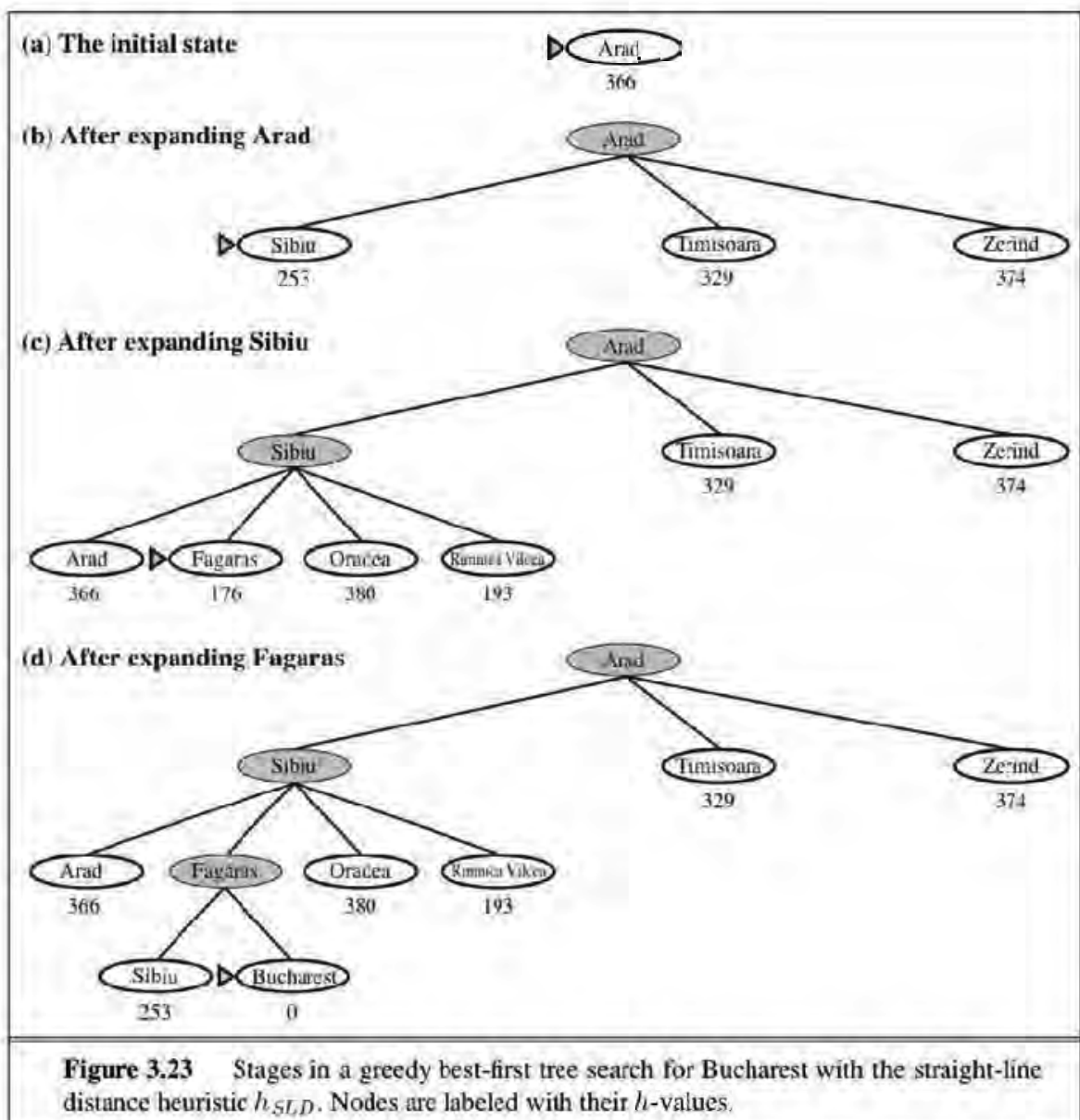
$$g(n) + h(n)$$

$$g(n)$$

$$n$$

$$n$$

$$g(n) + h(n)$$



ADMISSIBLE
HEURISTICS

$$f(n) =$$

$$h_{SLD}$$

h_{SLD}

mungkin

CONSISTENCY
MONOTONICITY

search. $14N)$
 $a.$

n'

n' :

$$h(n) \leq c(n, a, h(n'))$$

SEGI TIGA
INEQUALITY

n' .

G_n

G_n n'

$h(n)$,

$h(n)$

$G.$

h_{SLD} .

saya t



A^*

itu tree-search versi A^*

optimal if dapat diterima, sedangkan versi grafik-pencari optimal jika $h(n)$ konsisten.

g

A^*

if $h(n)$ konsisten, maka nilai-nilai

$f(n)$ any nondecreasing

$$g(n') = g(n) + c(n, a, n')$$

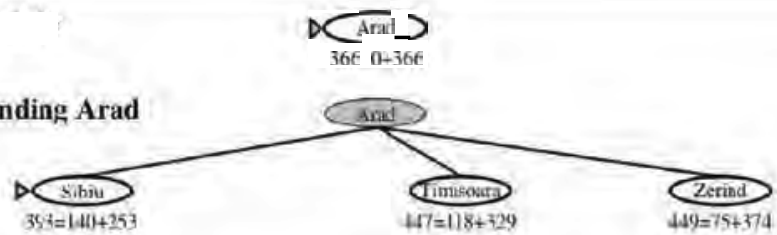
$$f(n') = g(n') \quad h(n') = c(n, a, n') + h(n) > h(n) = f(n)$$

setiap kali A^* memilih node n untuk ekspansi, jalur optimal

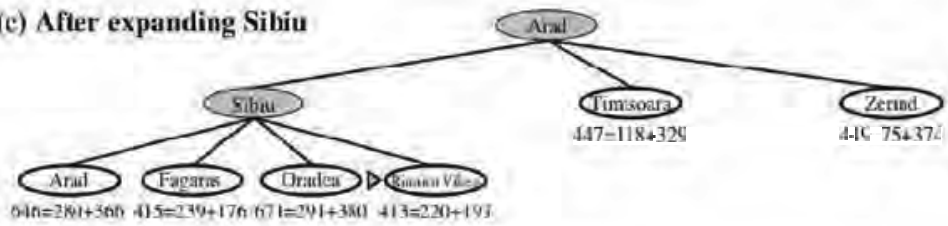
ke node yang telah ditemukan.

n'

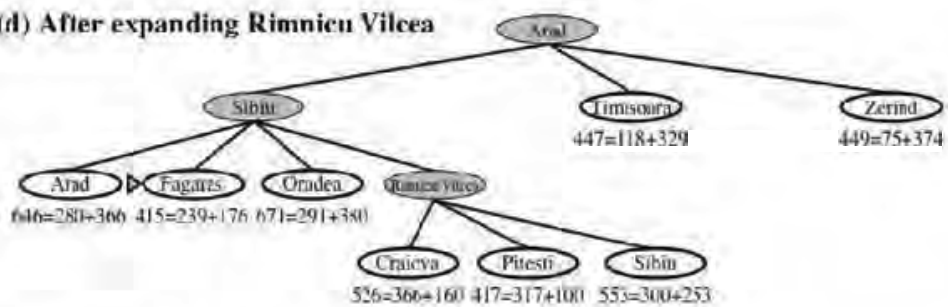
(b) After expanding Arad



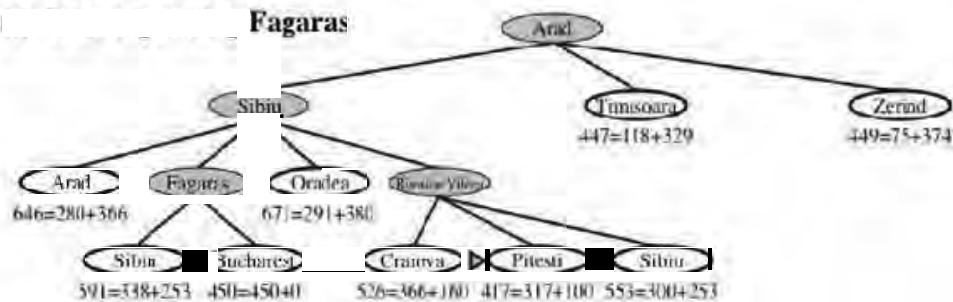
(c) After expanding Sibiu



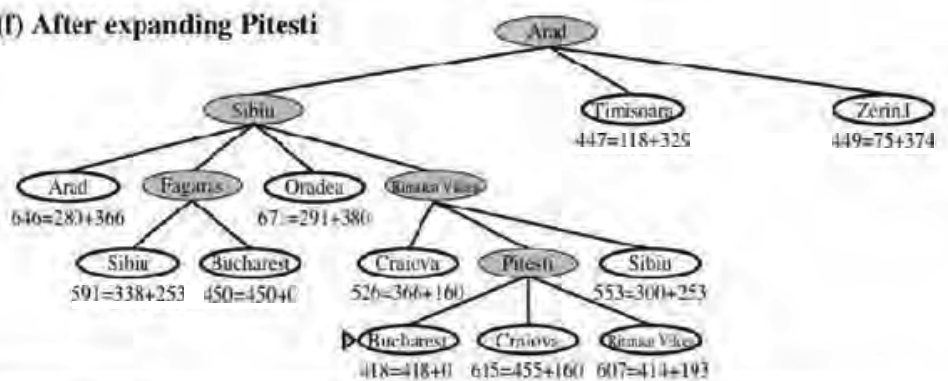
(d) After expanding Rimnicu Vilcea



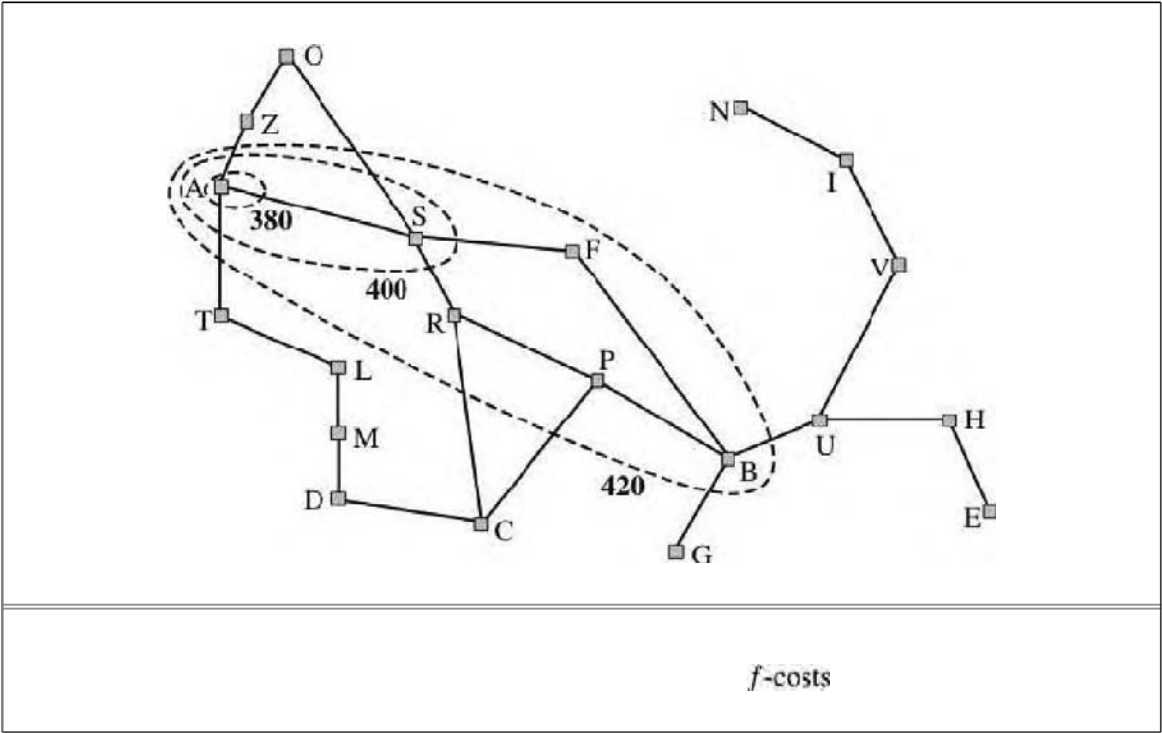
Fagaras



(f) After expanding Pitesti



$$f = h$$



nondecreasing n^l

GRAPH-SEARCH nondecreasing

f

KONTUR

example.

A^*

C^* .

A^*

C^*

C^* .

$f(n)$ C^* —for ϵ

Timisoara

PEMANGKASAN

$$h_{SLD}$$

pruning—eliminating

UPI: IMALIN
EFFIC: THT

information—A*

$$(\tau_i)$$

not

ABSOLUTE ERROR
ERROR RELATIF

$$h^* \frac{(h^* - h)}{h^*}$$

Sebuah

$$O(b^{\frac{n}{2}}).$$

A*

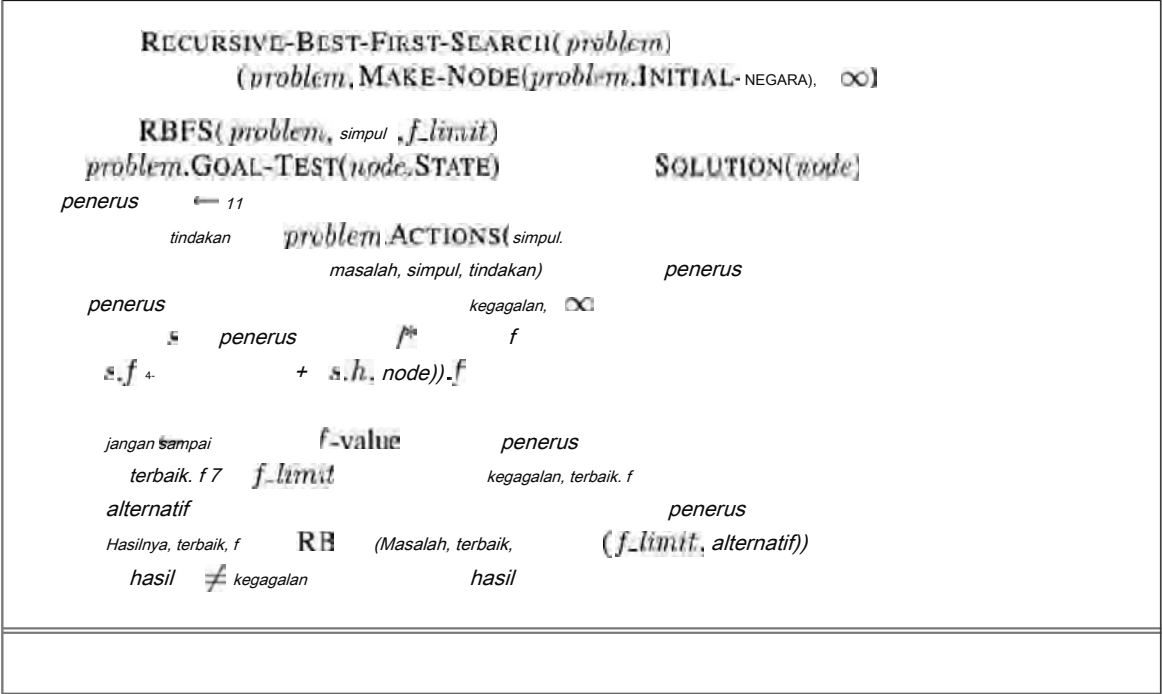
is,

$$\frac{(b^{\epsilon d})}{b^{\epsilon}} = ((b^{\epsilon})^{\frac{d}{\epsilon}})$$

$$2^{\frac{n}{2}}$$

C*—even

A*'s



A*

ITERATIVE-
DEEPENING

(IDA*)

IDA*

f (*g* +

IDA*

RECURSIVE
BEST-FIRST PENCARIAN

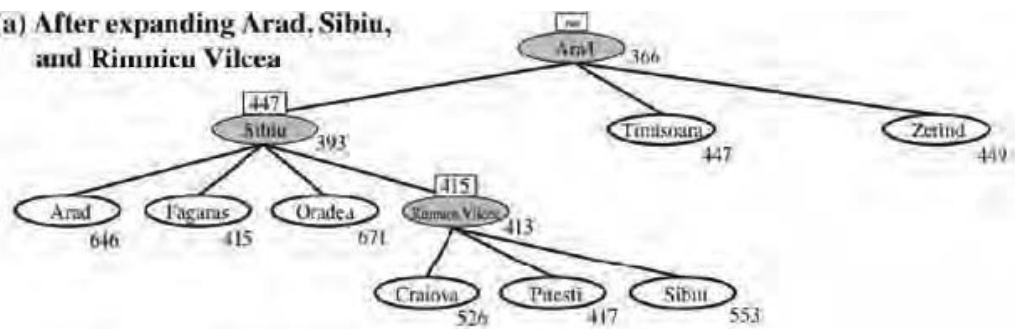
f_limit

alternatif

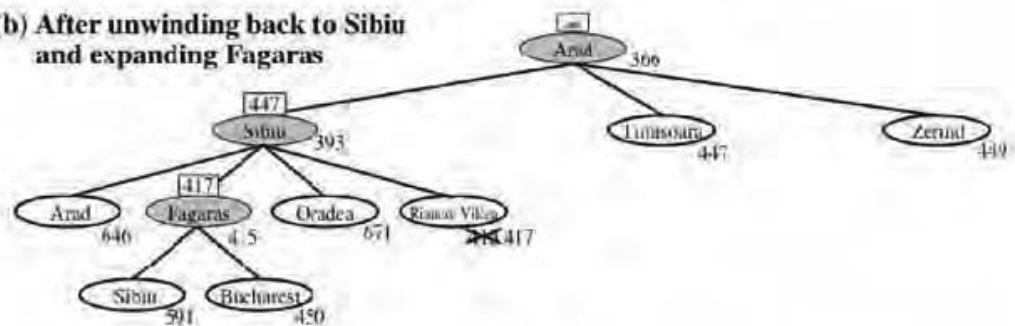
BACKUP NILAI

children.

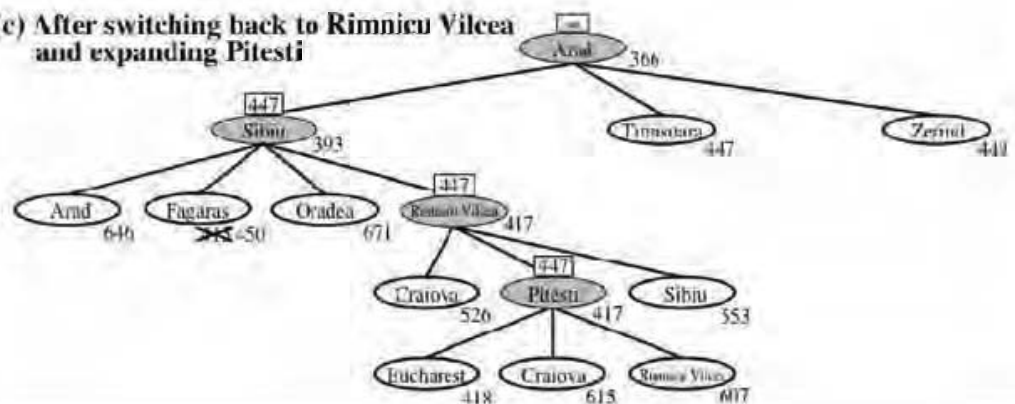
(a) After expanding Arad, Sibiu, and Rimnicu Vilcea



(b) After unwinding back to Sibiu and expanding Fagaras



(c) After switching back to Rimnicu Vilcea and expanding Pitesti



RBFS

f -cost.

Rimnicu

(Fagaras).

subtree

Rimnicu Vilcea;

Fagaras;

Rimnicu

RBFS

IDA*

RBFS

Rimnicu

increase— h

IDA*

SEBAGAI

$h(n)$

terlalu sedikit

SMA* A* SMA* MA* SMA*

terburuk

SMA*
SMA*

SMA*

n_1

n_2

SMA*

terbaru

sulung

bahkan jika itu adalah pada

jalur solusi optimal,
be

SMA*

d ,

SMA*

the

SMA*

baru

of computation waktu.

keterbatasan memori dapat membuat masalah terselesaikan

from inti nya

ETALEVEL NEGARA
SPACE

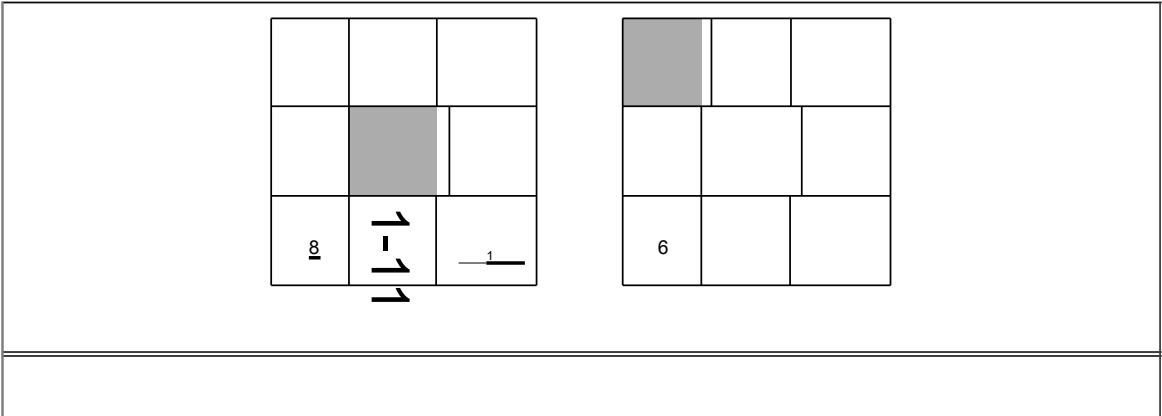
OBJECTLEVEL

belajar

metalevel

metalevel

ETALEVEL



15-puzzle 10¹³,

h_1 an

MANHATTAN
DISTANCE 2

$h_2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18.$

EFFECTIVE b^* .
faktor pencabangan

A^* $(b^*)^2 \cdot d$

b^* A^*

fairly

$h_2.$ [illegible] h_2 $h_1.$
$$n, h_2(n) > (n)$$

DOMINATION

A²⁴
$$(N) = \frac{C^*}{h(n)} \cdot \frac{C^*}{C^*}$$

h2

10

8-puzzle

 k_z $h_z?$ lv_2

RELAXED

h_2

supergraph

lebih baik
solusi optimal untuk masalah santai adalah heuristik diterima untuk masalah asli.

biaya dari

to

1

tanpa

pencarian,

obtain.

problem

(Prieditis, 1993).

$h_1 \dots h_m$

$$h(n) = \max\{h_1(n), \dots, h_m(n)\} .$$

h .

* 2		4
*		
*	3	1

		2
	4	

subproblem

saya t

saya t

from

subproblem

3.28.

POLA DATABASE

instance—in

h_{DB}
back

15-puzzles

menambahkan,

5-6-7-8

By

PATTERN
DATABASES

database pola menguraikan. Dengan database seperti itu,

15-puzzles

2008),

nonadditive

$$h(n)$$

$$h(n)$$

$$h(n)$$

18.

fitur

$$x_1(n) \quad x_2(n)$$

$$h(n).$$

$$x_1(n) \quad x_2(n)$$

$$h(n)?$$

$$c_1 x_1(n) + c_2 x_2(n) .$$

$$c_1 \quad c_2$$

pasang

$$h(n) =$$

Breadth-first

Uniform-cost

$g(n)$,

$h(n)$

best-first

$h(n)$.

$$g(n) + h(n).$$

GAMBAR-SEARCH).

RBFS

SMA*

A*

inition,

precomputed

defi-

of

Bellman

(1968).

8-puzzle

15-puzzle

mid-1870s.

American Journal of Mathematics

$n \rightarrow n$

problems.

Schach

undecidable.

et al.,

Shahookar

Mazumder

Bellman

Martelli's

cases.

search"

"penetrance"

$g(n)$.

A*.

et al.

Huyn
2007). “effective

$$O((b^*)^{\frac{1}{2}}), \quad O(b^{\frac{1}{2}}),$$

search, Helmert

$$A^* \\ f_w(n) = w_g g(n)$$

$$g(n) +$$

$$h(n)$$

$$w_g$$

$$w_h$$

~~is~~-admissible—that

$$\epsilon$$

$$\epsilon$$

$$A_\epsilon^*$$

$$\epsilon$$

$$A^*$$

$$A^*$$

et dari, 2006).

$$IDA^*$$

1966).

1988).

(1978)

1992).

RBFS

[WU

A* DTA*

Vefald,

MA*

et al.

IE

SMA*

minimum-spanning-tree

Priedi-

tis
Hemadvolgyi

Edelkamp

Iansson

(1984) *Heuristics*

et al.

Artificial Intelligence majalah *of* yang ACM_

PARALLEL_ PENCARIAN

Sebuah

et al.,

(Ralphs *et al.,*

Korf

labirin

b.

c.

d.

details.

$$d(i, j)$$

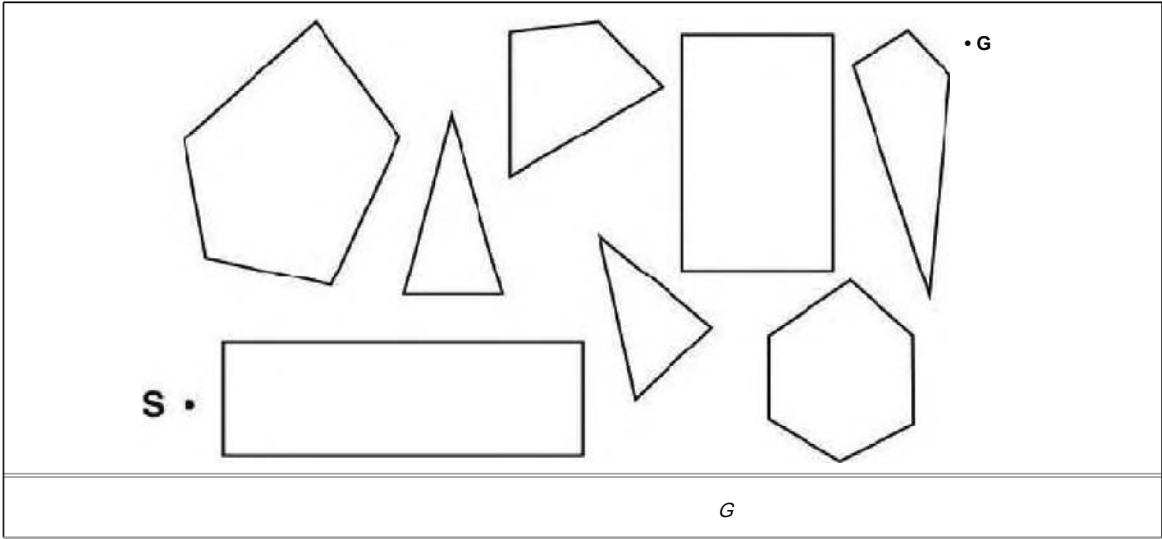
$$D(i, j) = D[i, j]/2.$$

e.

(1982).)

$$\sqrt[3]{n!}$$

are



records.
records.

—
—

(x, y)

polygons.

e?

■

adalah

(Amarel

Go(Sibiu)

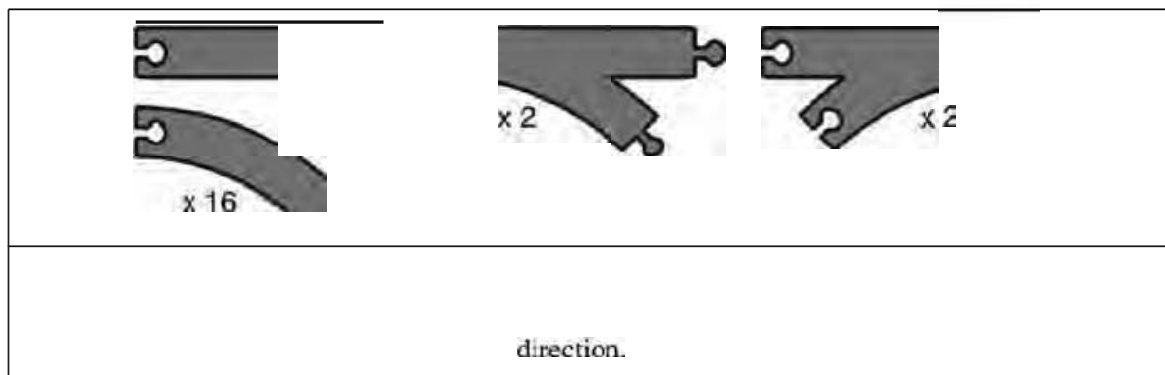
etc_

Pergi actions.

Go(Sibiu)Go(Rimnicu Vilcea)Go(Ploesti)Go(Bucharest).

(Petunjuk:

afterwards.)



b. $h(n)$ if
 A^*

allowed.

11.

problem + all?

3.16

3.32

$[\epsilon, 1], \quad \leq \leq$

3.18

$O(n^2) \quad O(n))$.



URLs



3.20

b.

Construct

formance

search

n
 $n?$

per-

Sebuah

the

MST—see Exercise
RBFS

8-puzzle

A*

algoritma akan mempertimbangkan dan the dan skor li setiap node.

$h(n)$

Hal RISTIC 51.114
ALGORITHM

$, F(n) = w)g(n)$

untuk =

$(x, y).$

sebuah

(i.e., n

$(i, \dots i + 1,$

$n.$

$n.$

$y_i);$

h_i

+

grid.

$\sum_{i=1}^n z_i$
 $\max\{h_1,$

,

h

Sebuah.

e

d.

3,31

heuristis

Gaschnig's

Gaschnig ini

1

Gaschnig's

3,32