Searching

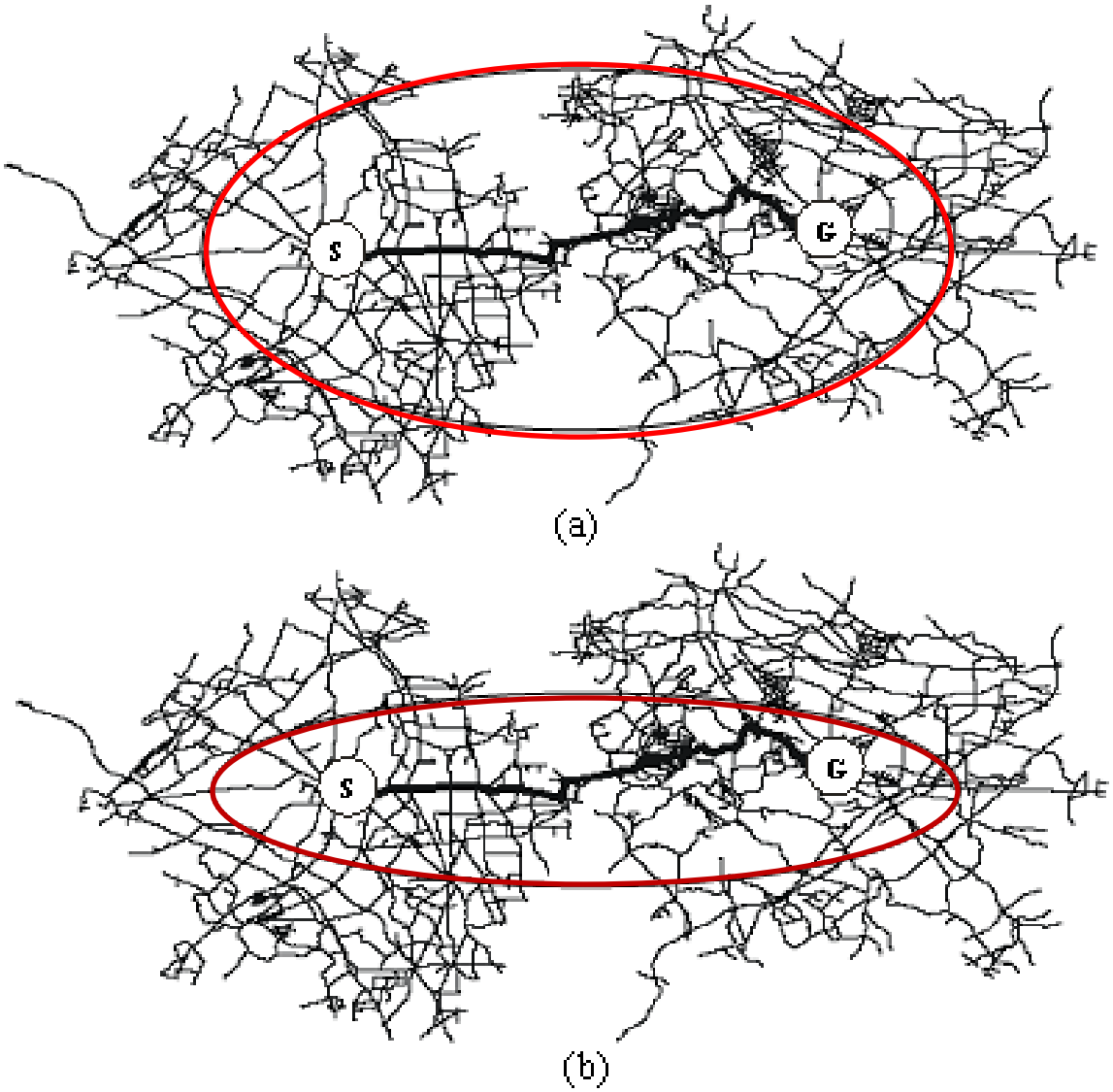
Mesin Learning

By. Gunawansyah

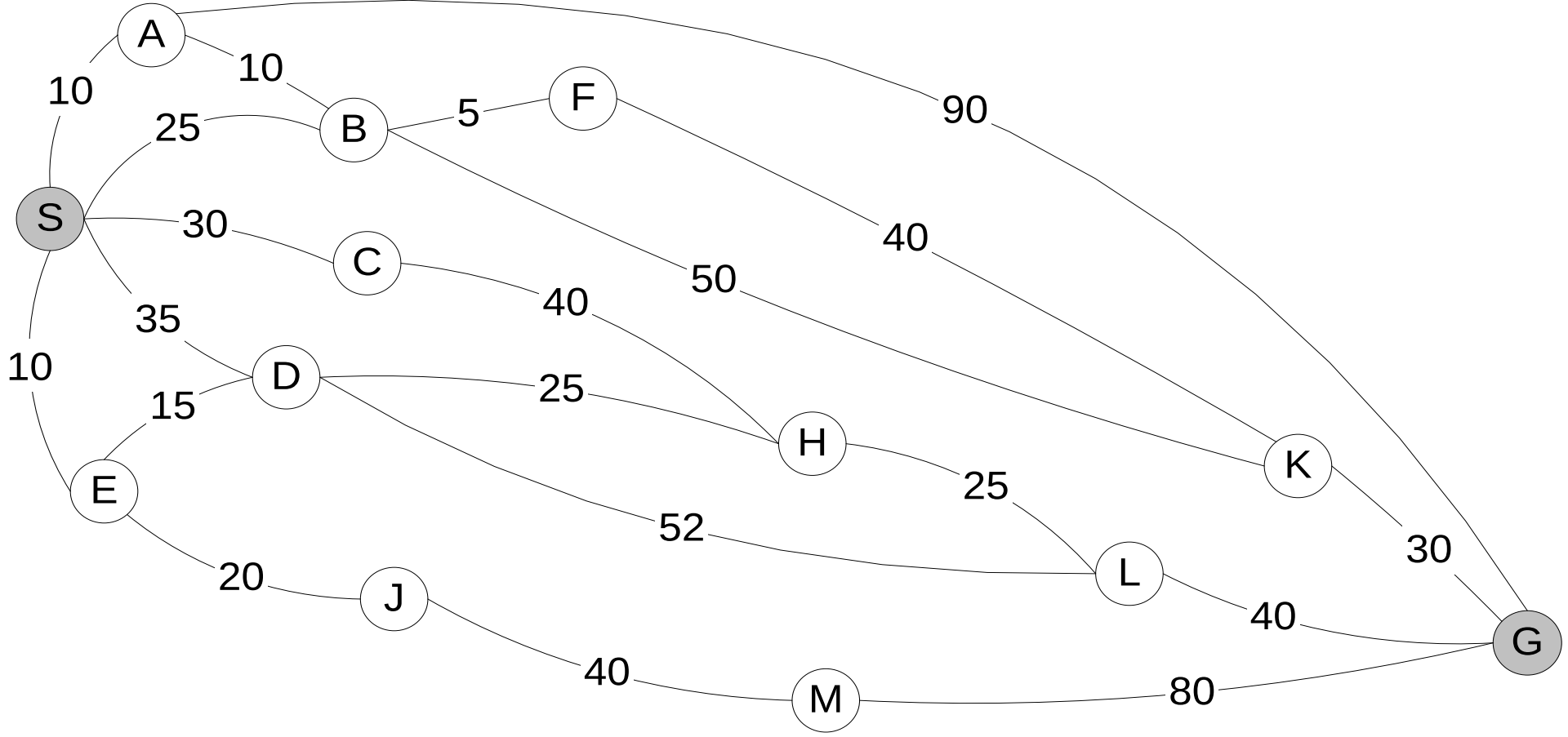
*Bi-directional A** (BDA*)

*Bi-directional A** (BDA*)

- Algoritma A* dari dua arah: simpul asal dan tujuan.
- Pencarian dihentikan jika *BestNode* dari simpul asal telah berada di dalam *CLOSED* dari simpul tujuan. Cek apakah harus mengganti *parent* dari *BestNode* tersebut dari arah simpul tujuan.
- Atau sebaliknya, pencarian dihentikan jika *BestNode* dari simpul tujuan telah berada di dalam *CLOSED* dari simpul asal. Cek apakah harus mengganti *parent* dari *BestNode* tersebut dari arah simpul asal.



Area pencarian yang dilakukan oleh BDA* (b) lebih sempit dibandingkan dengan area pencarian A* (a).



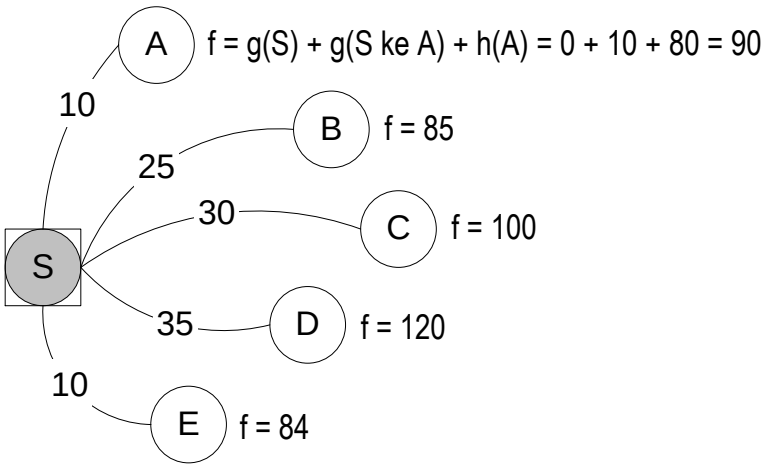
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_s(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

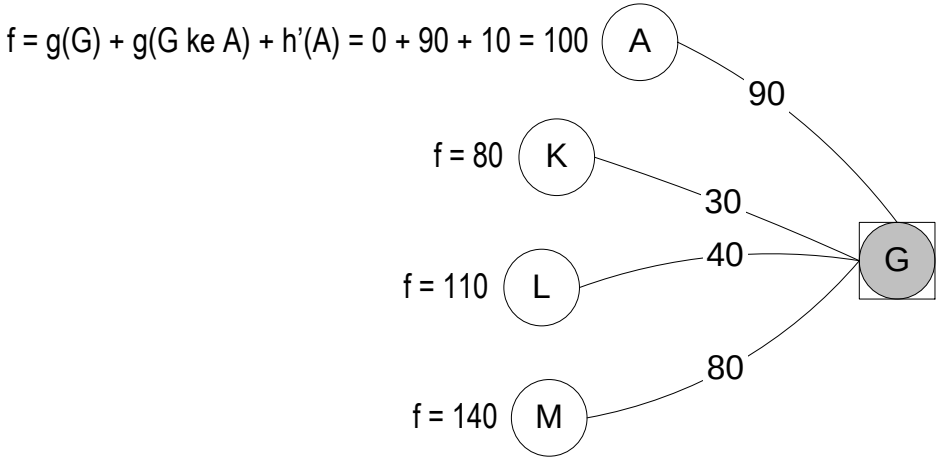
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_g(n)$	0	10	15	25	30	5	20	90	45	25	50	70	60

Pencarian Maju (dari S ke G)

Pencarian Mundur (dari G ke S)

Langkah 1

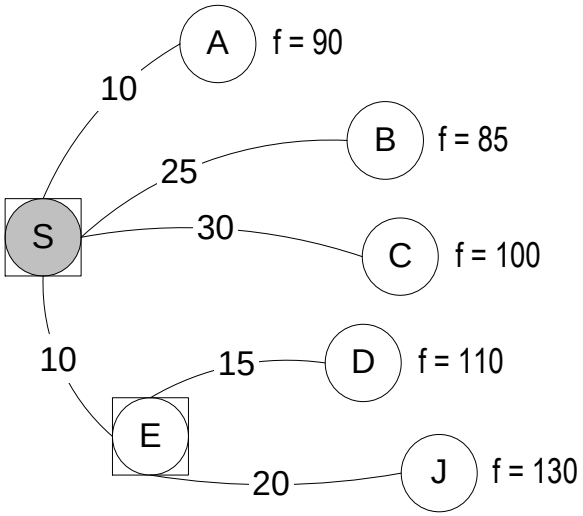


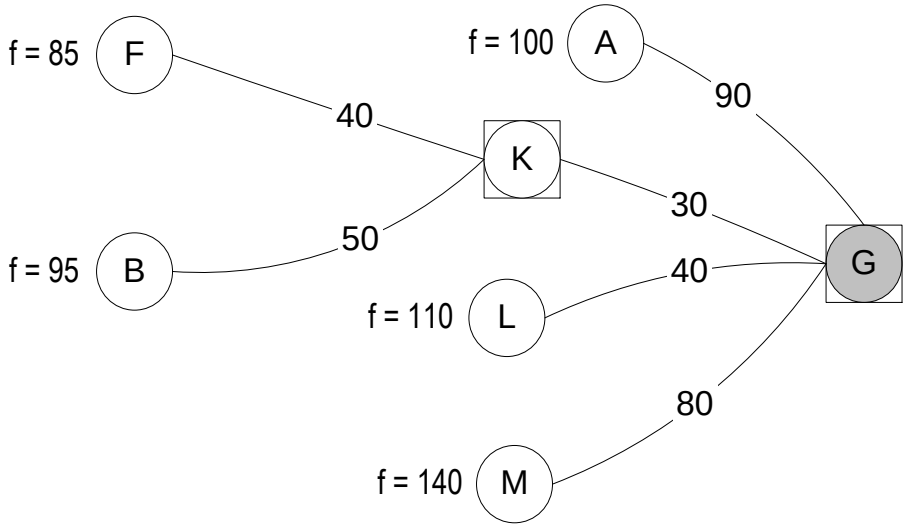


n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_s(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_g(n)$	0	10	15	25	30	5	20	90	45	25	50	70	60

Langkah 2

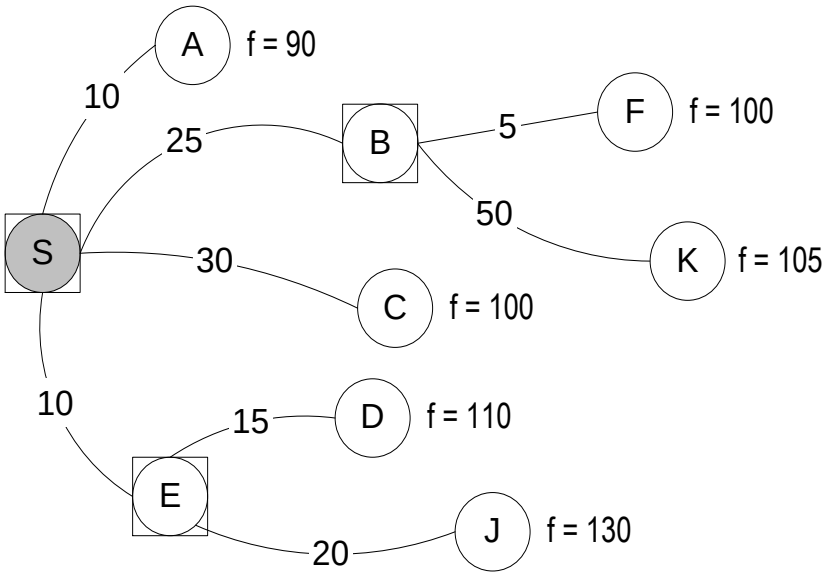


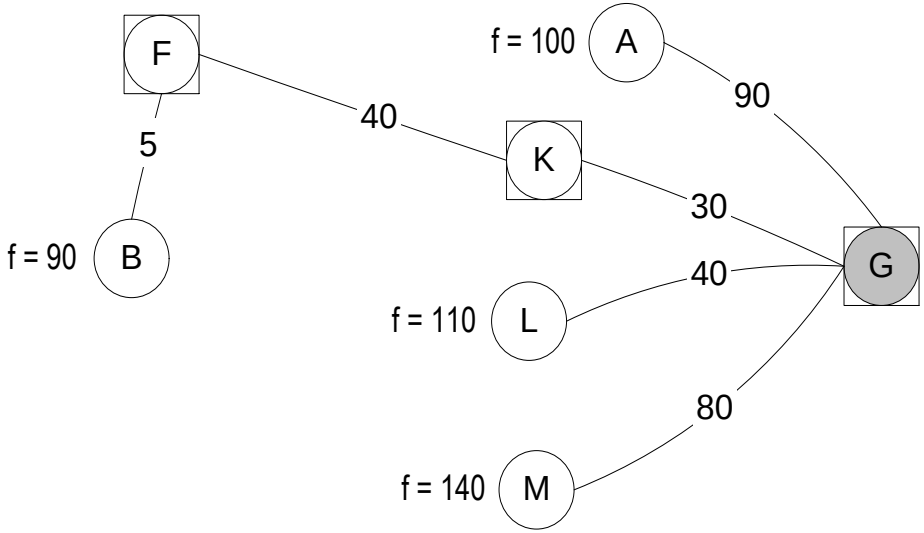


n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_s(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_g(n)$	0	10	15	25	30	5	20	90	45	25	50	70	60

Langkah 3





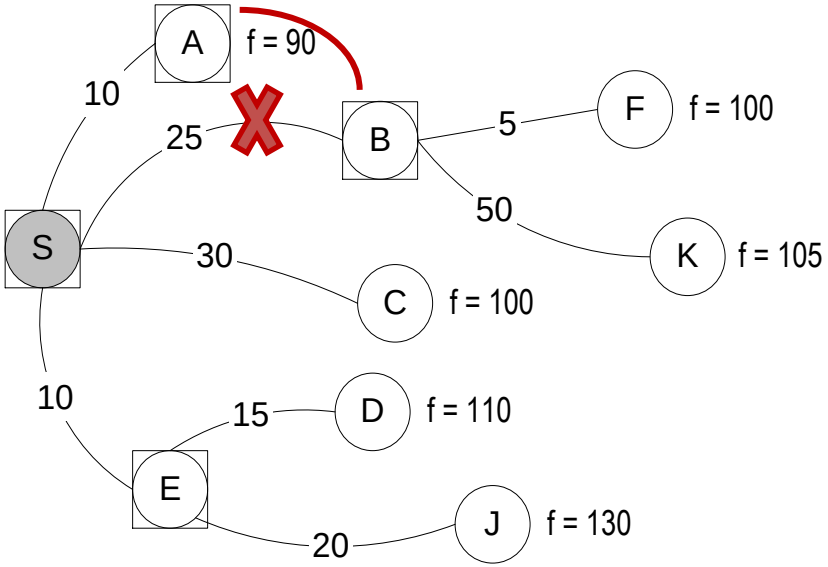
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_s(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

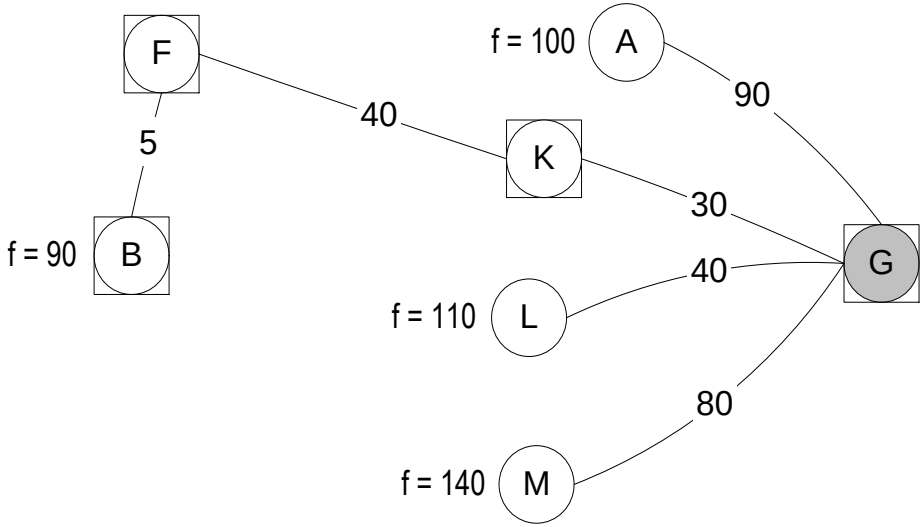
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_g(n)$	0	10	15	25	30	5	20	90	45	25	50	70	60

n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_s(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_g(n)$	0	10	15	25	30	5	20	90	45	25	50	70	60

Langkah 4





Rute: S-A-B-F-K-G

Total Biaya = 95

Biaya optimal = 95

BDA*

- Cari suksesor dari B yang sudah ada di dalam *CLOSEDs*. A adalah suksesor dari B dan berada di dalam *CLOSEDs*.
- *g*(S,B) melalui A lebih kecil daripada *g*(S,B) langsung, maka *parent* dari B diubah (dari S manjadi A)
- Nilai *g* dan *f* pada B juga diubah.
- Hasil penelusuran balik menghasilkan **S-A-B-F-K-G** dengan total jarak = **95**.
- BDA* adalah ***complete*** dan **optimal**.

*Modified Bi-directional A** (MBDA*)

*Modified Bi-directional A** (MBDA*)

- Fungsi heuristik untuk simpul *n* pada pencarian maju (dari S ke G):

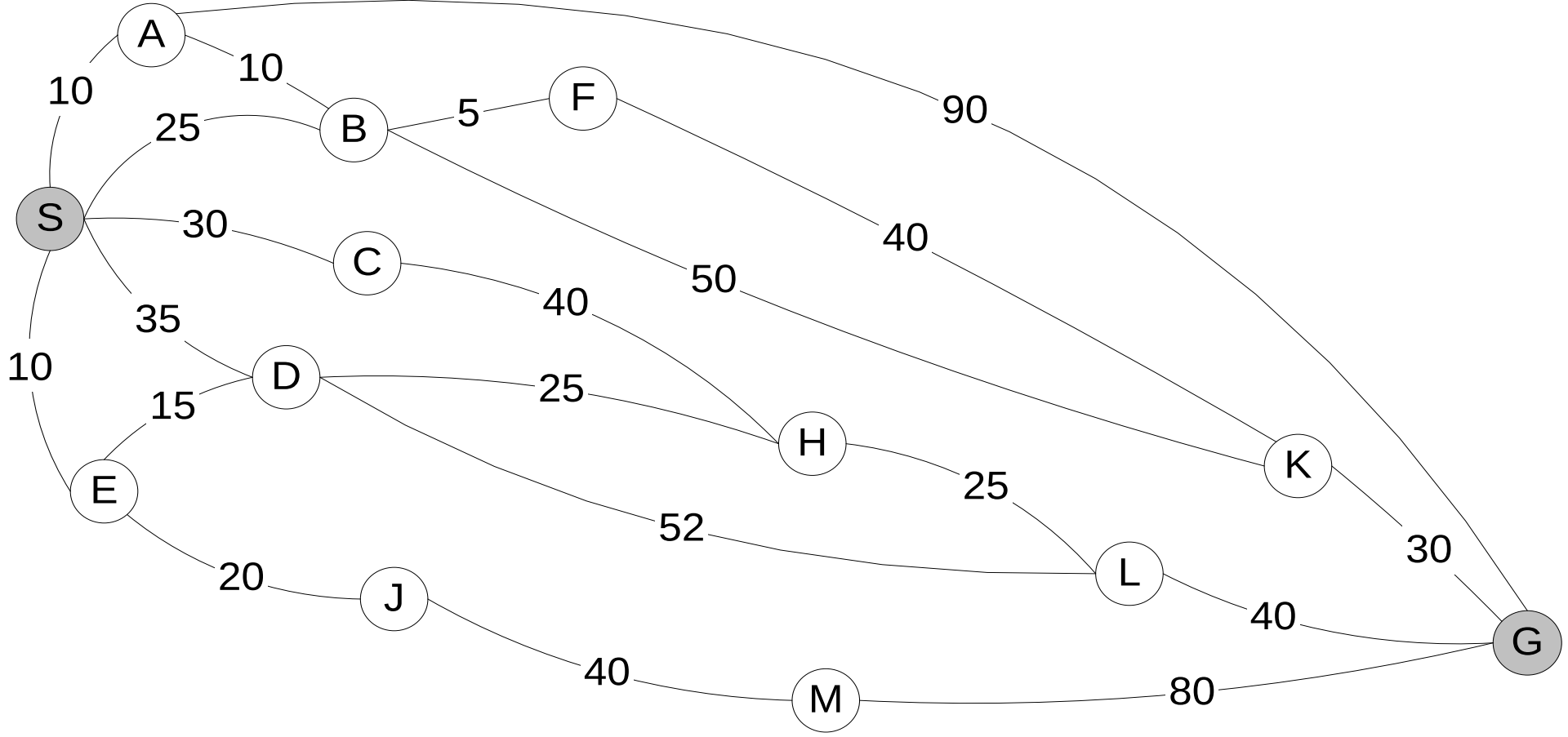
$$f = g(S,n) + \frac{1}{2}\left[h_s(n) - h_g(n)\right]$$

- Fungsi heuristik untuk simpul *n* pada pencarian mundur (dari G ke S):

$$f = g(G,n) + \frac{1}{2}\left[h_g(n) - h_s(n)\right]$$

*Modified Bi-directional A** (MBDA*)

- S : simpul asal atau *initial state*
- G : simpul tujuan atau *goal state*
- $g(S,n)$: biaya sebenarnya dari *S* ke *n*
- $g(G,n)$: biaya sebenarnya dari *G* ke *n*
- $h_s(n)$: biaya perkiraan dari *n* ke *G*
- $h_g(n)$: biaya perkiraan dari *n* ke *S*



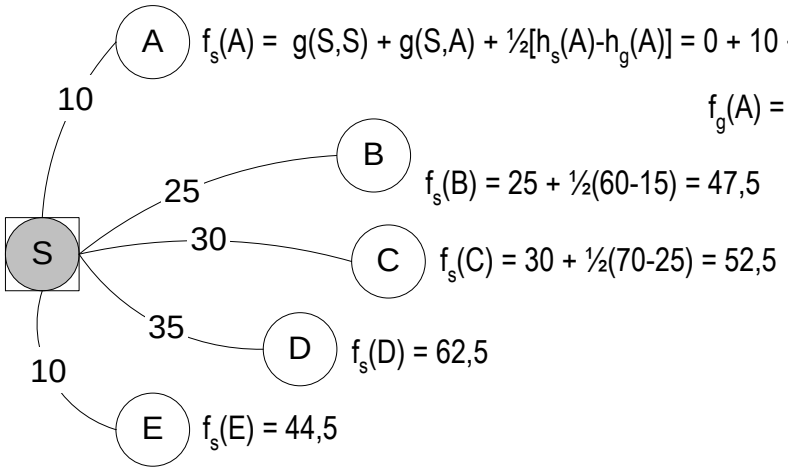
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_s(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

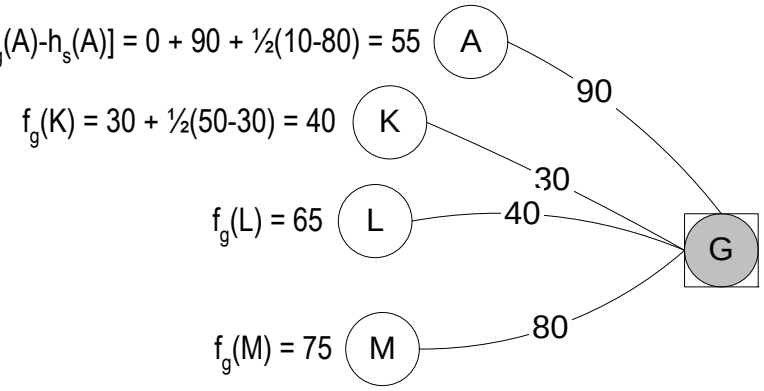
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_g(n)$	0	10	15	25	30	5	20	90	45	25	50	70	60

Pencarian Maju (dari S ke G)

Pencarian Mundur (dari G ke S)

Langkah 1

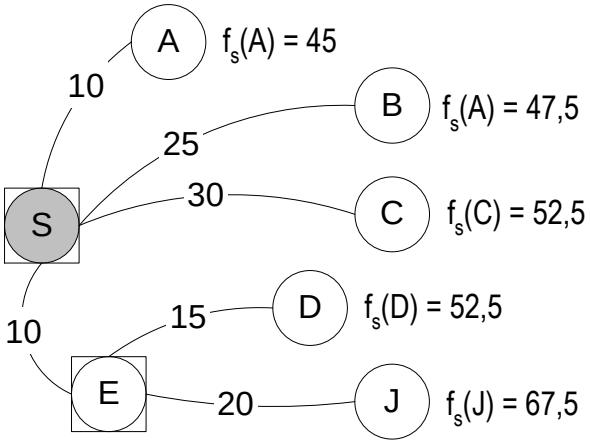


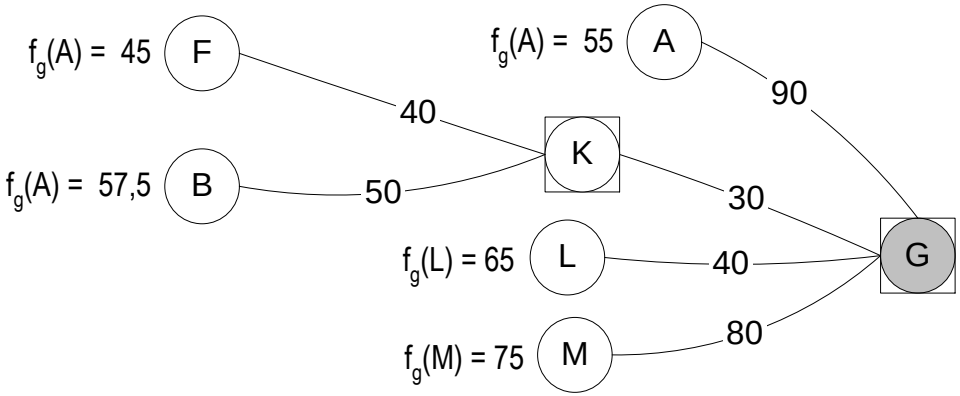


n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_s(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

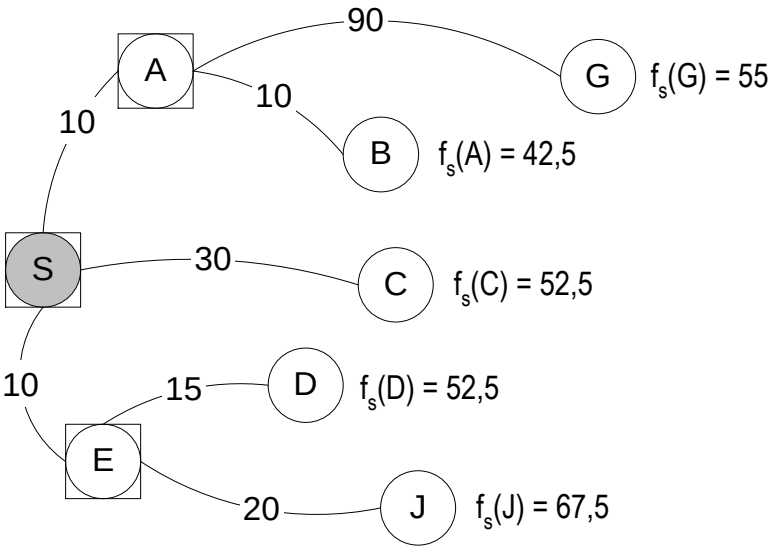
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h_g(n)$	0	10	15	25	30	5	20	90	45	25	50	70	60

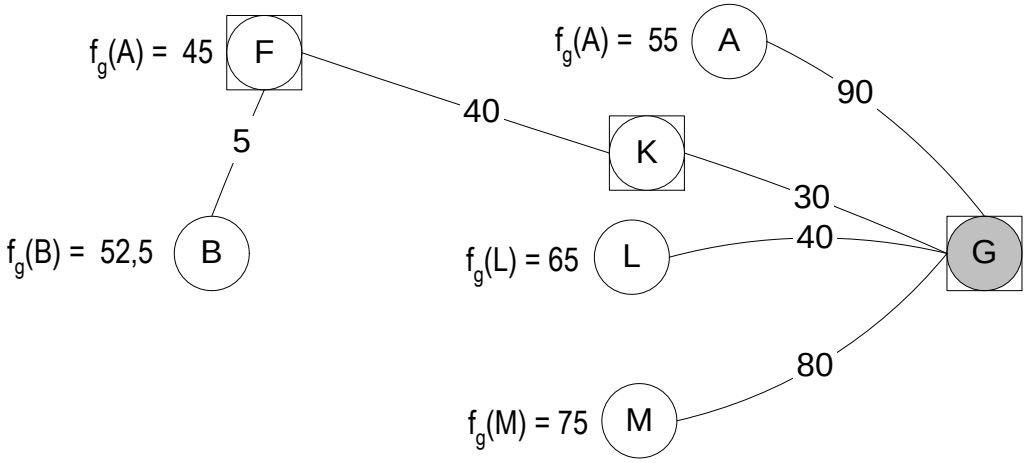
Langkah 2


A
$f_s(A) = 45$
10
B
$f_s(A) = 47,5$
25
S
30
C
$f_s(C) = 52,5$
10
15
D
$f_s(D) = 52,5$
E
20
J
$f_s(J) = 67,5$

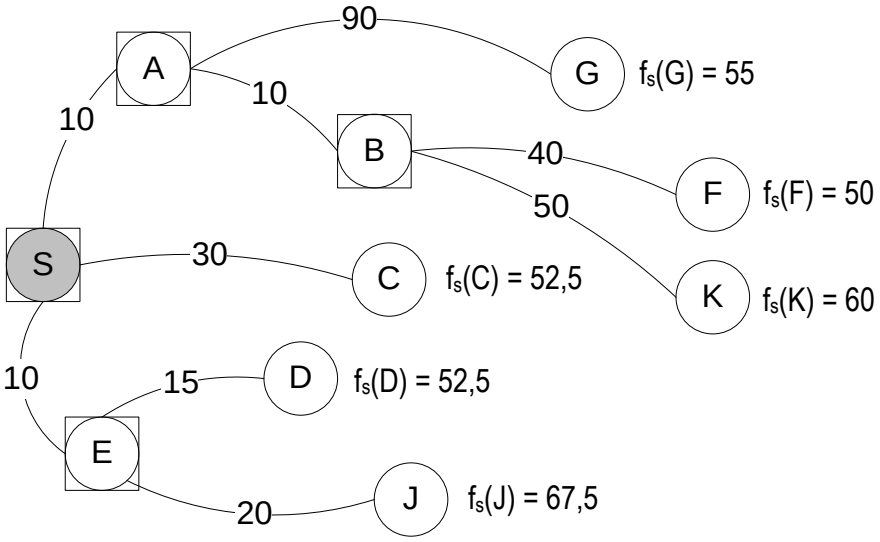

$f_g(A) = 45$
F
$f_g(A) = 55$
A
90
40
K
$f_g(A) = 57,5$
B
50
30
G
$f_g(L) = 65$
L
40
$f_g(M) = 75$
M
80

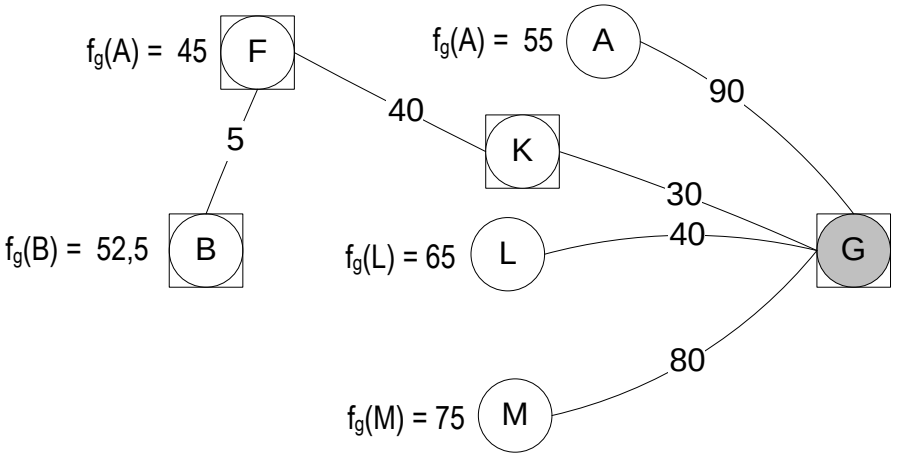
Langkah 3


A
G
B
S
C
D
E
J
90
10
10
10
30
15
20
$f_s(G) = 55$
$f_s(A) = 42,5$
$f_s(C) = 52,5$
$f_s(D) = 52,5$
$f_s(J) = 67,5$


$f_g(A) = 45$
F
5
$f_g(B) = 52,5$
B
40
$f_g(A) = 55$
A
90
K
30
$f_g(L) = 65$
L
40
G
80
$f_g(M) = 75$
M

Langkah 4





Rute: S-A-B-F-K-G

Total Biaya = 95

Biaya optimal = 95

MBDA*

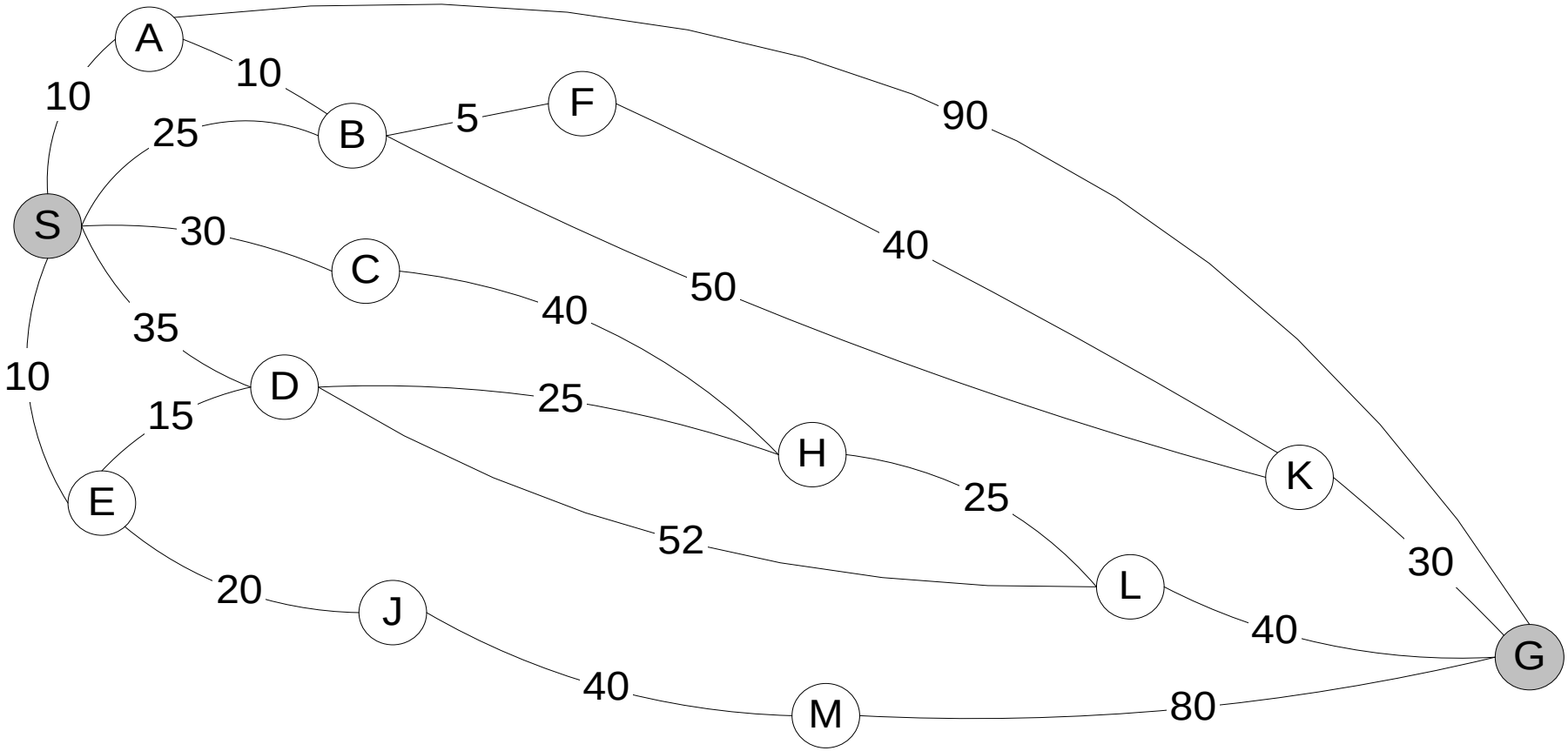
- Eksperimen yang dilakukan Tetsuo Shibuya terhadap jaringan jalan sesungguhnya (*actual road network*) di Tokyo membuktikan bahwa jumlah simpul yang dibangkitkan oleh MBDA* adalah setengah dari jumlah simpul yang dibangkitkan oleh A* [TET97].

*Dynamic Weighting A** (DWA*)

*Dynamic Weighting A** (DWA*)

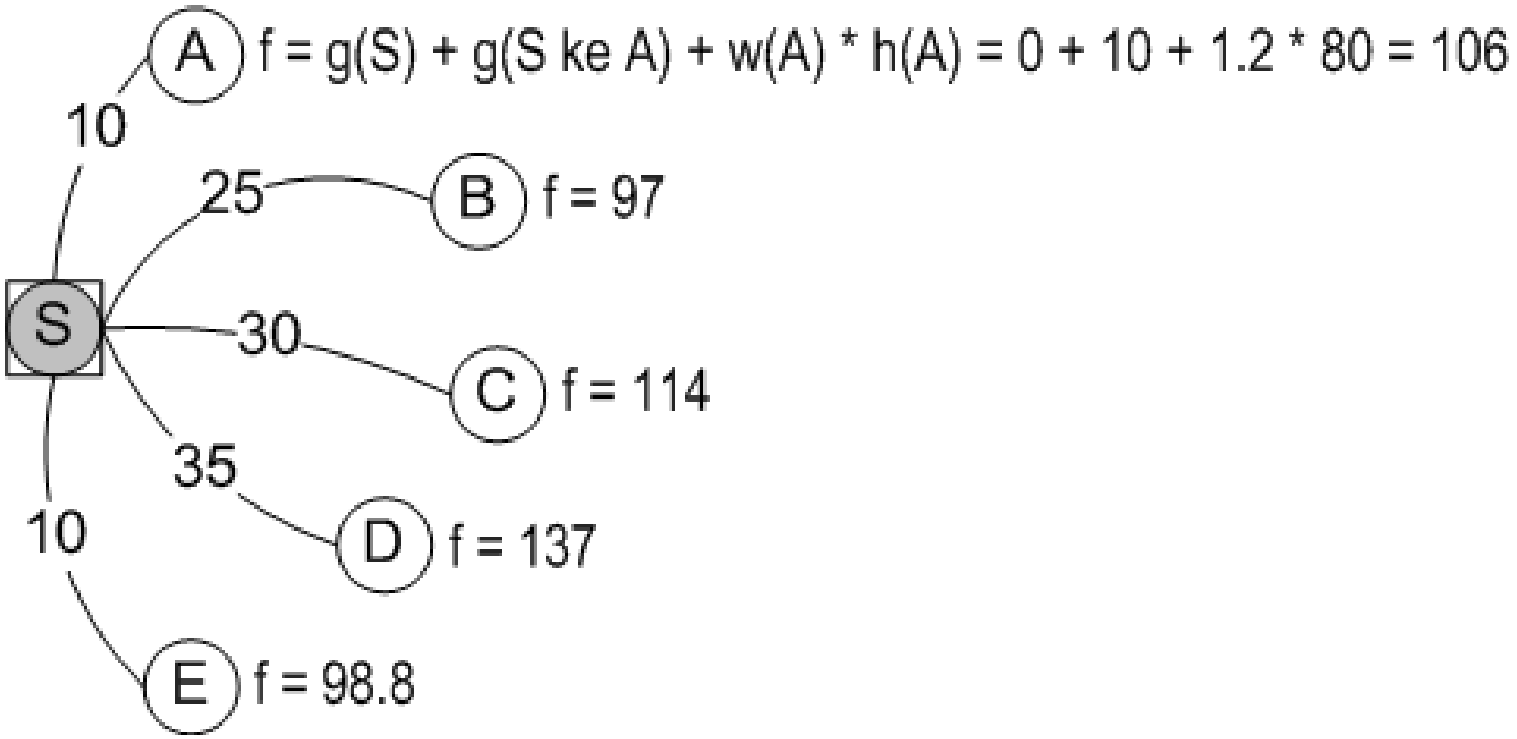
- Fungsi heuristik *h diberi* bobot dinamis.
- Pada awal iterasi, lebih baik pencarian dilakukan ke arah mana saja.
- Tetapi, ketika *goal* sudah dekat, barulah pencarian difokuskan ke arah *goal*.
- Fungsi heuristik yang digunakan

$$f(n) = g(n) + w(n) * h(n)$$

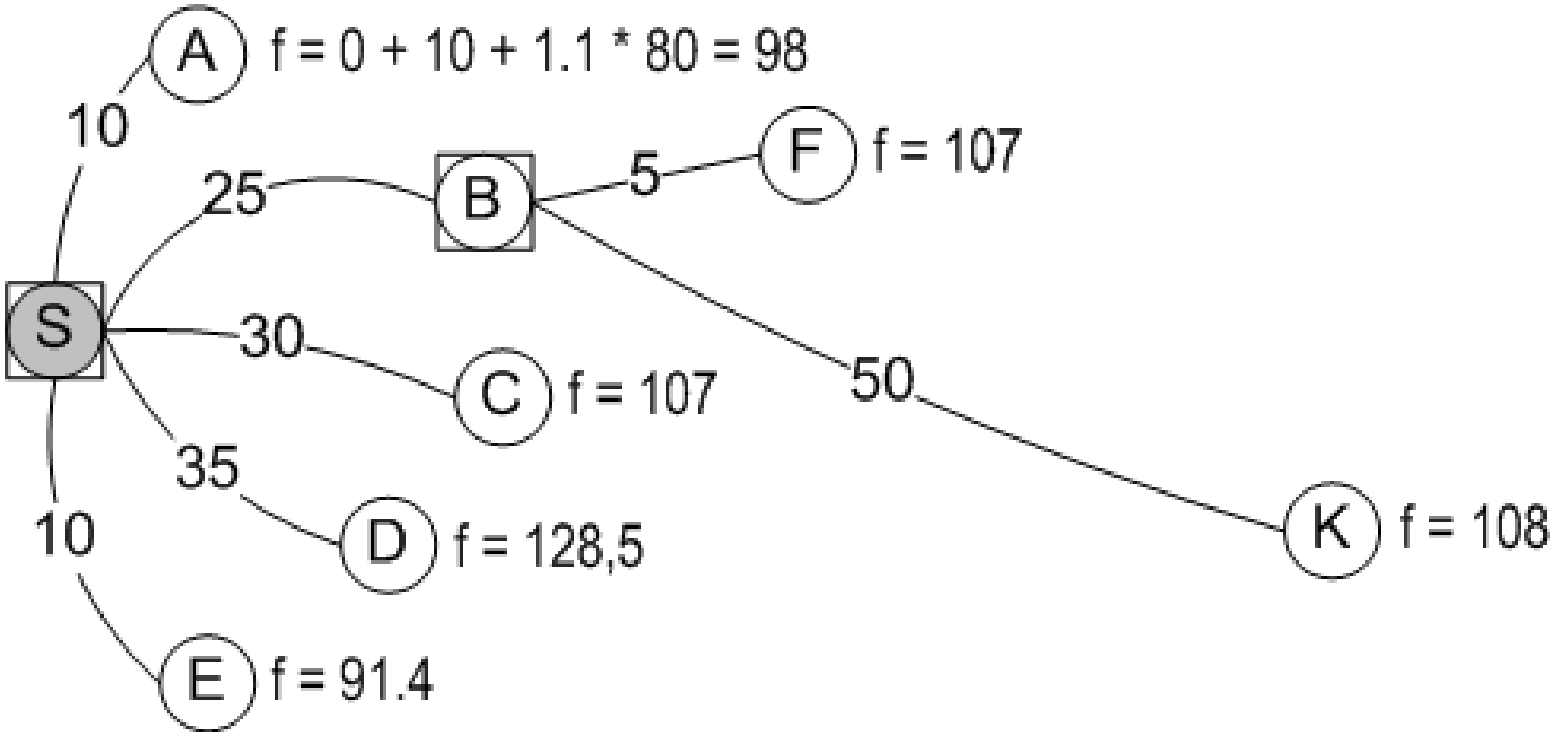


n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

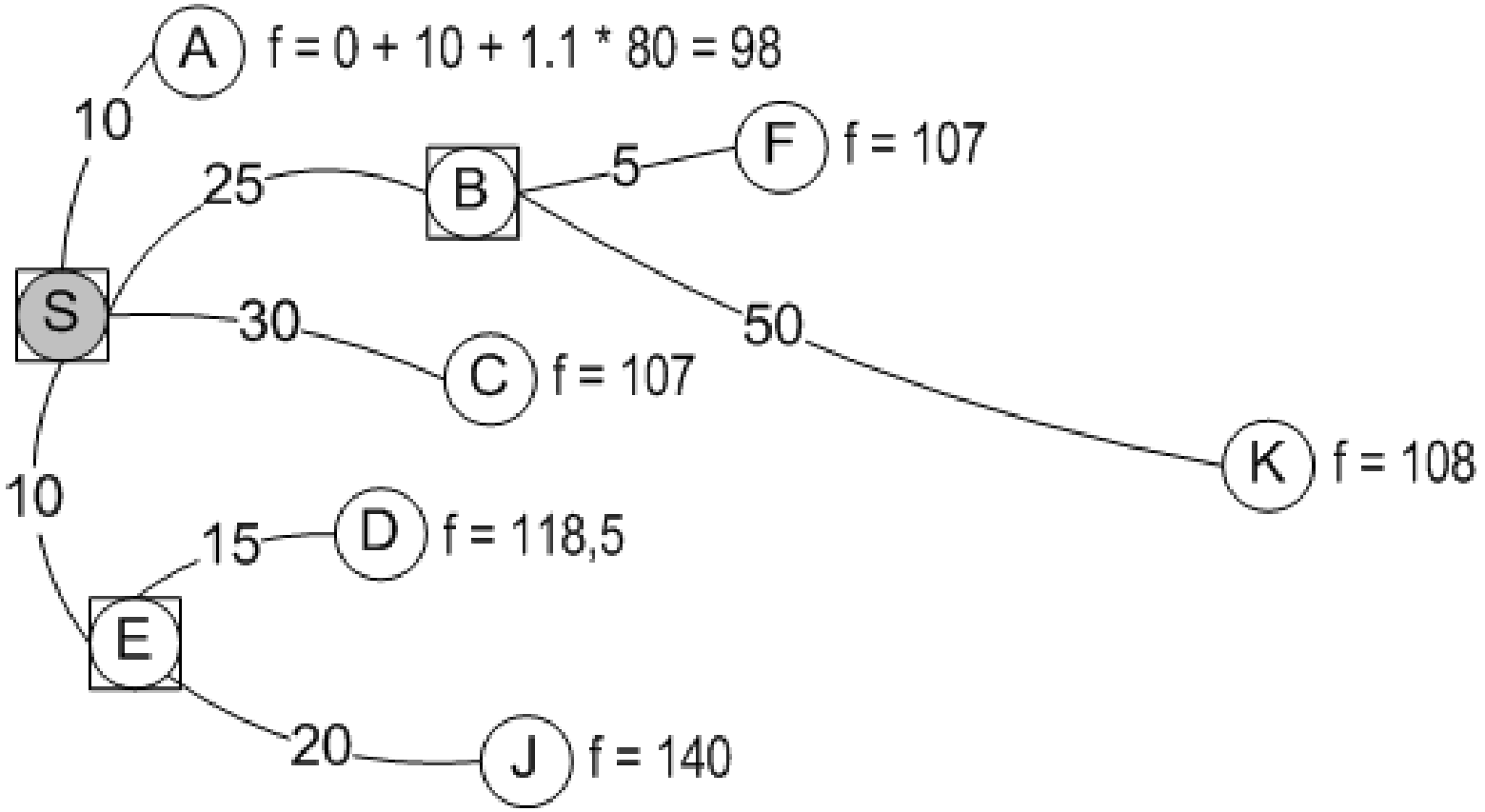
Langkah 1 (w = 1.2)



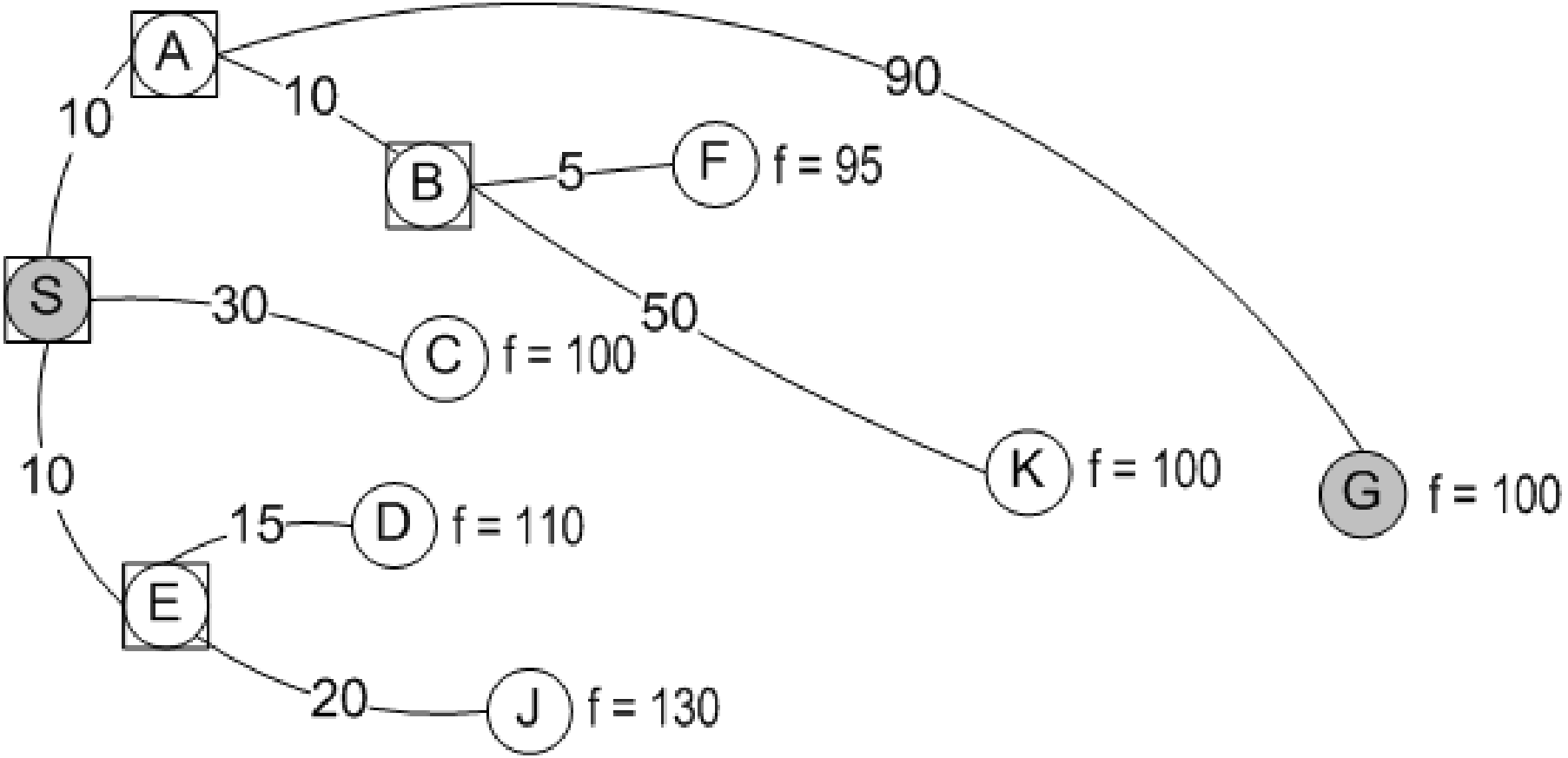
Langkah 2 (w = 1.1)



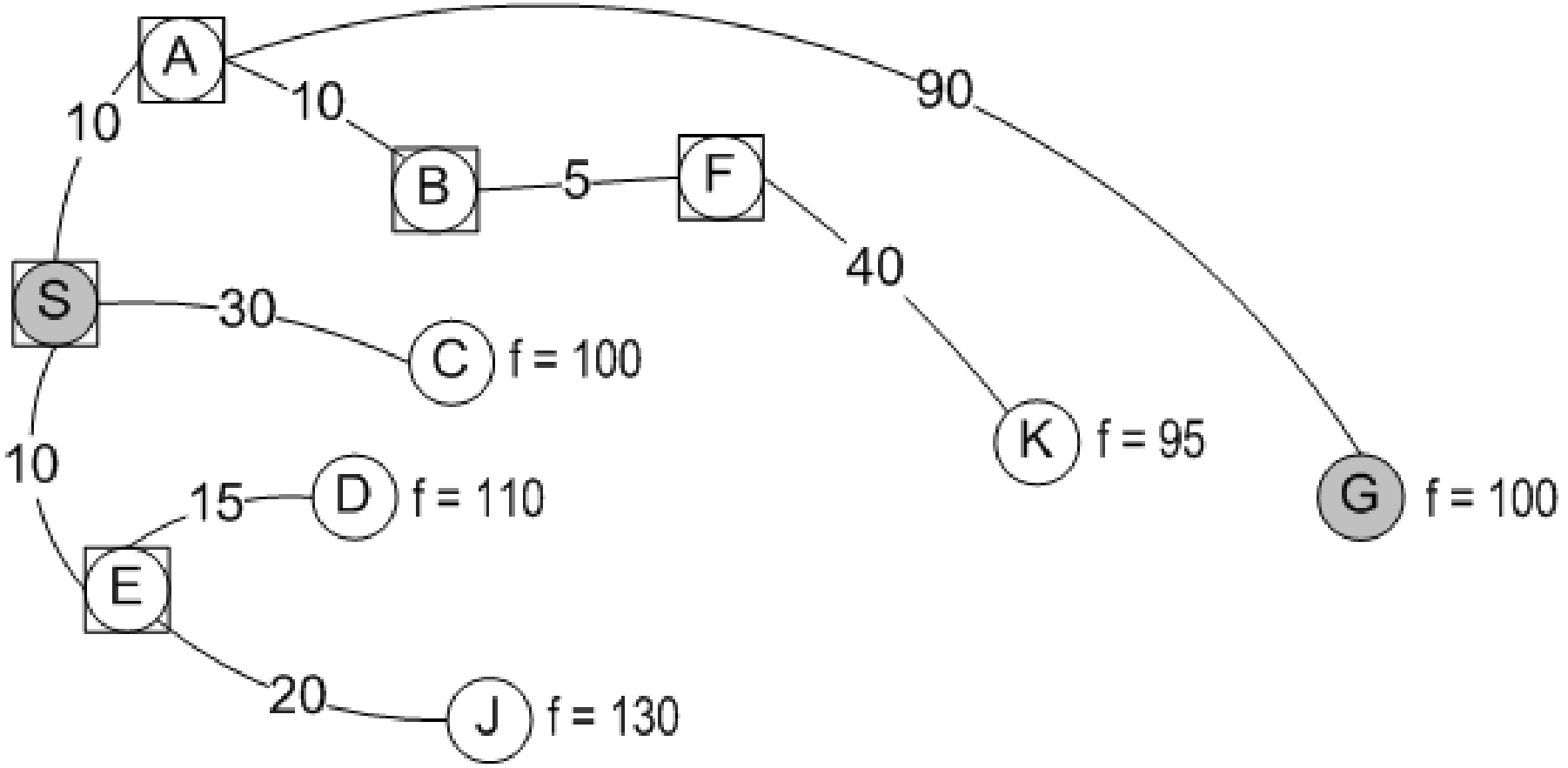
Langkah 3 (w = 1.1)



Langkah 4 (w = 1.0)



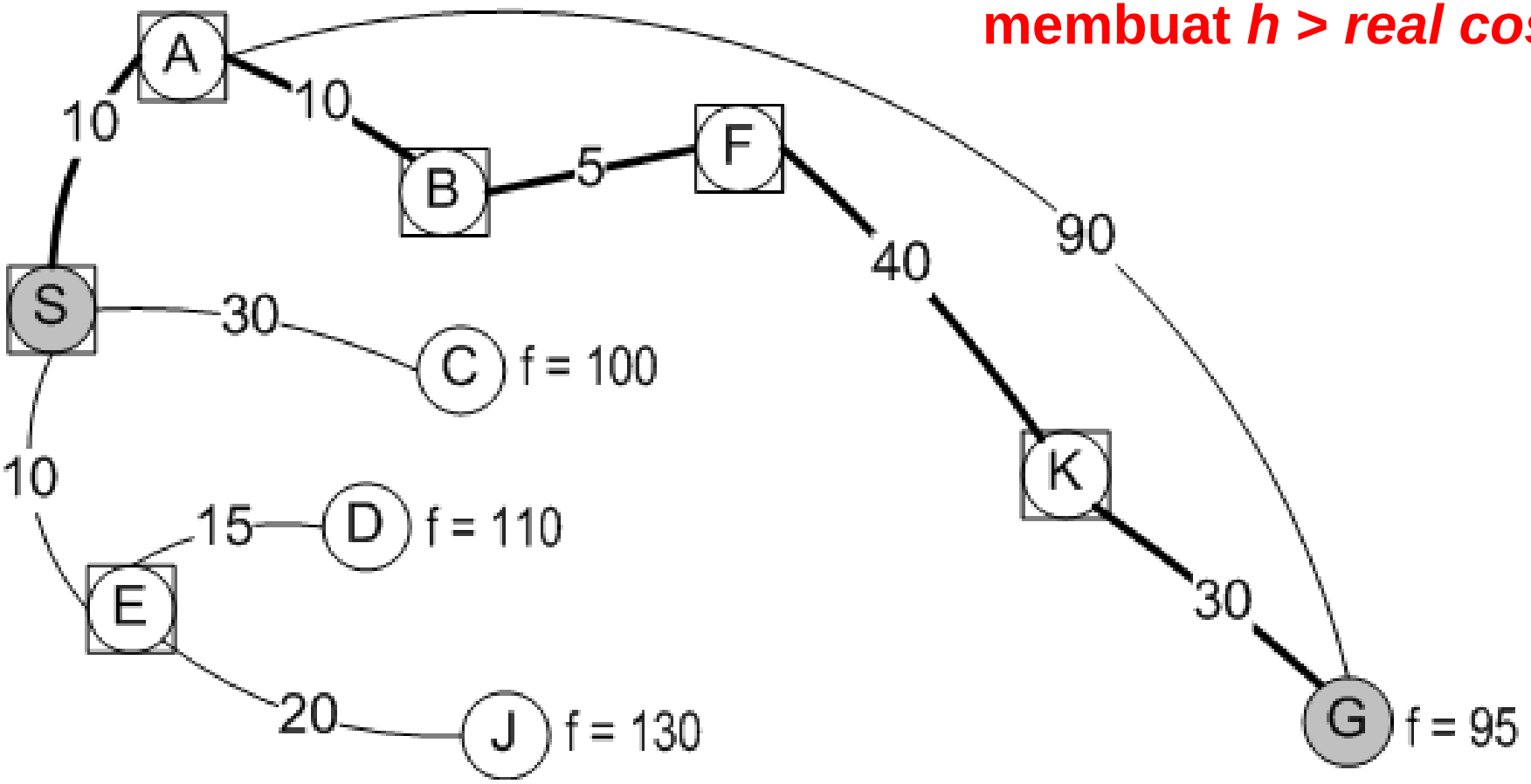
Langkah 5 (w = 1.0)



Total Biaya = 95

Biaya optimal = 95

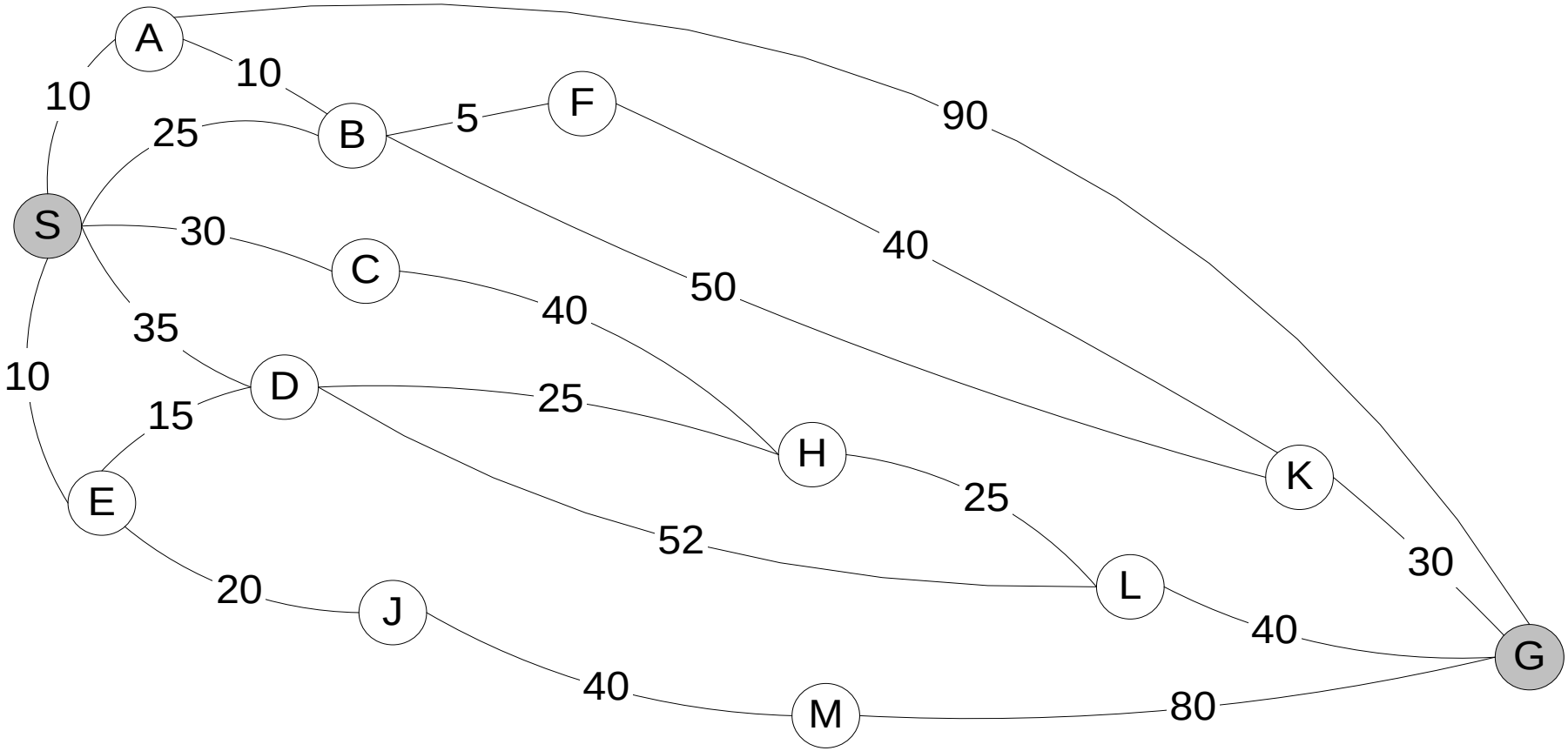
DWA optimal untuk *w* yang tidak membuat *h* > *real cost*



*Beam A** (BA*)

*Beam A** (BA*)

- Membatasi jumlah simpul yang bisa disimpan di dalam *OPEN*.
- Ketika jumlah simpul di *OPEN* sudah melebihi batas tertentu, maka simpul dengan nilai *f* terbesar akan dihapus.
- Sedangkan jumlah simpul di *CLOSED* tidak dibatasi karena tidak boleh dihapus (untuk *backtrack*).
- Dengan membatasi jumlah simpul di *OPEN*, maka pencarian menjadi lebih terfokus seperti sinar (*beam*).

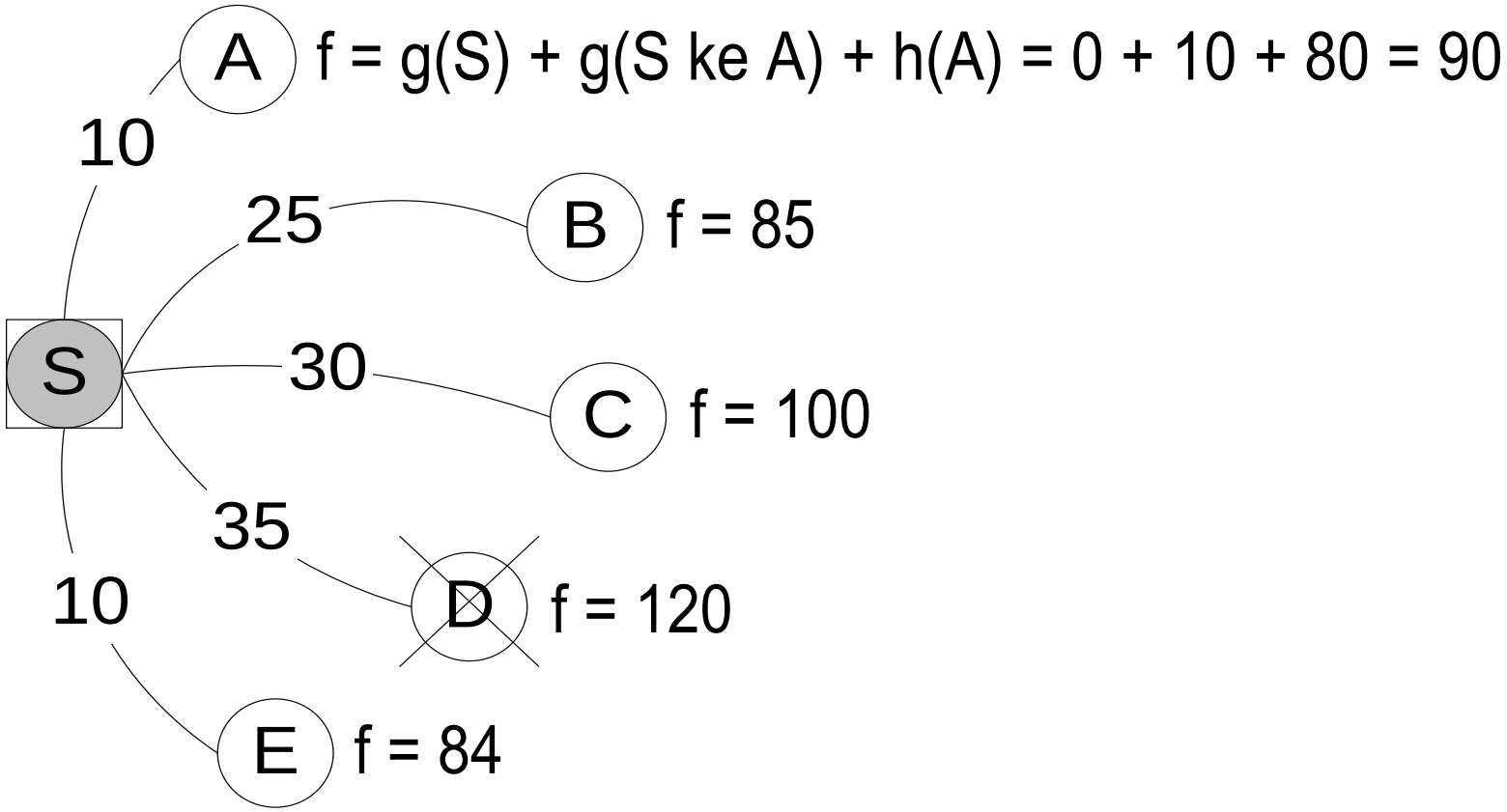


n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

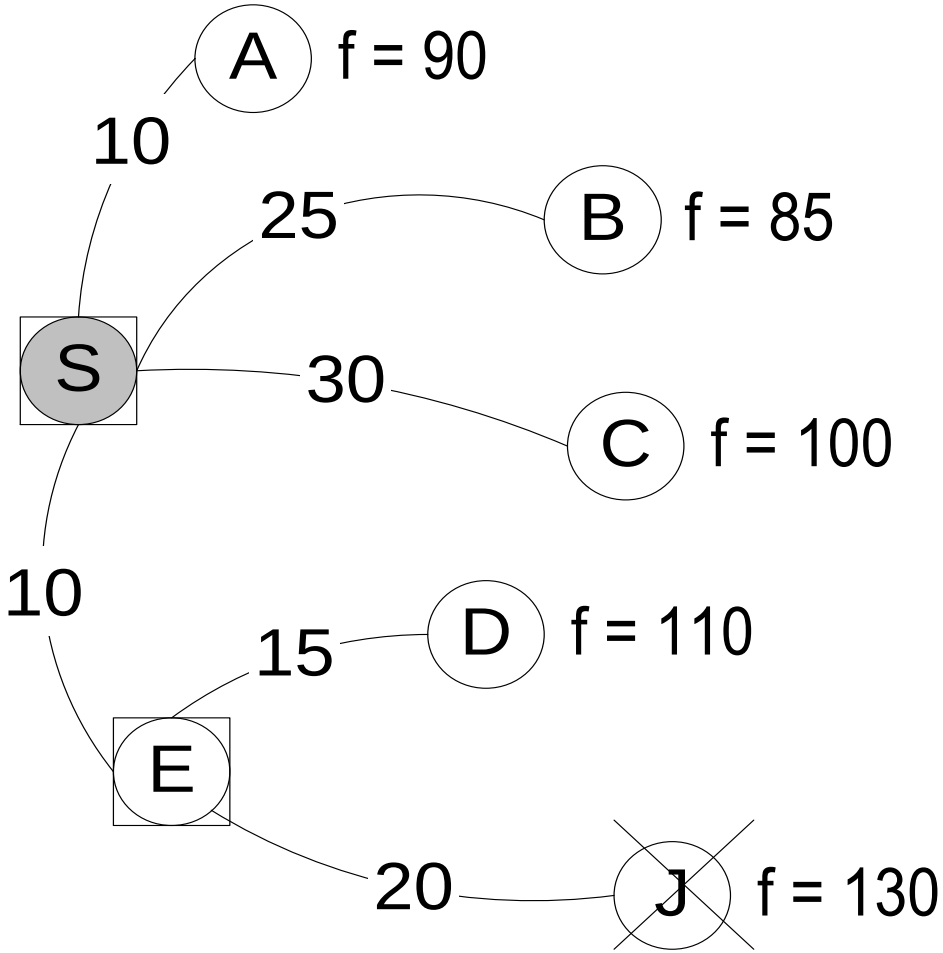
*Beam A** (BA*)

- Pada kasus ini, misalkan jumlah simpul maksimum yang bisa disimpan di dalam *OPEN* adalah 4.
- Bagaimana BA* menemukan solusi?

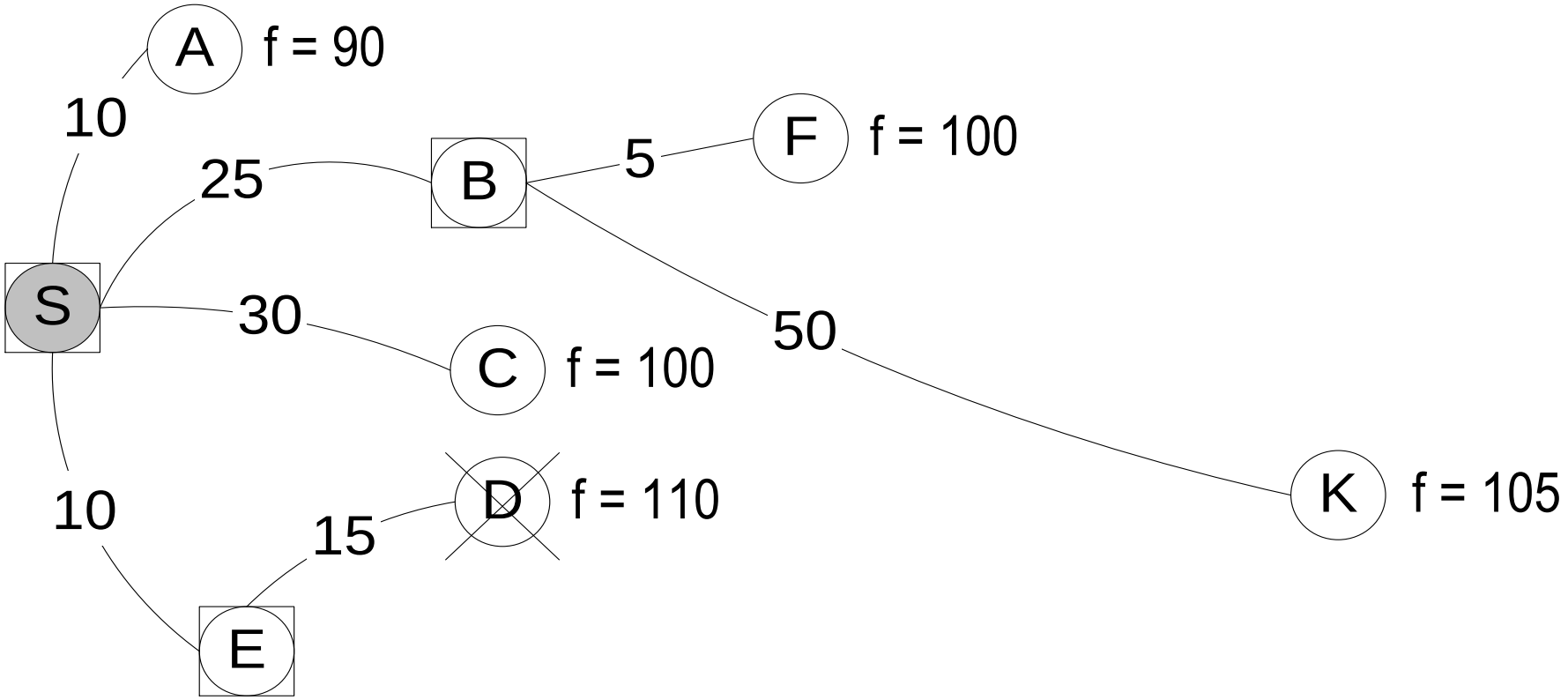
Langkah 1



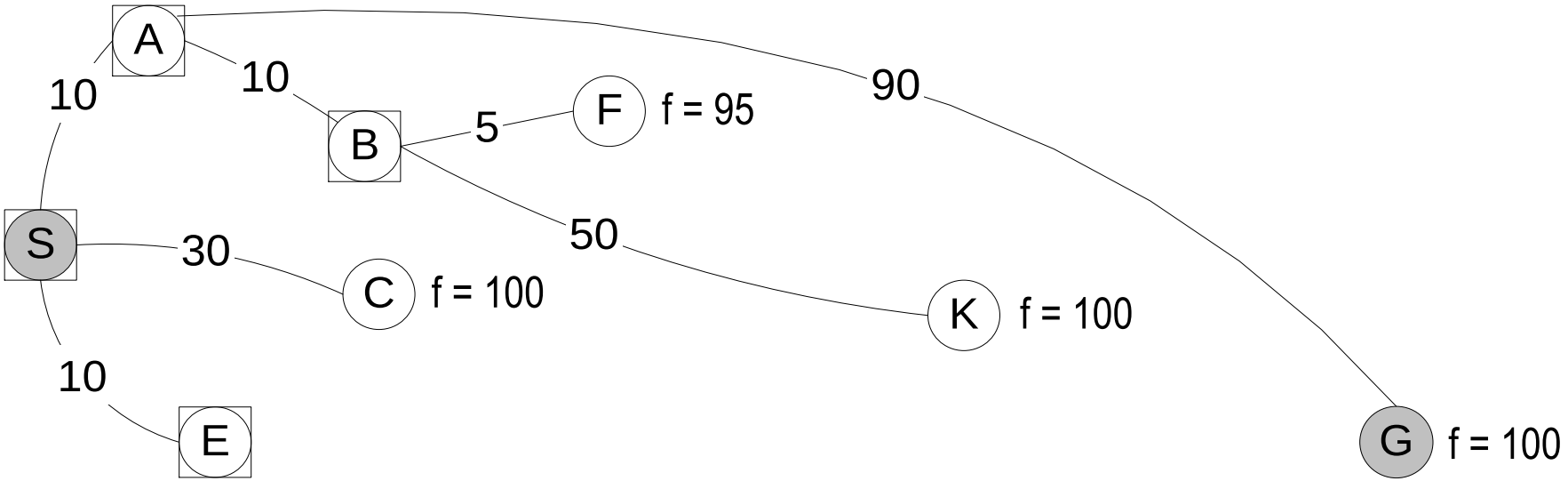
Langkah 2



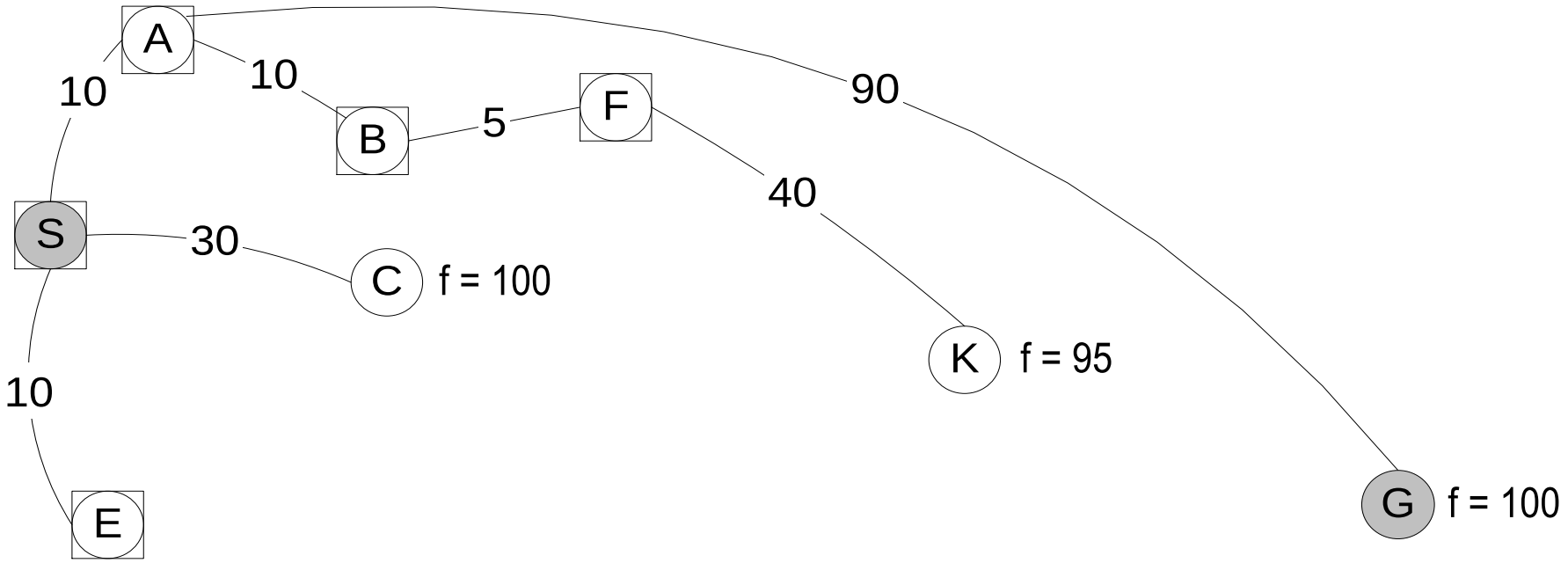
Langkah 3



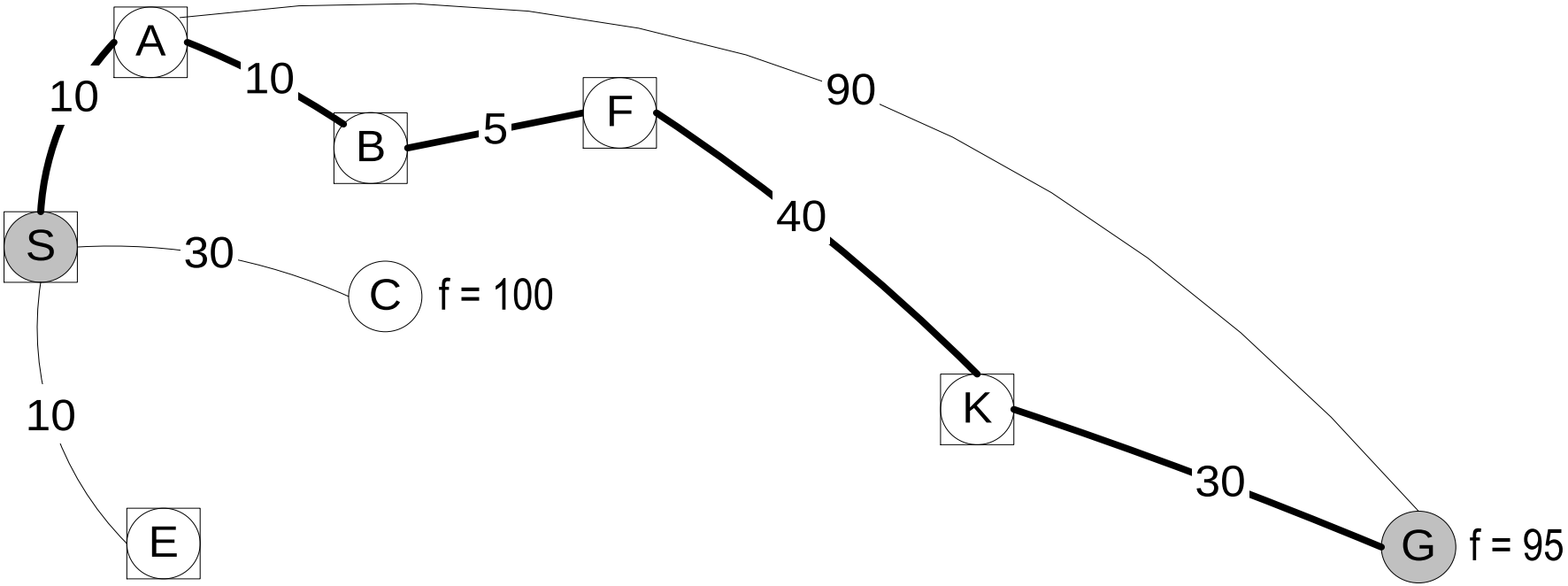
Langkah 4



Langkah 5



Langkah 6



Total Biaya = 95

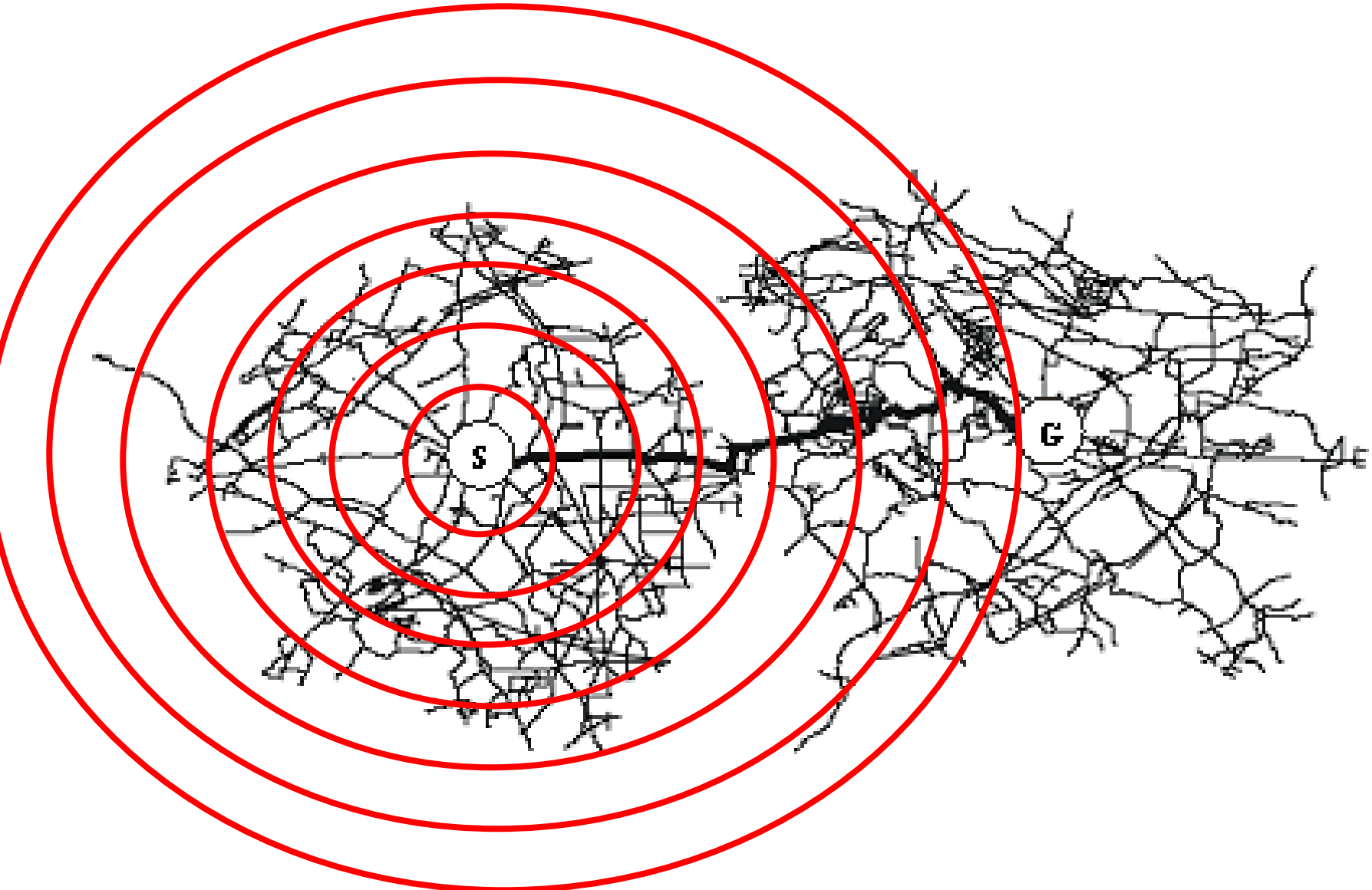
Biaya optimal = 95

Optimal jika kapasitas OPEN mencukupi

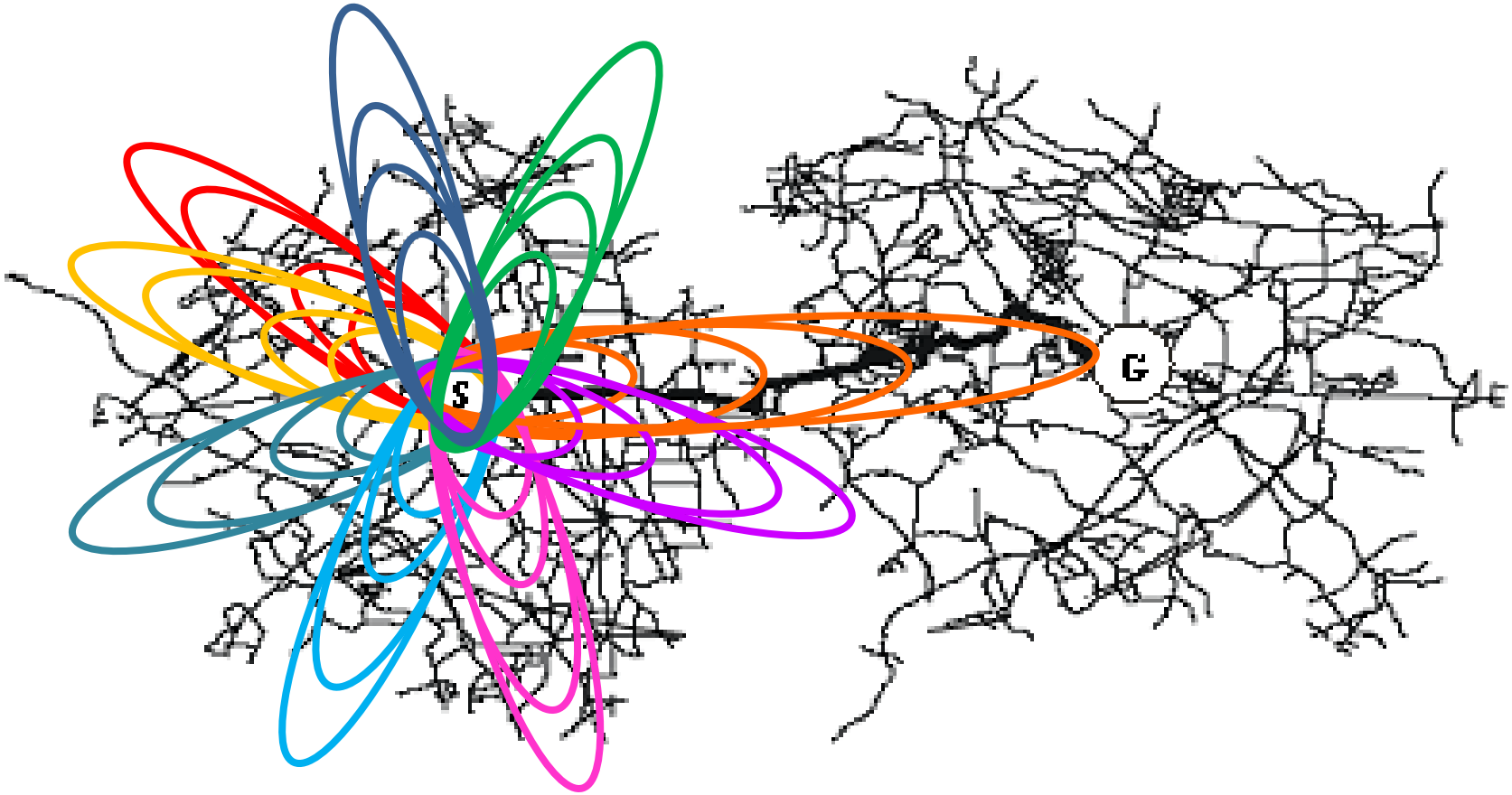
*Beam A** (BA*)

- Bagaimana menentukan jumlah simpul maksimum yang bisa disimpan di dalam *OPEN* ?
- Jika penentuannya kurang tepat, BA* mungkin **tidak complete** dan **tidak optimal**

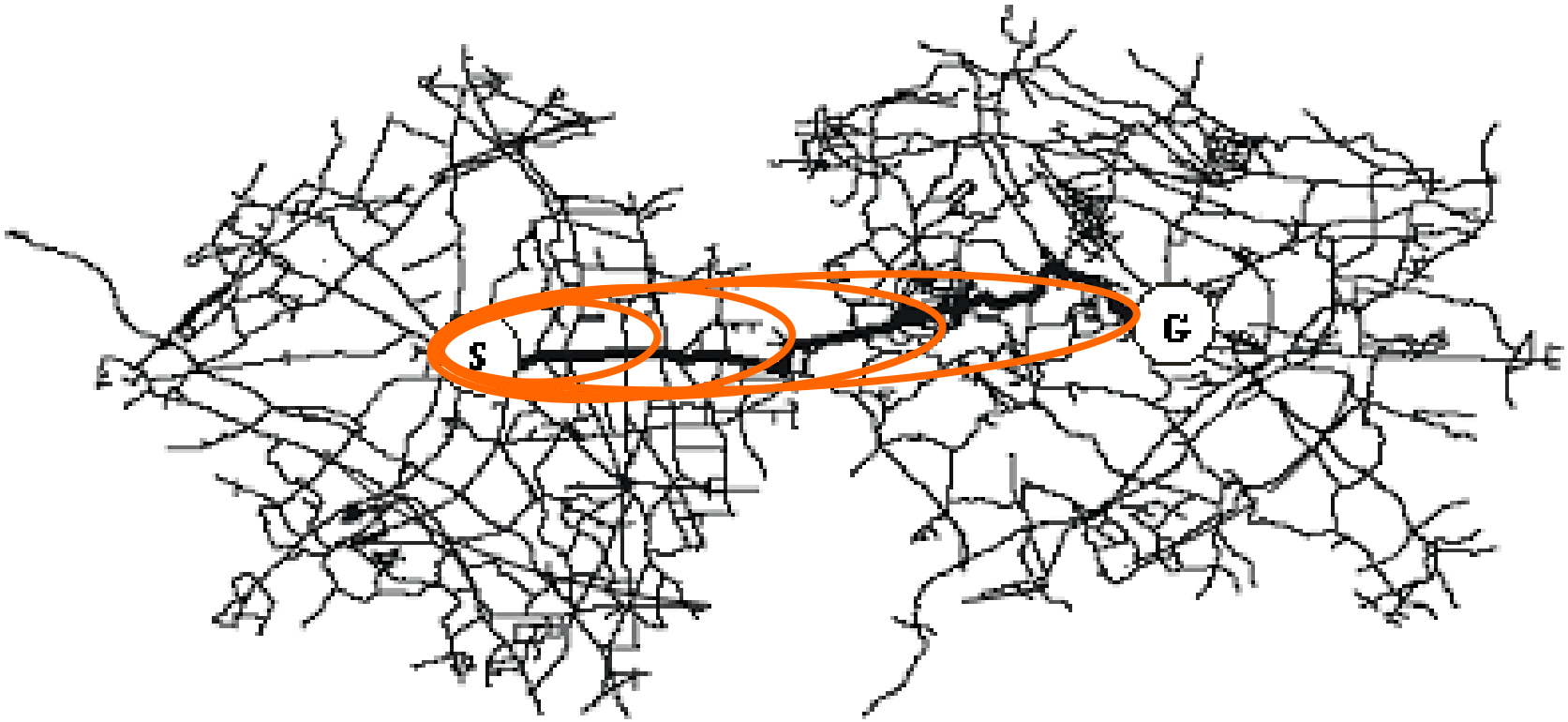
Breadth-First Search (BFS)



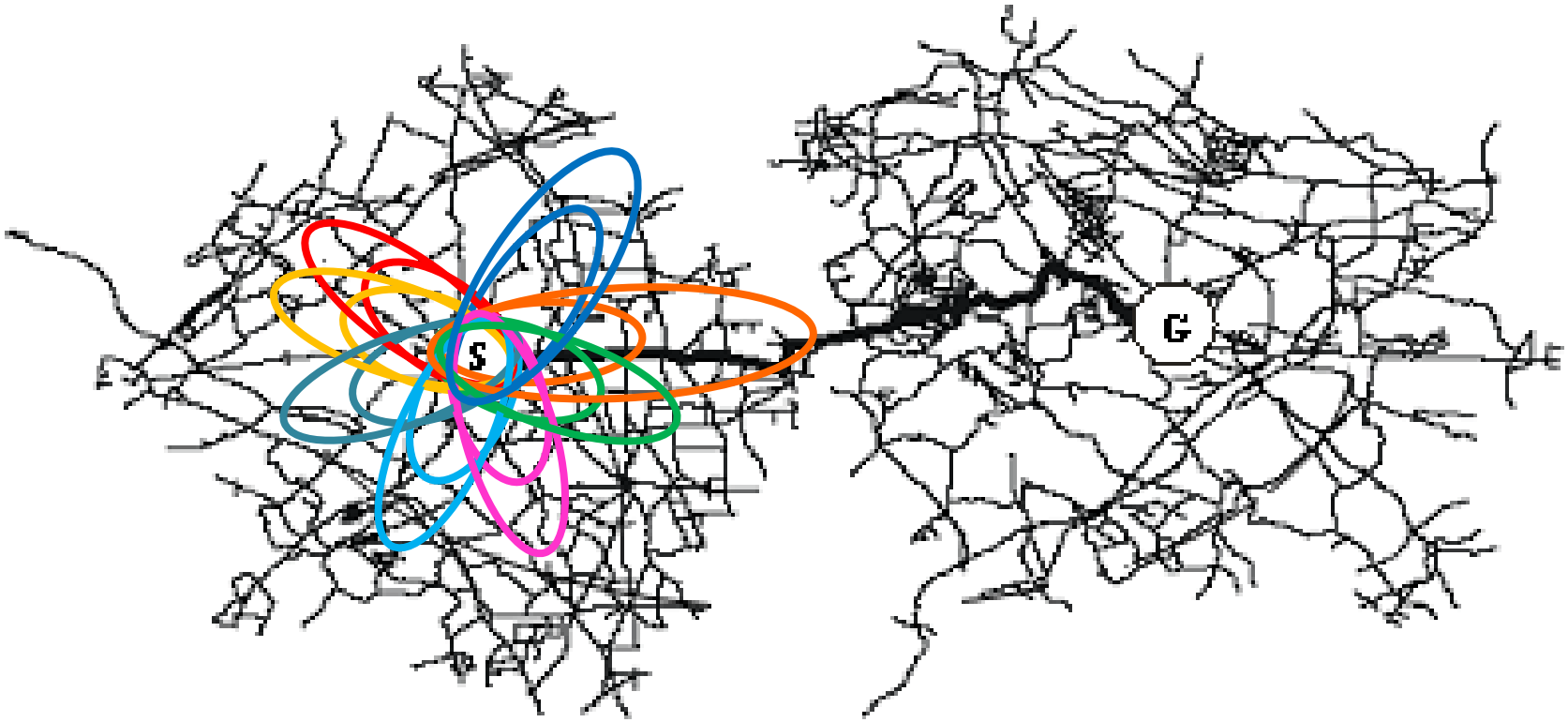
Depth-First Search (DFS) - *Worst-Case*



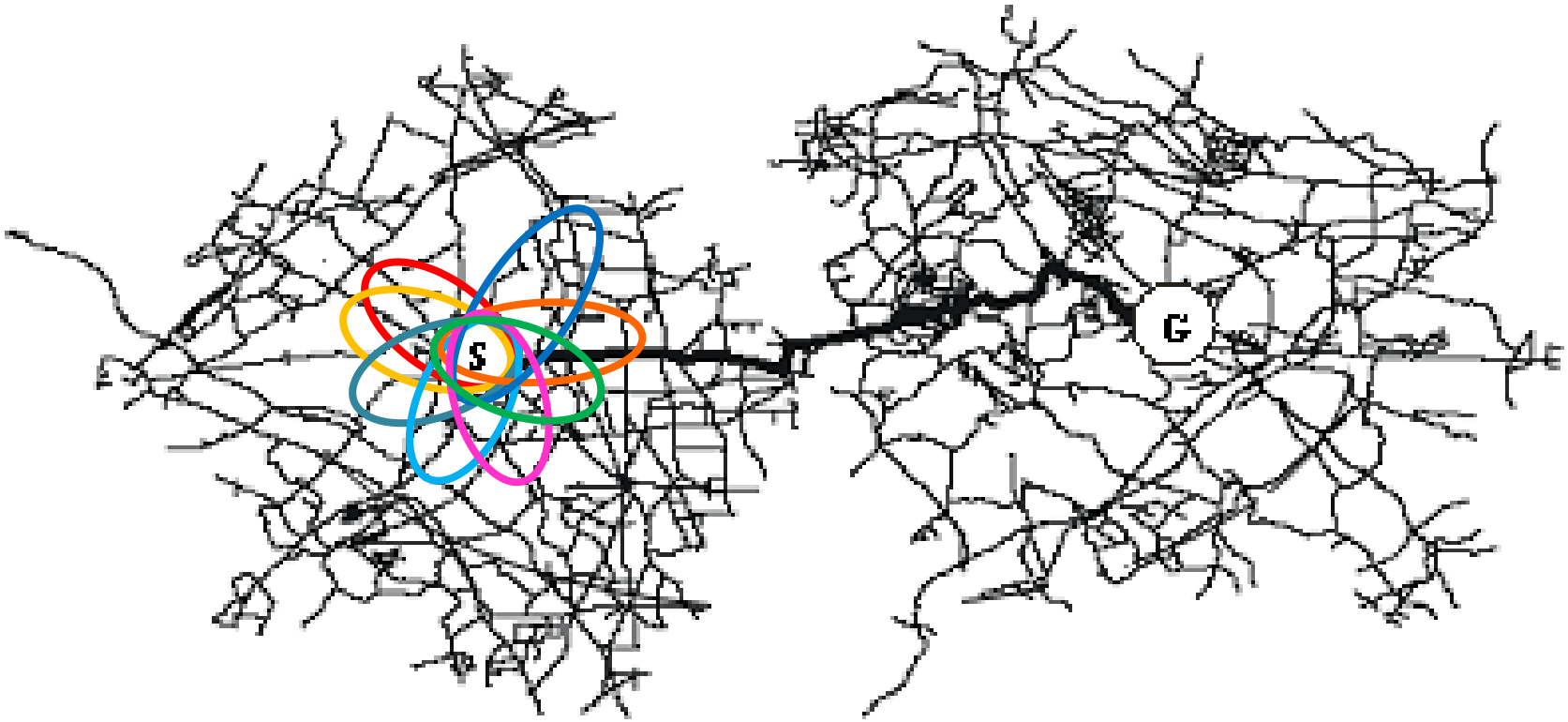
Depth-First Search (DFS) - *Best-Case*



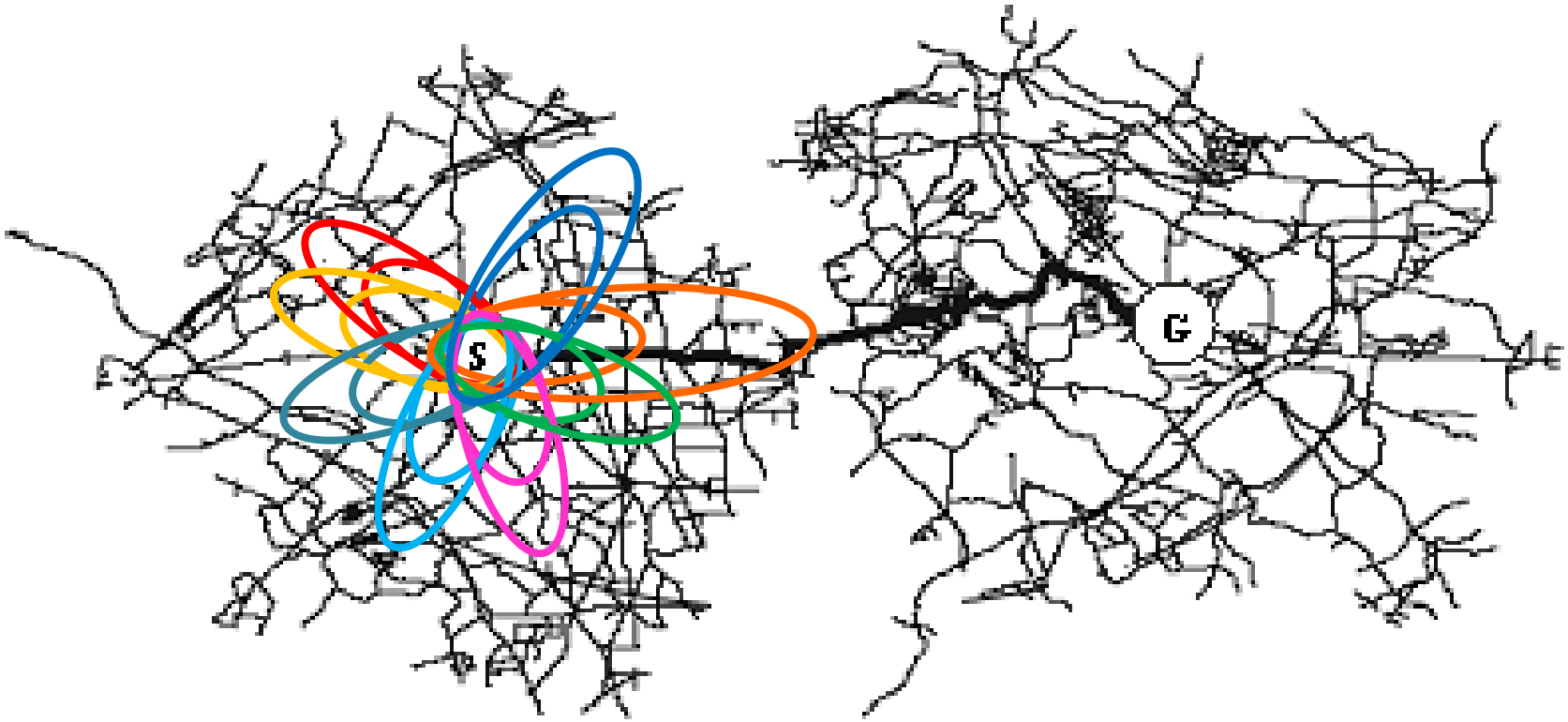
Depth Limited Search (DLS): L = 2



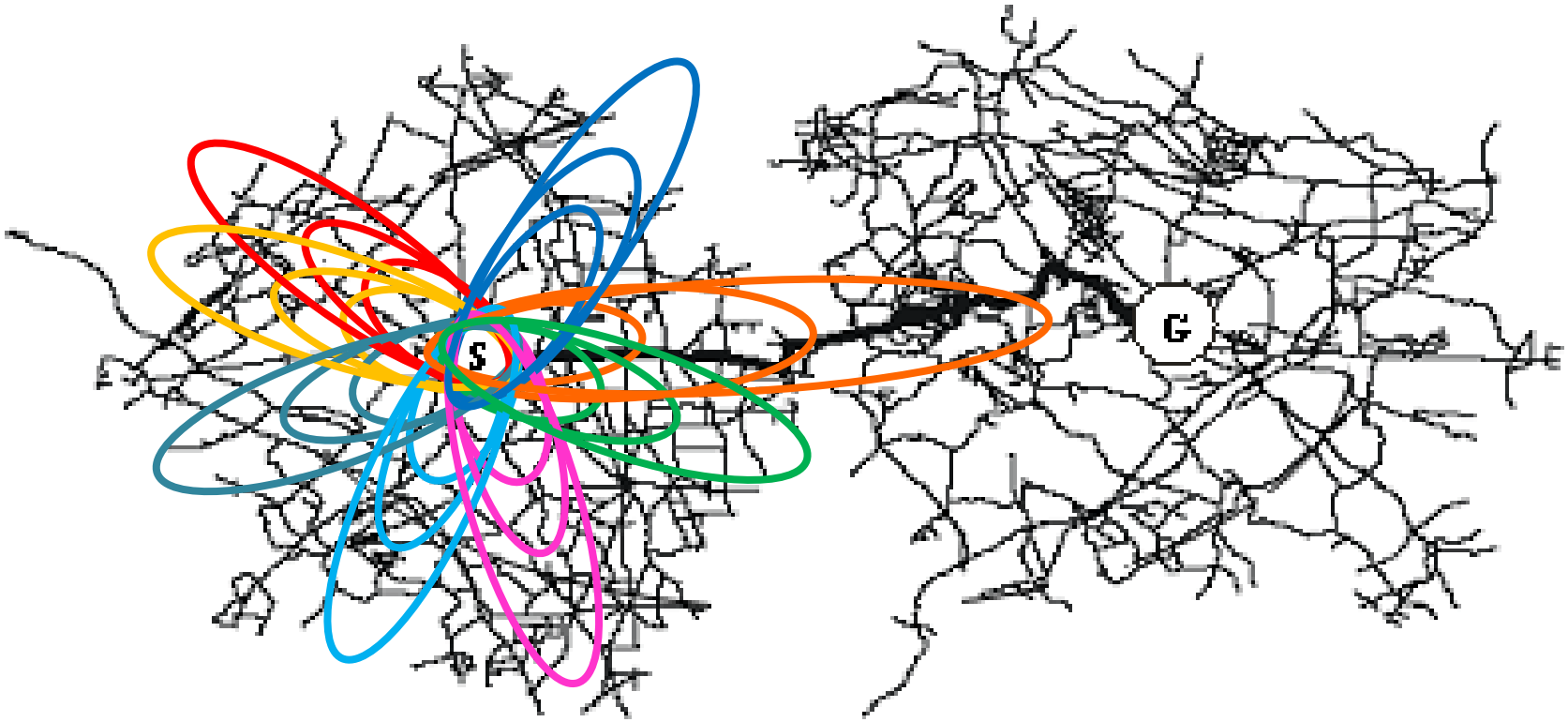
Iterative Deepening Search (IDS): L = 1



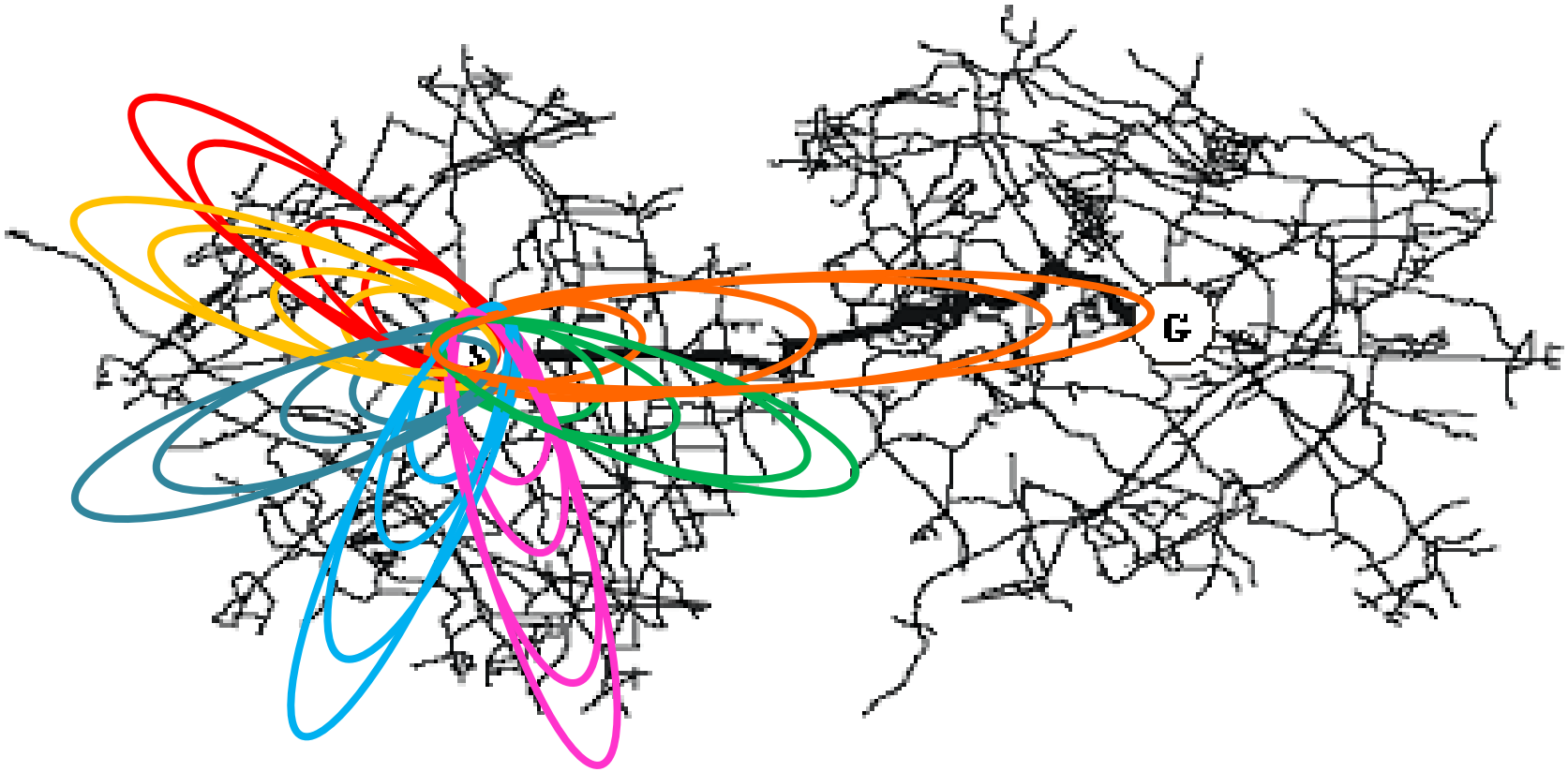
Iterative Deepening Search (IDS): L = 2



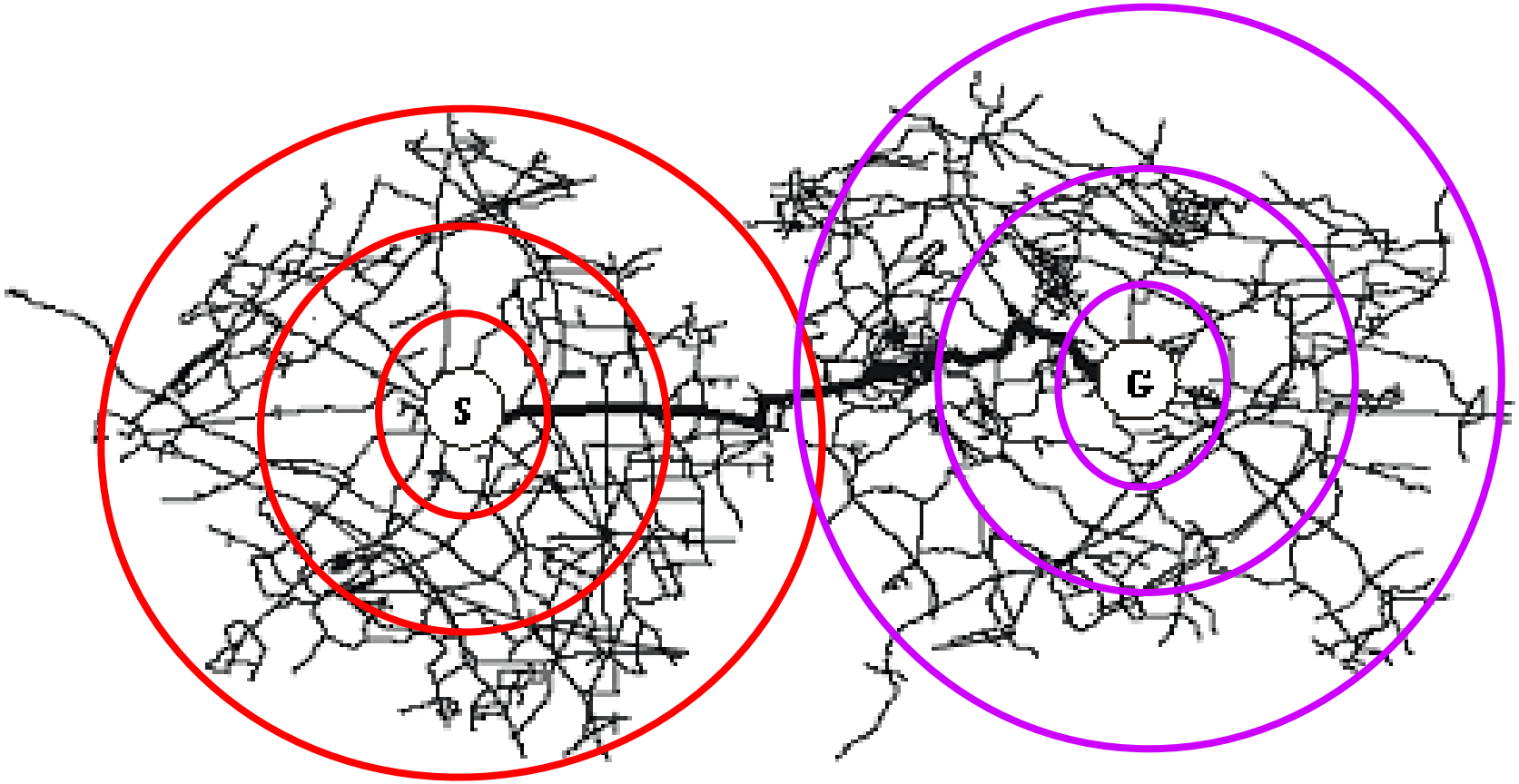
Iterative Deepening Search (IDS): L = 3



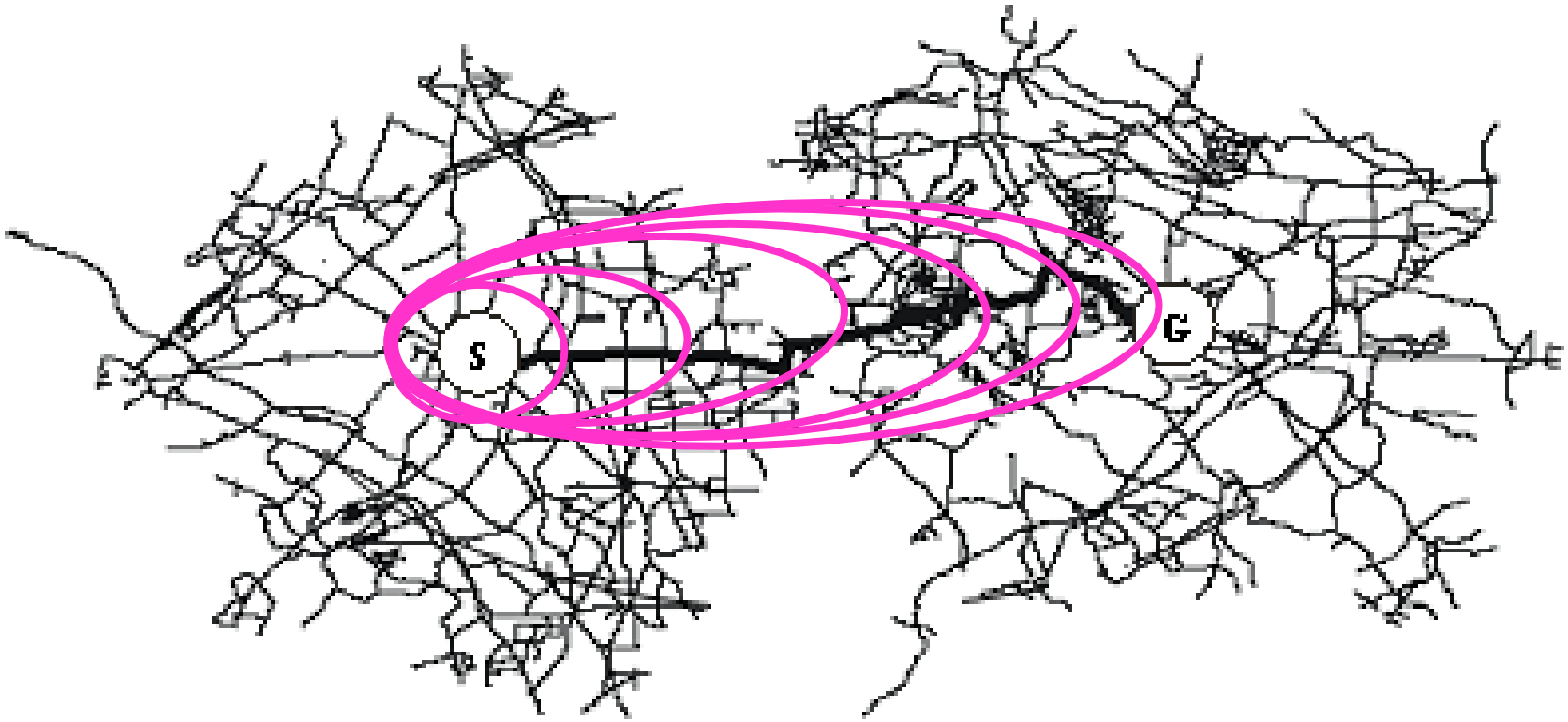
Iterative Deepening Search (IDS): L = 4



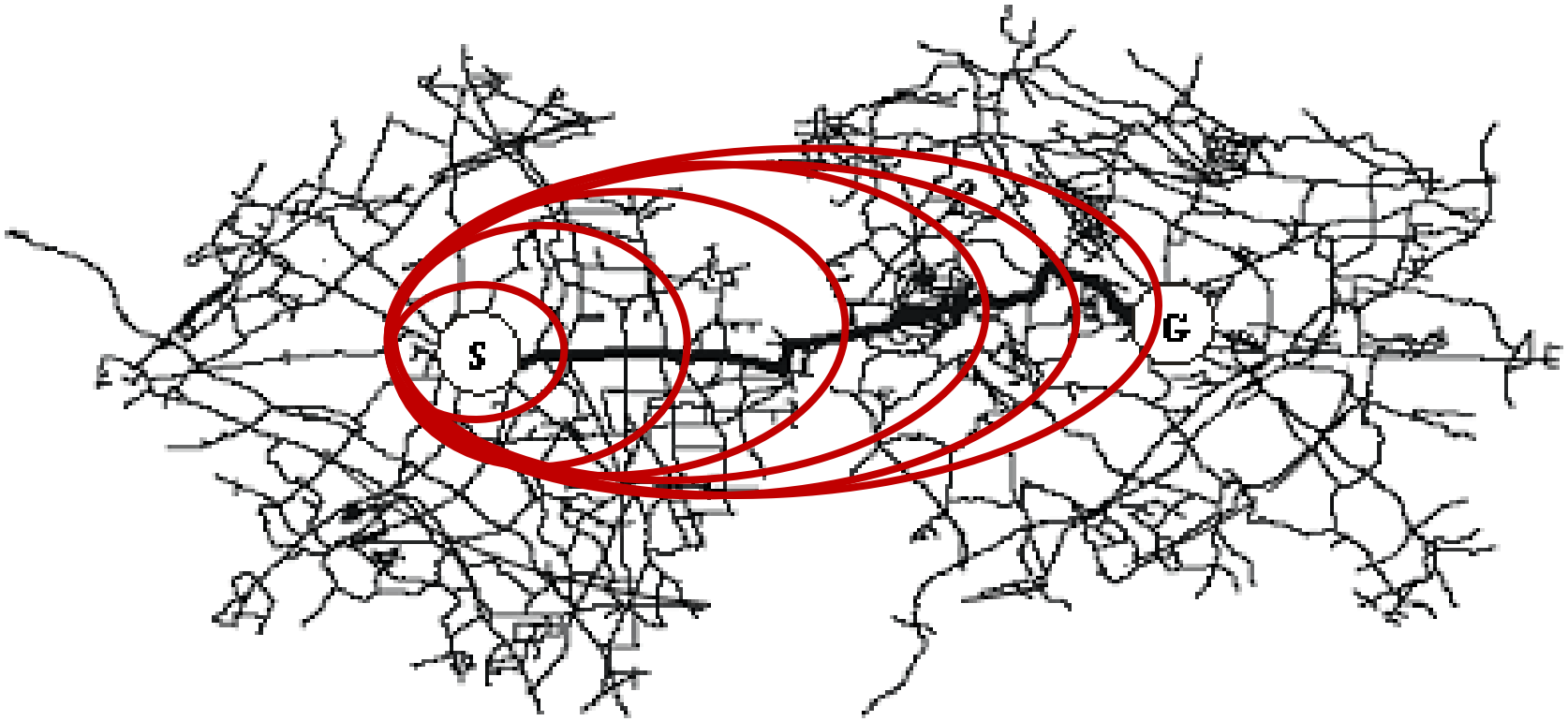
Bi-directional Search (BDS)



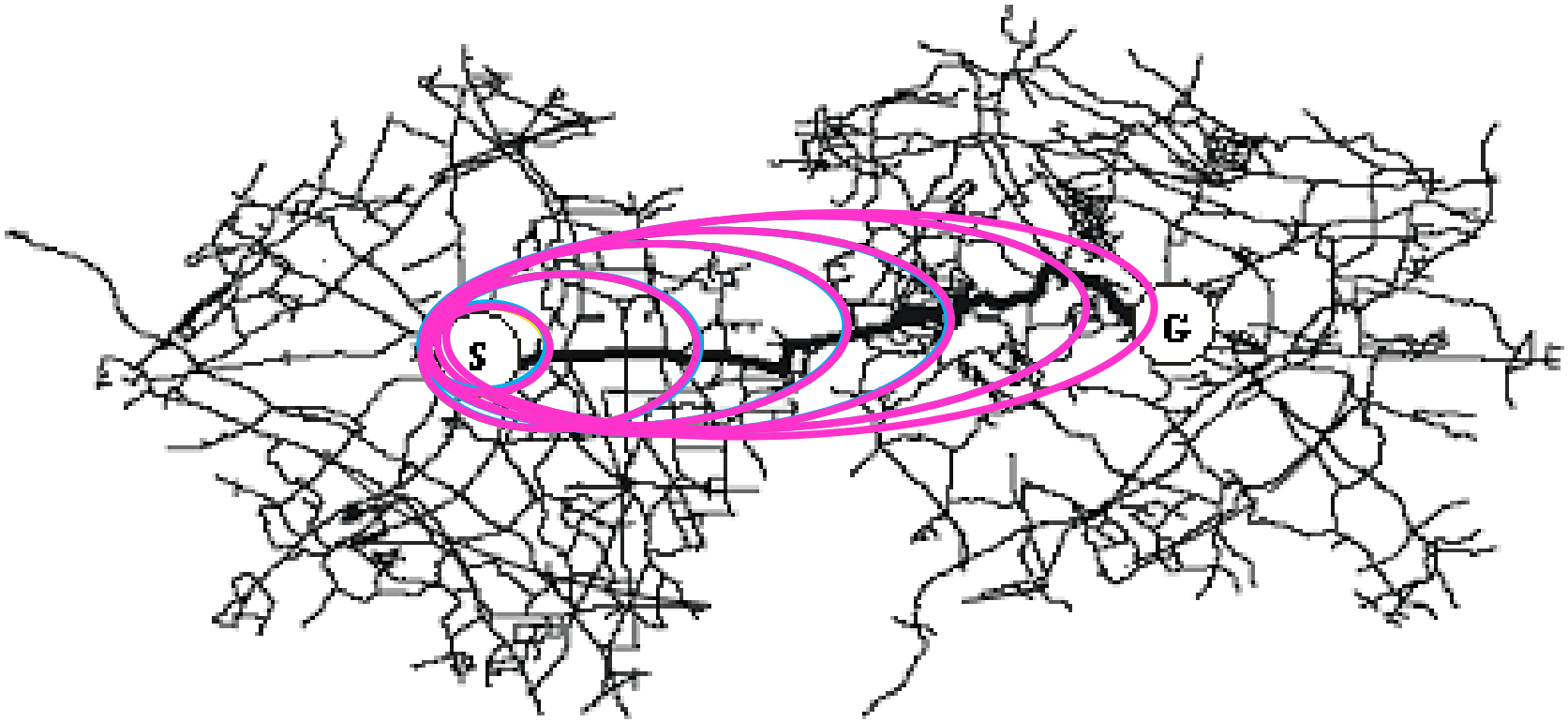
Greedy Best First Search



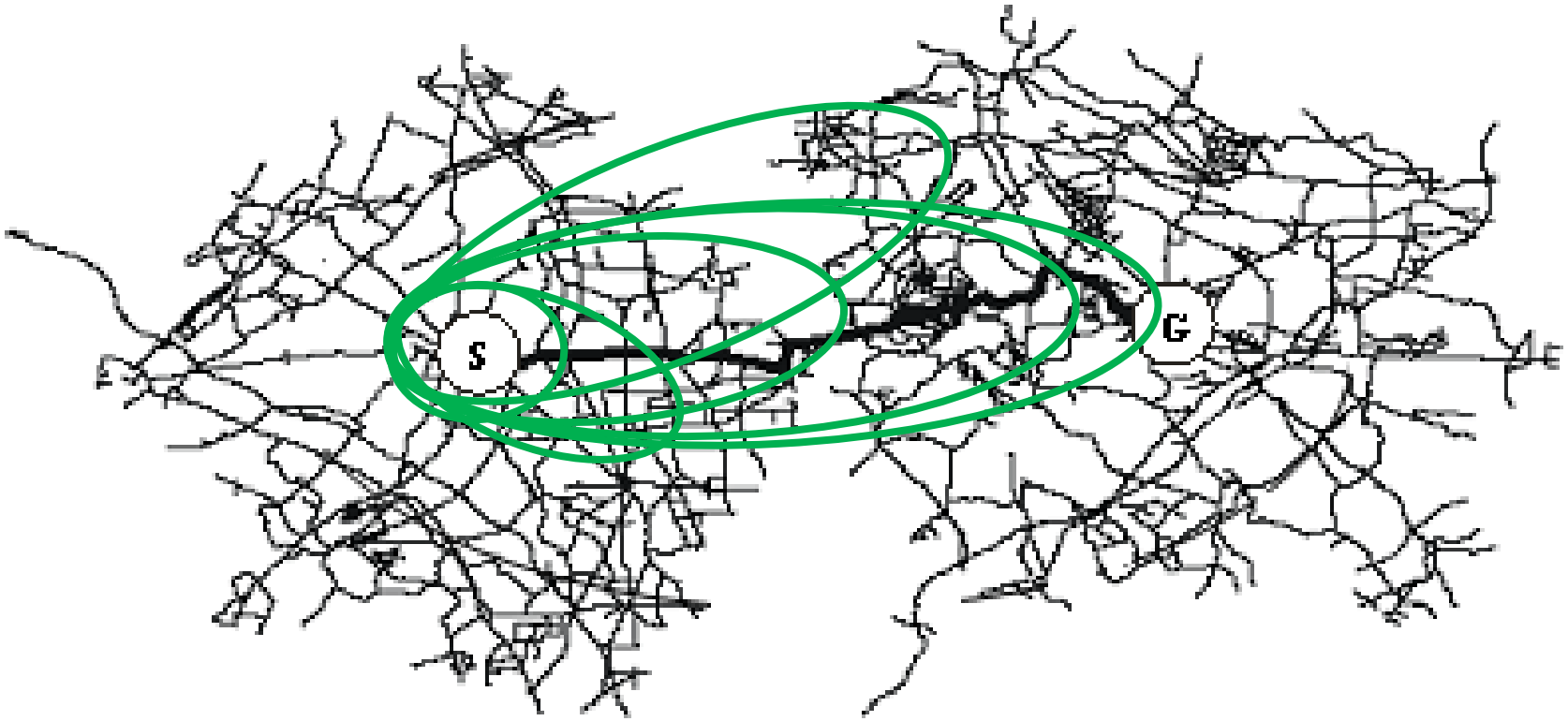
A*



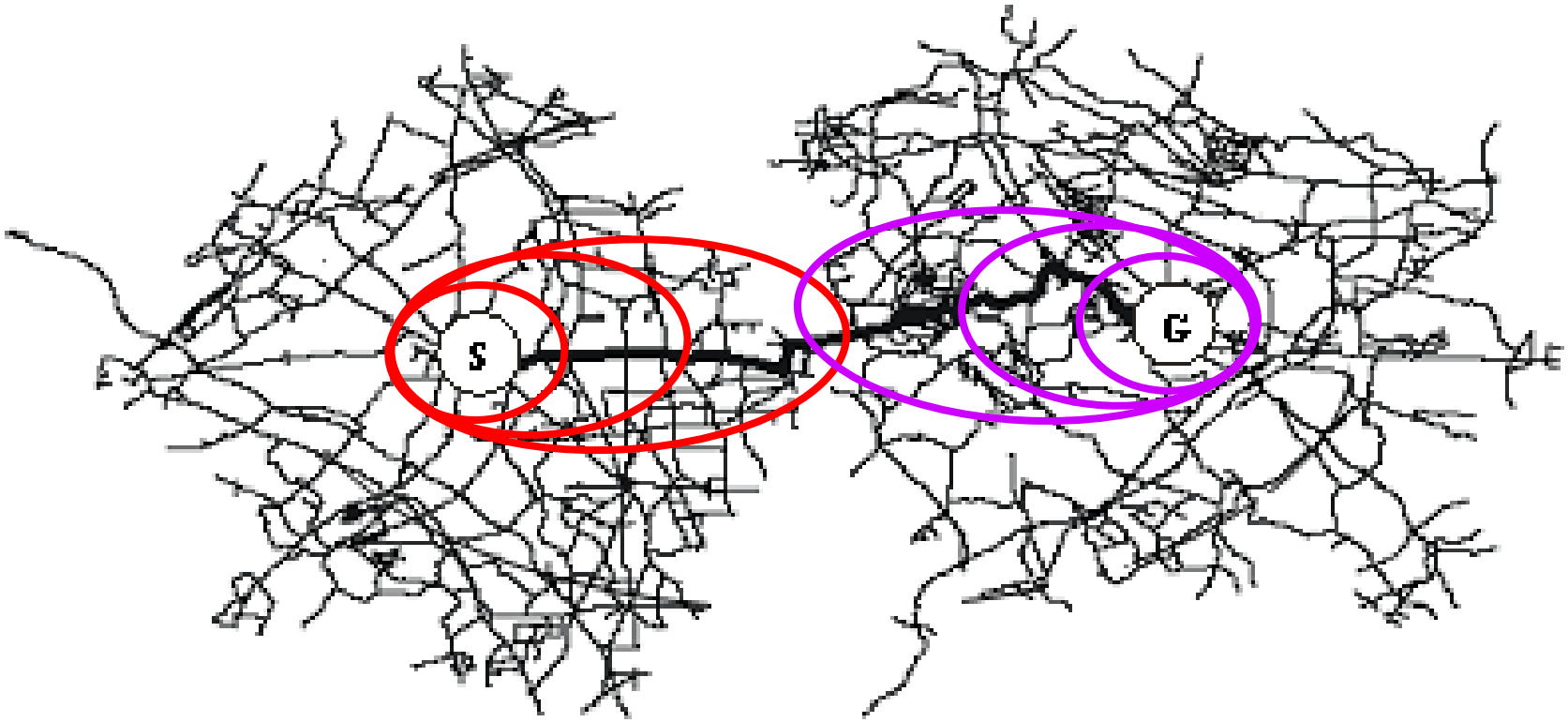
IDA*



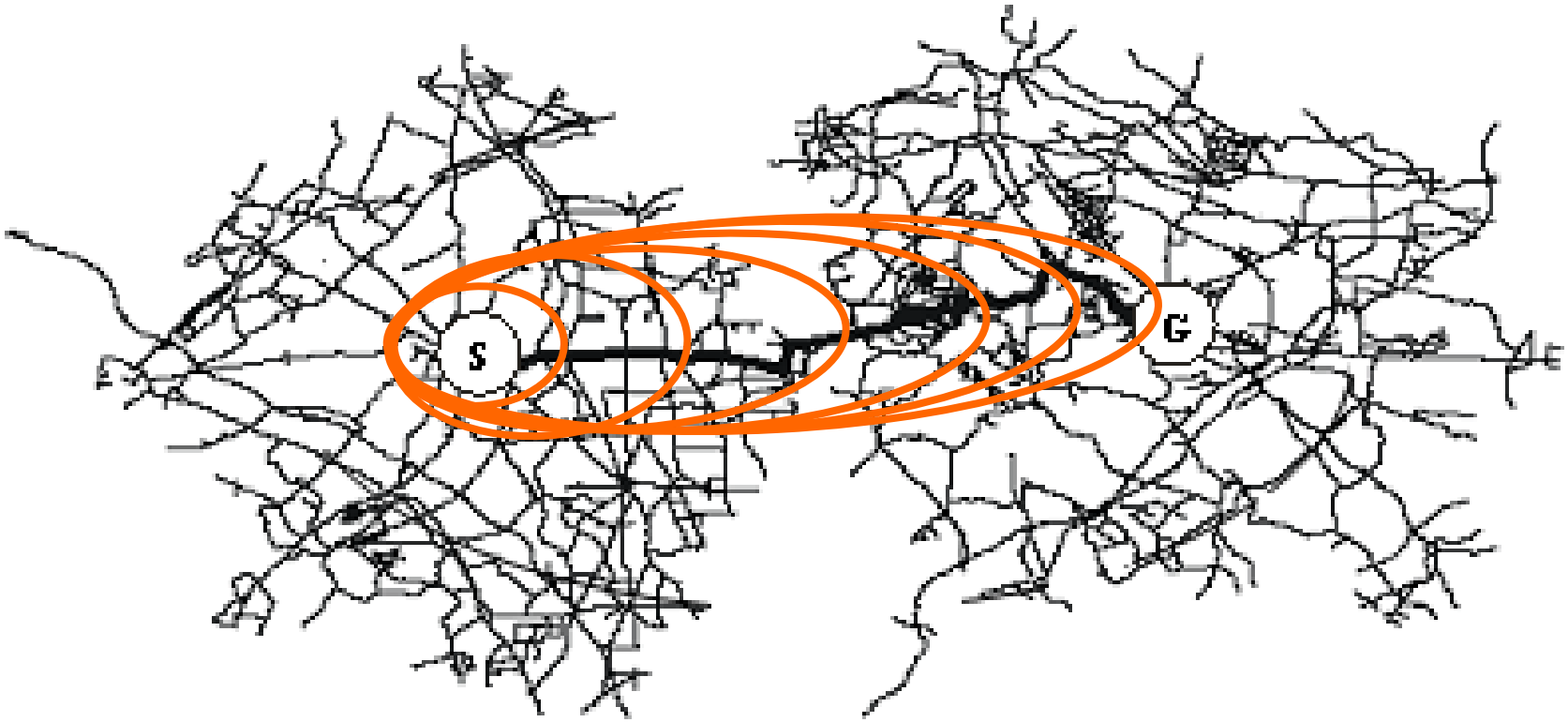
SMA*



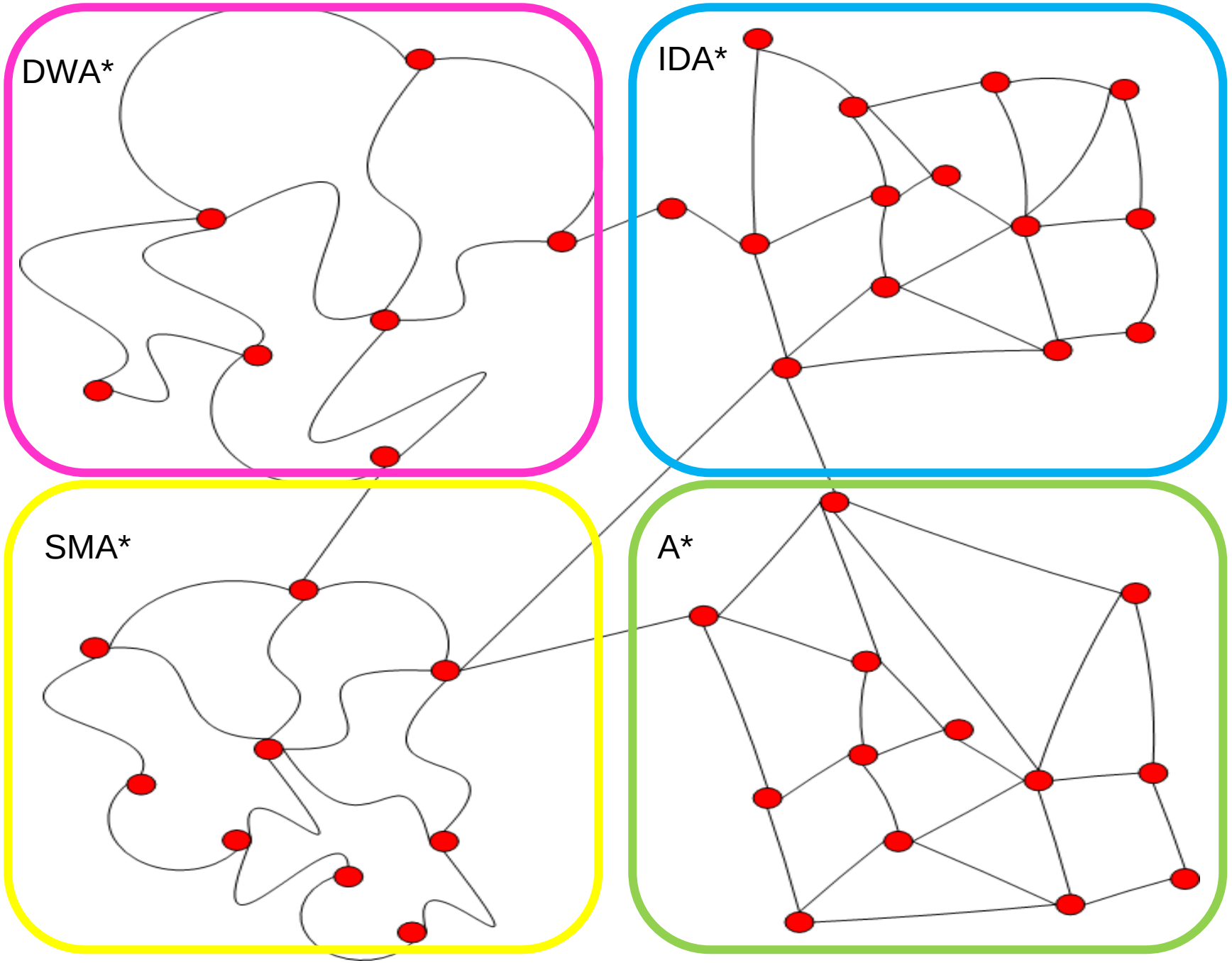
MBDA*



Beam A*




BEDOK NORTH ST 1
NORTH AVE 1
Gaden Hill PK
Bedok Adventure Park
Bedok Town Gdn
Skate Pk
Bedok Community Library
East Coast Singhealth Bedok Polyclinic
Bedok Central P.O.
Geylang United FC
Heavy Vehicle CP
Princess Theatre
BEDOK NTH DR
BEDOK BUS INTERCHANGE
勿洛巴士转换站
BEDOK CENTRAL
Bedok Tennis Ctr
Bedok Swimming Cplx
Bedok Sports Hall
Bedok Stadium
Bedok Sports & Fitness Ctr
CHANGI
UPPER CHANGI
NEW UPPER CHANGI RD
EW5
BEDOK
TAXI
Ansar Gdns
CHEE AVE
Chai Chee NPP
Tampines Pr Sch (Temporary)
淡滨尼小学(临时)
Bedok Green Pr Sch
育青小学
Bedok CC
BEDOK SOUTH AVE 2
Pasir Ris Sec Sch (temporary)
巴西立中学(临时)
Panasonic
Yokogawa
SCS Living Technology
Fujitec
Hamilton Sundstrand
Hitachi Chemical
Plant Nurseries
Esso
Bedok NPP
Bedok Lutheran Ch
Lions Home for The Elders
BEDOK SOUTH RD
BEDOK SOUTH
Temasek Junior College
淡马锡初院
Opera Estate
SENNETT TER
SENNETT AVE
SENNETT LANE
SENNETT DR
LUCKY CRES
LUCKY RISE
AIDA ST
DAFNE ST
DIDO ST
TOSCA TER
CARMEN TER
LAKME TER
27
28
30A CP


DWA*
IDA*
SMA*
A*

Real world

- Mobile Navigation Systems
- Vehicle Routing Problems
- Search Engines
- Games


南
9