Universidade Federal de Viçosa
Departamento de Informática
Programa de Pós-Graduação em Ciência da Computação

On Selecting Heuristic function Subsets for Domain-Independent Planning

Marvin Abisrror - 84075

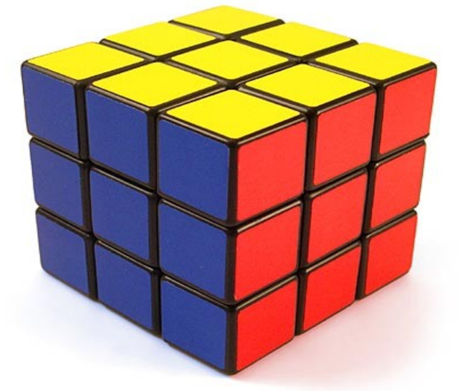
Advisor:

Levi Henrique Santana de Lelis

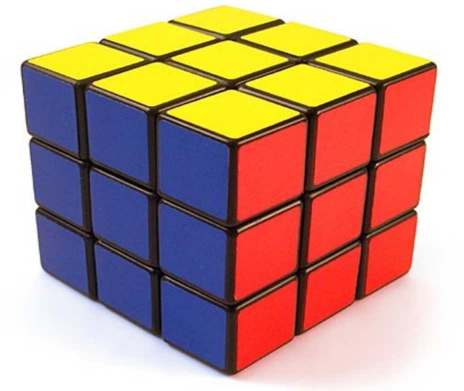
Index

- Heuristic Search
- Greedy Heuristic Selection
- Stratified Sampling (SS)
- Culprit Sampling (CS)
- Experiments
- Conclusions
- References

Heuristic Search

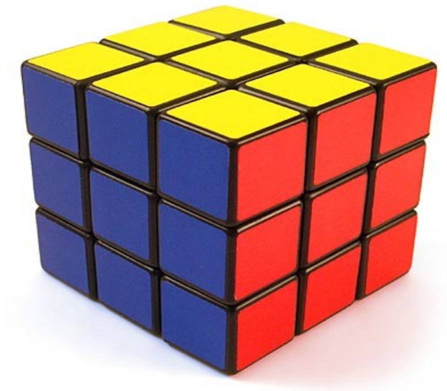


Heuristic Search

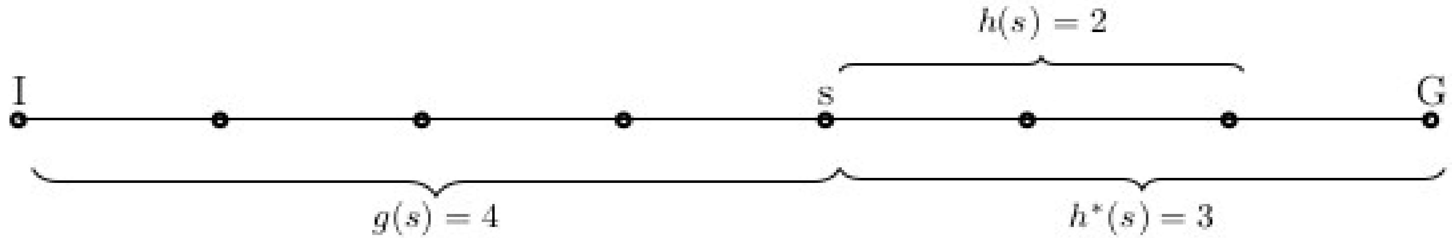


$$f(s) = g(s) + h(s)$$

Heuristic Search



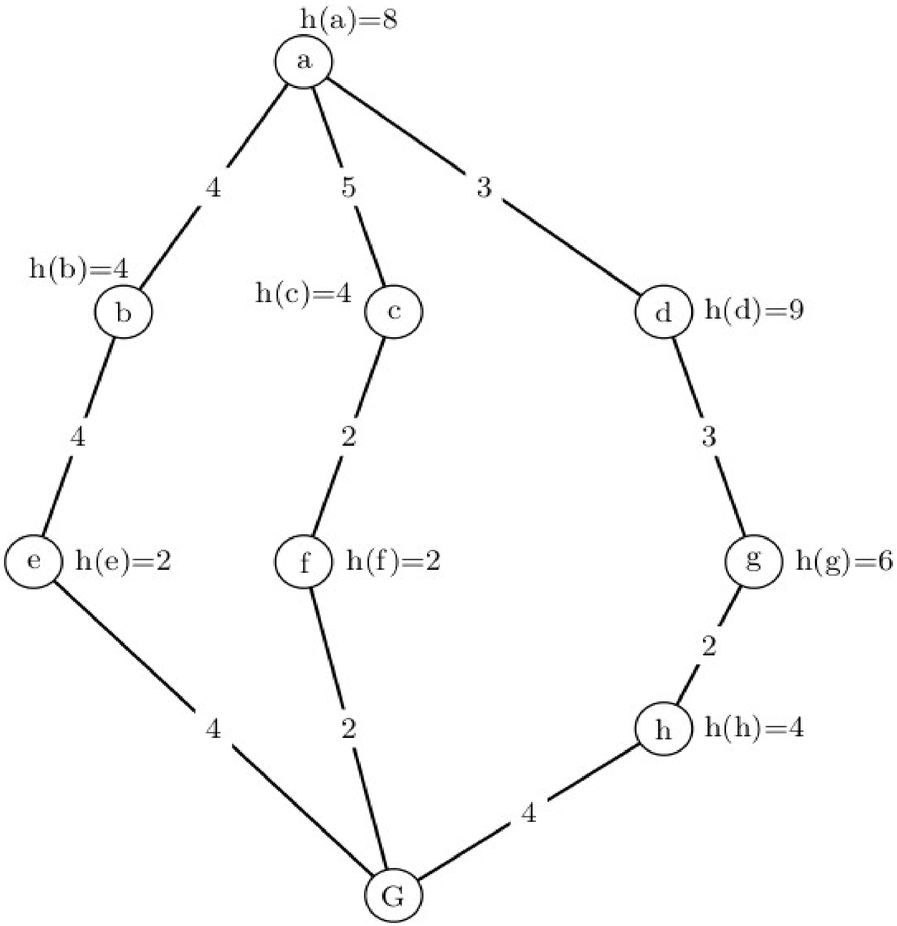
$f(s) = g(s) + h(s)$



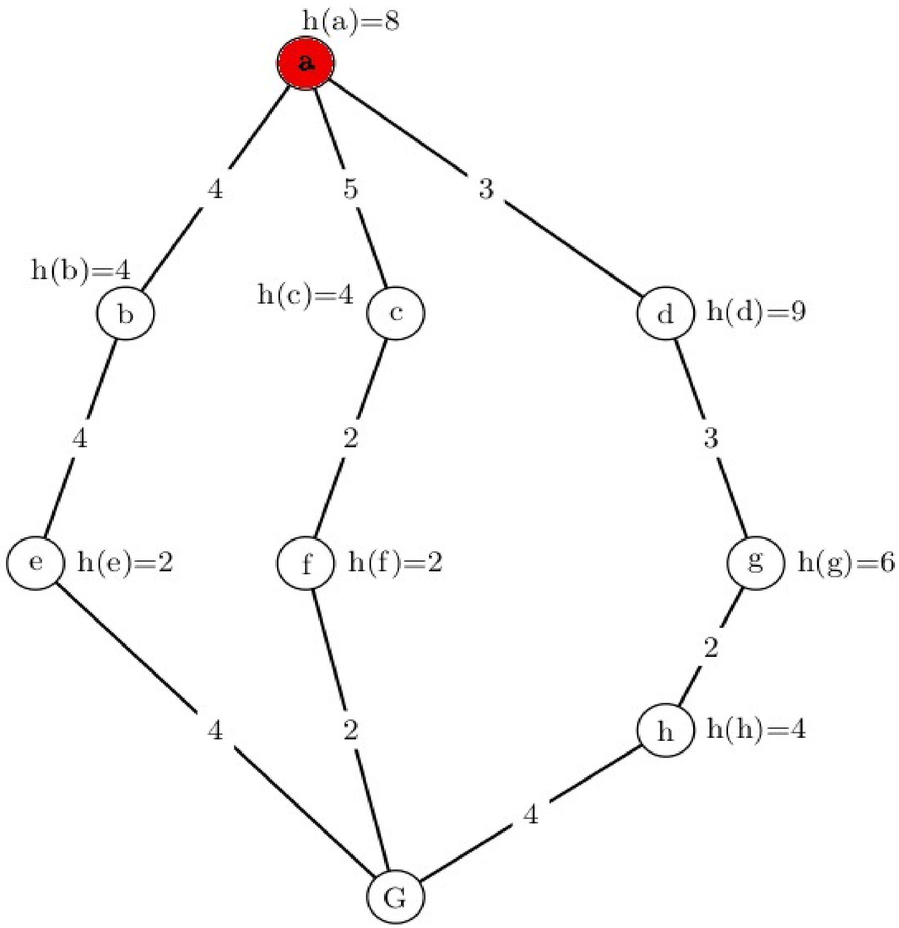
Universidade Federal de Viçosa
Departamento de Informática

A*+h1

A*+h1



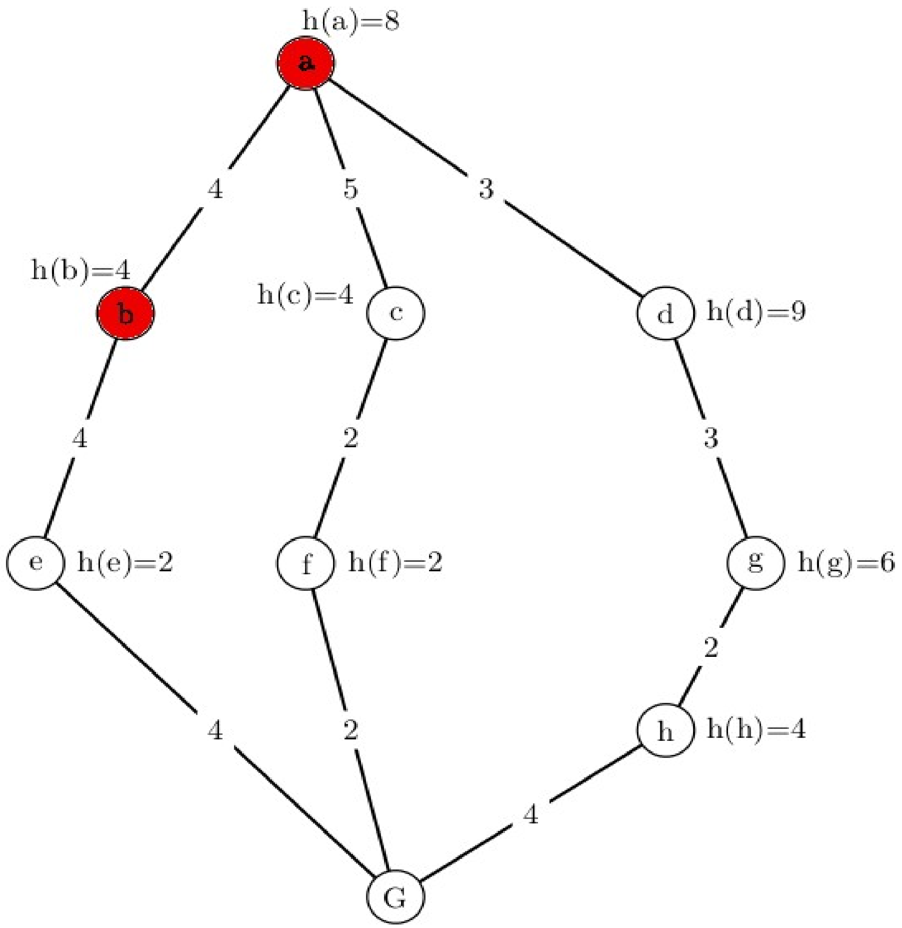
A*+h1



OPEN

f(b)=4+4
f(c)=5+4
f(d)=3+9

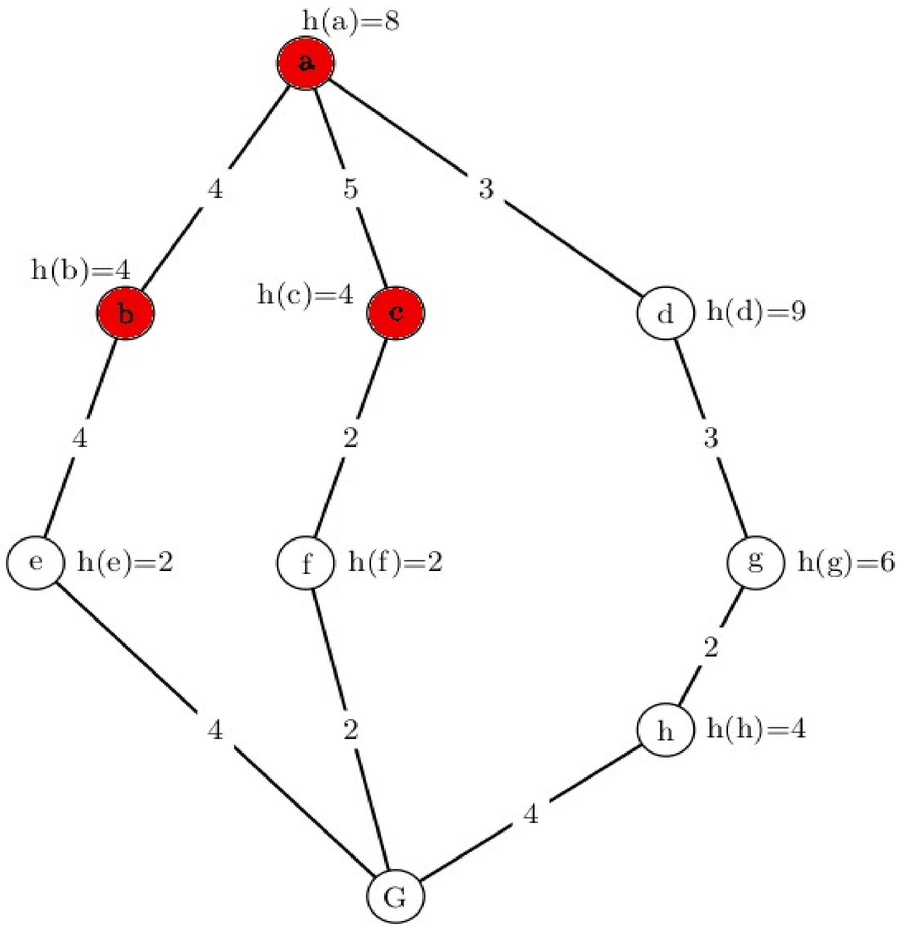
A*+h1



OPEN

f(c)=5+4
f(d)=3+9
f(e)=8+2

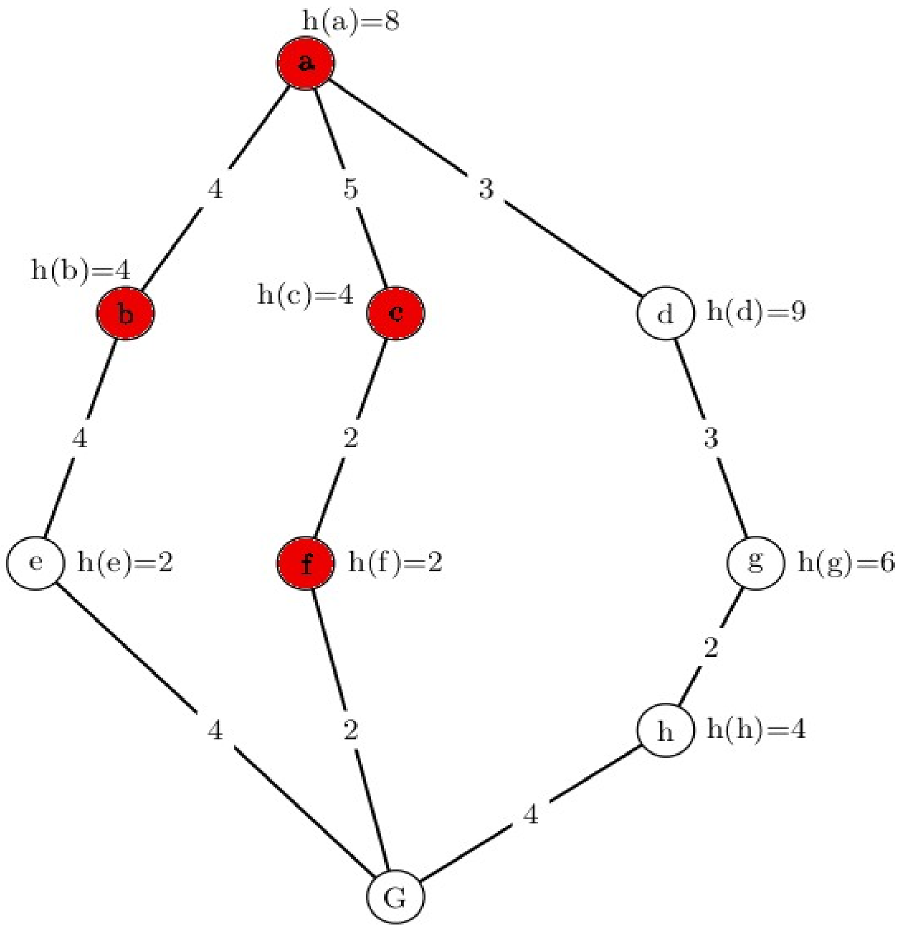
A*+h1



OPEN

f(d)=3+9
f(e)=8+2
f(f)=7+2

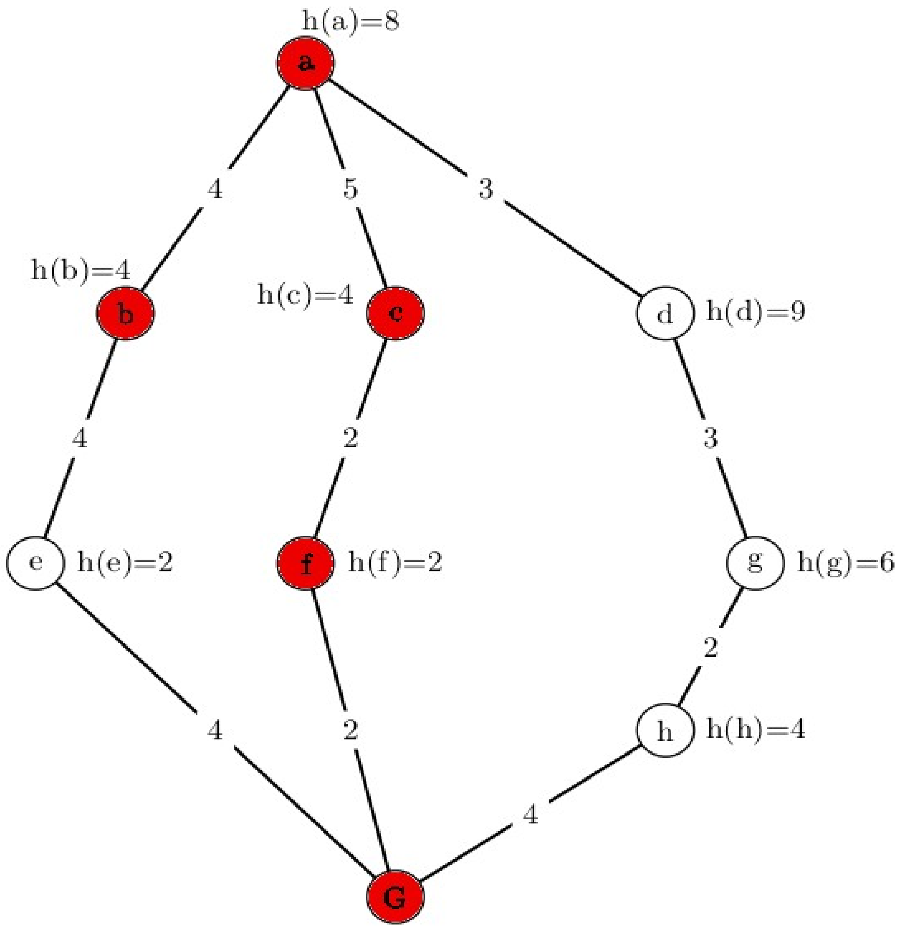
A*+h1



OPEN

f(d)=3+9
f(e)=8+2
f(G)=9+0

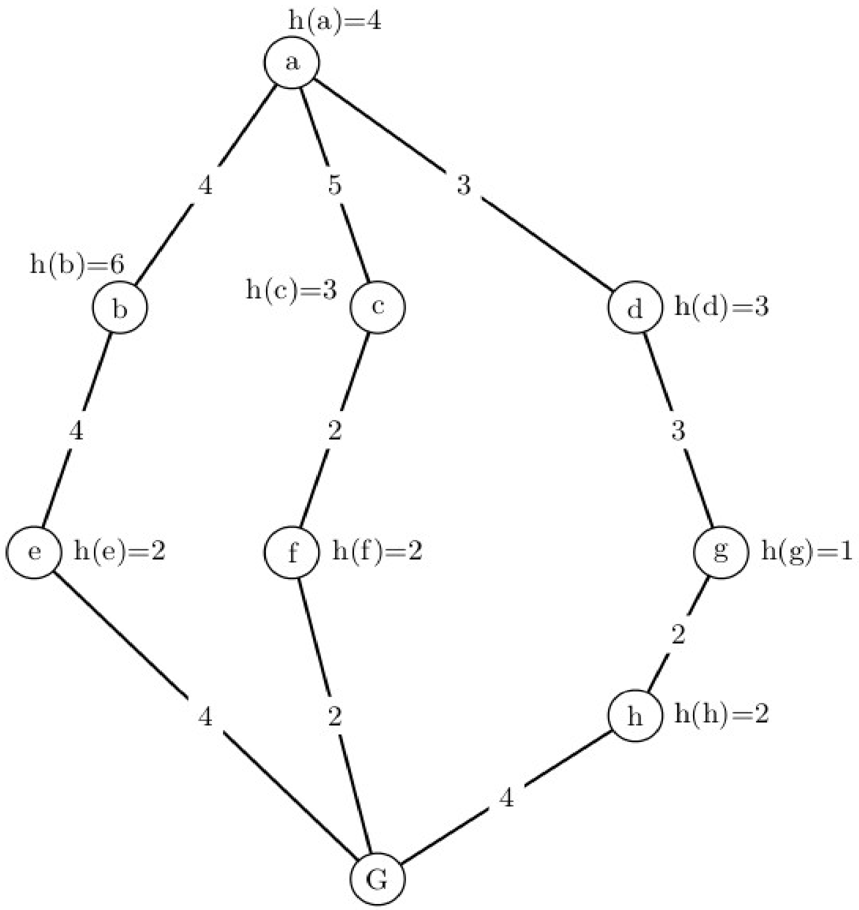
A*+h1



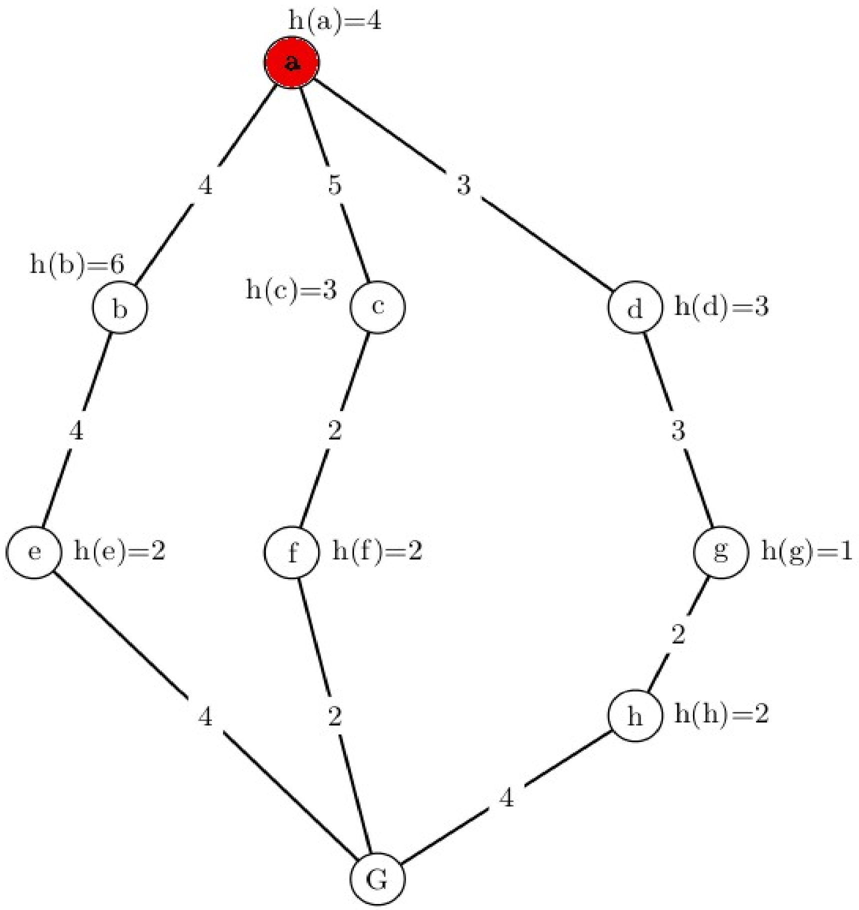
OPEN

f(d)=3+9
f(e)=8+2

A*+h2



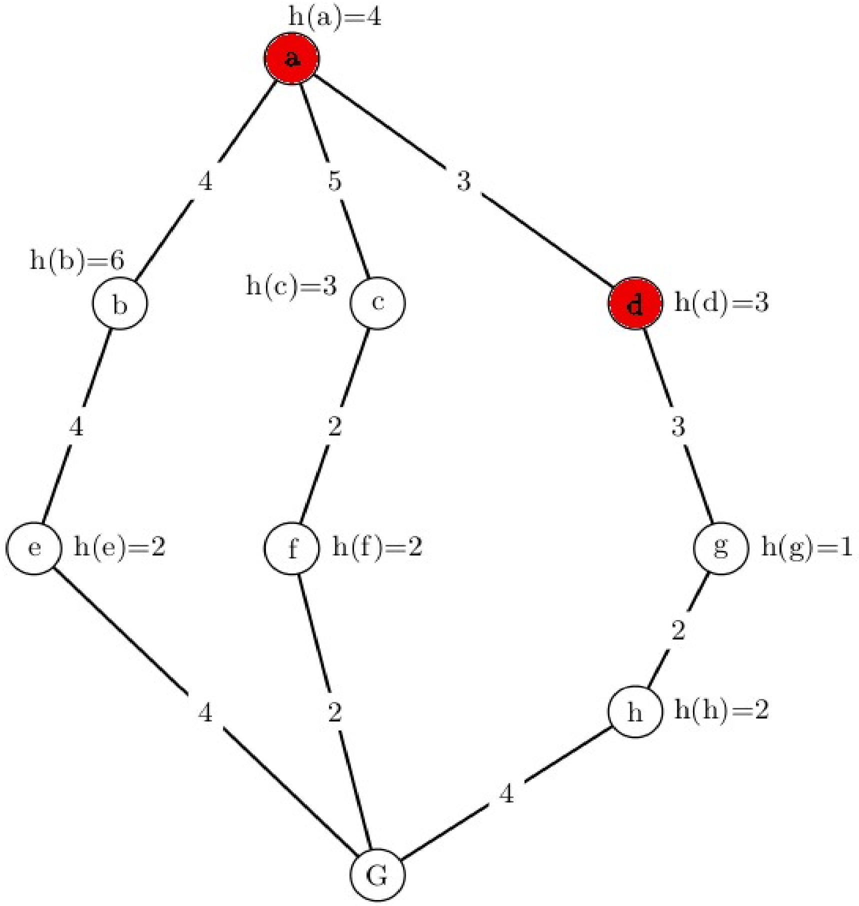
A*+h2



OPEN

f(b)=4+6
f(c)=5+3
f(d)=3+3

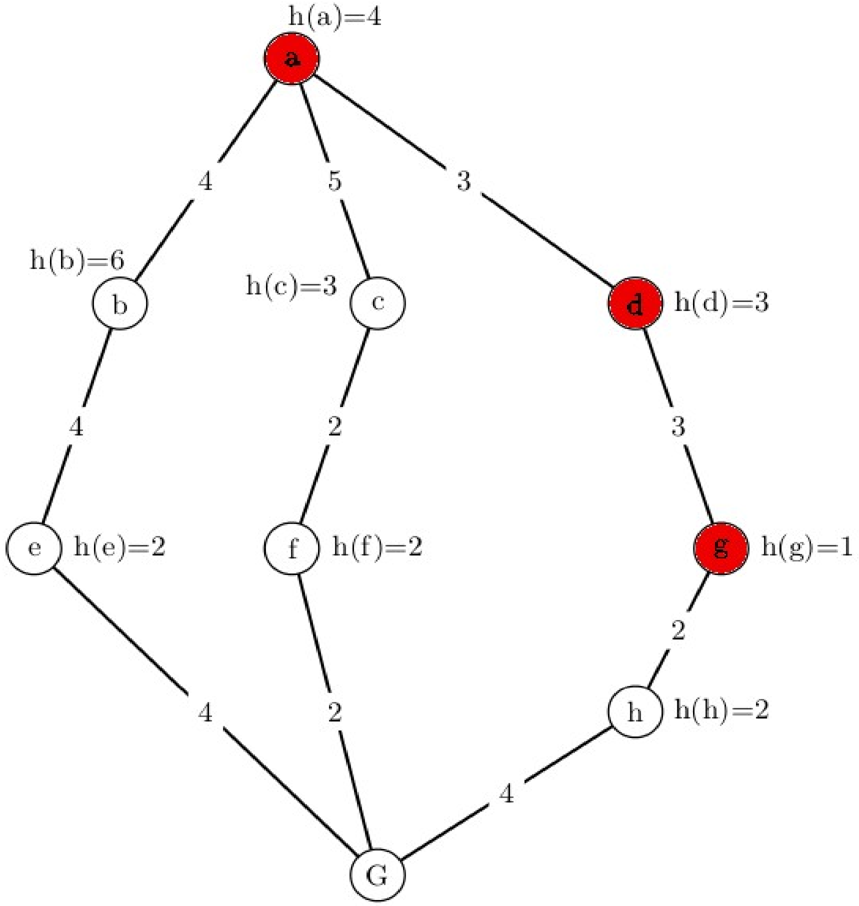
A*+h2



OPEN

f(b)=4+6
f(c)=5+3
f(g)=6+1

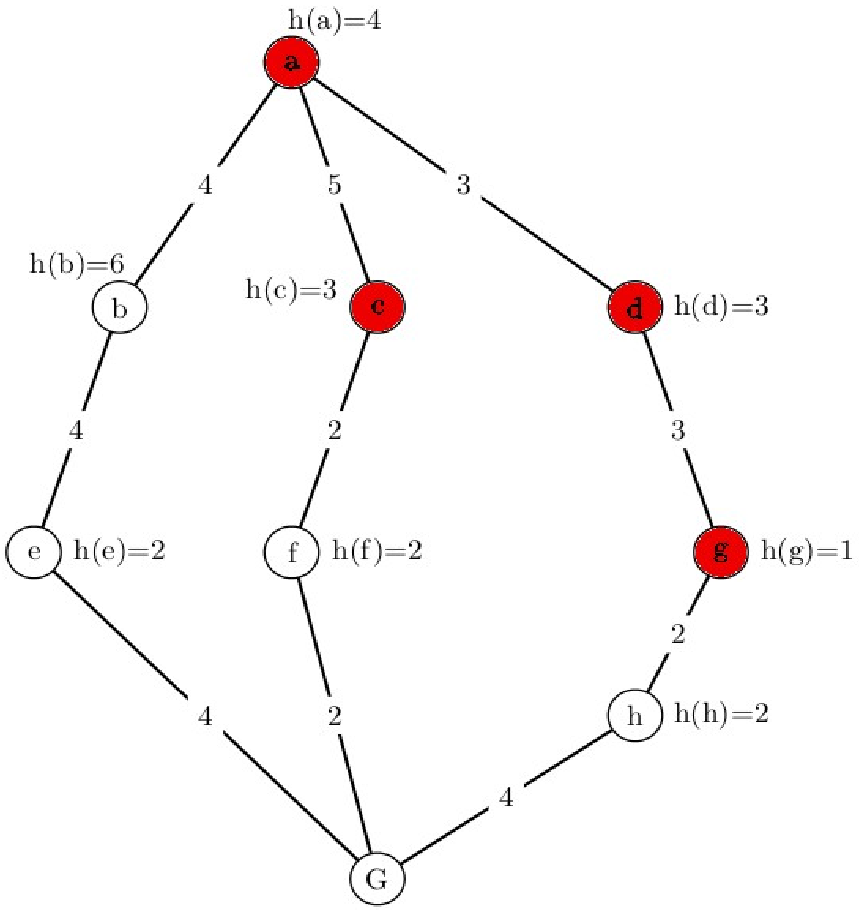
A*+h2



OPEN

f(b)=4+6
f(c)=5+3
f(h)=8+2

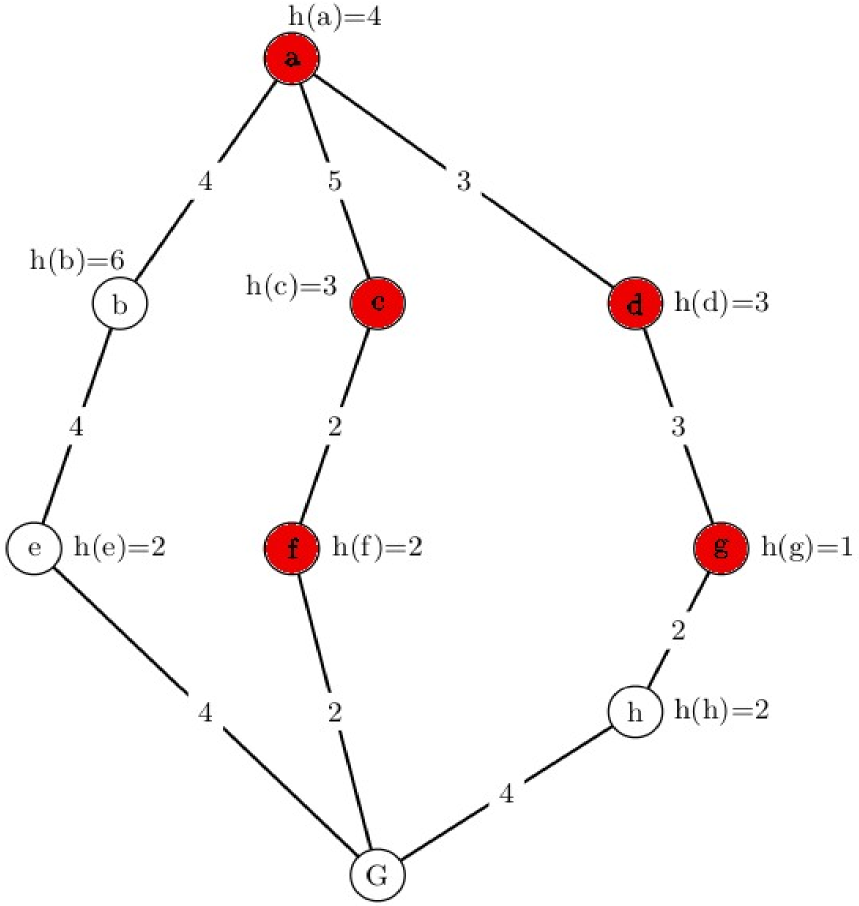
A*+h2



OPEN

f(b)=4+6
f(h)=8+2
f(f)=7+2

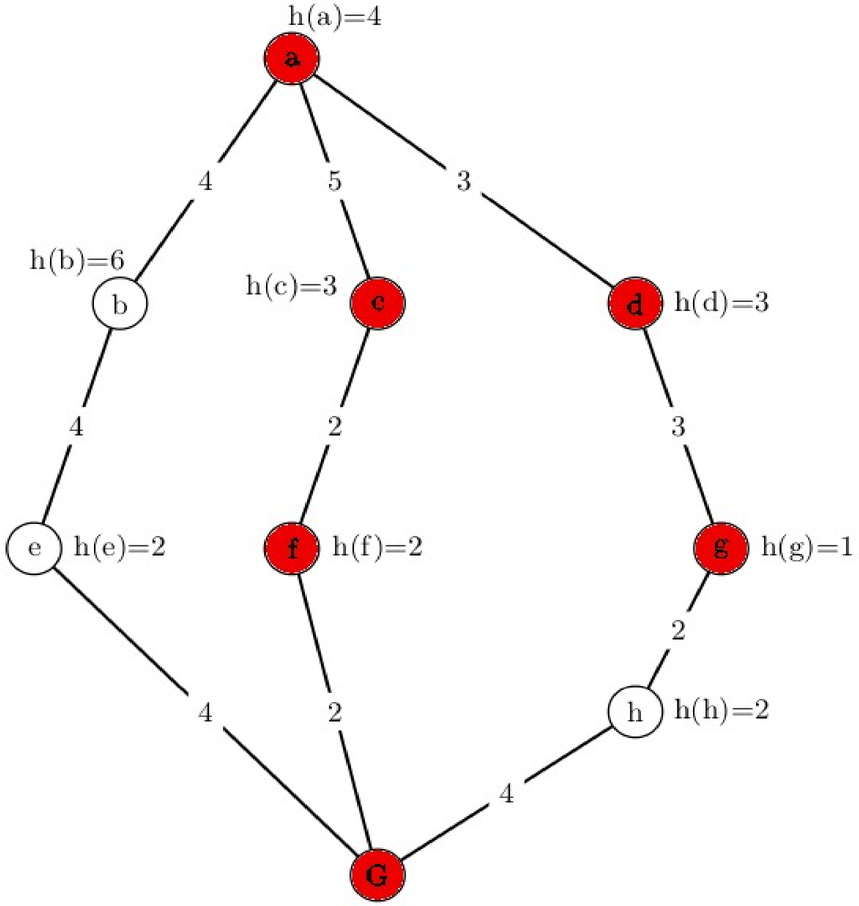
A*+h2



OPEN

f(b)=4+6
f(h)=8+2
f(G)=9+0

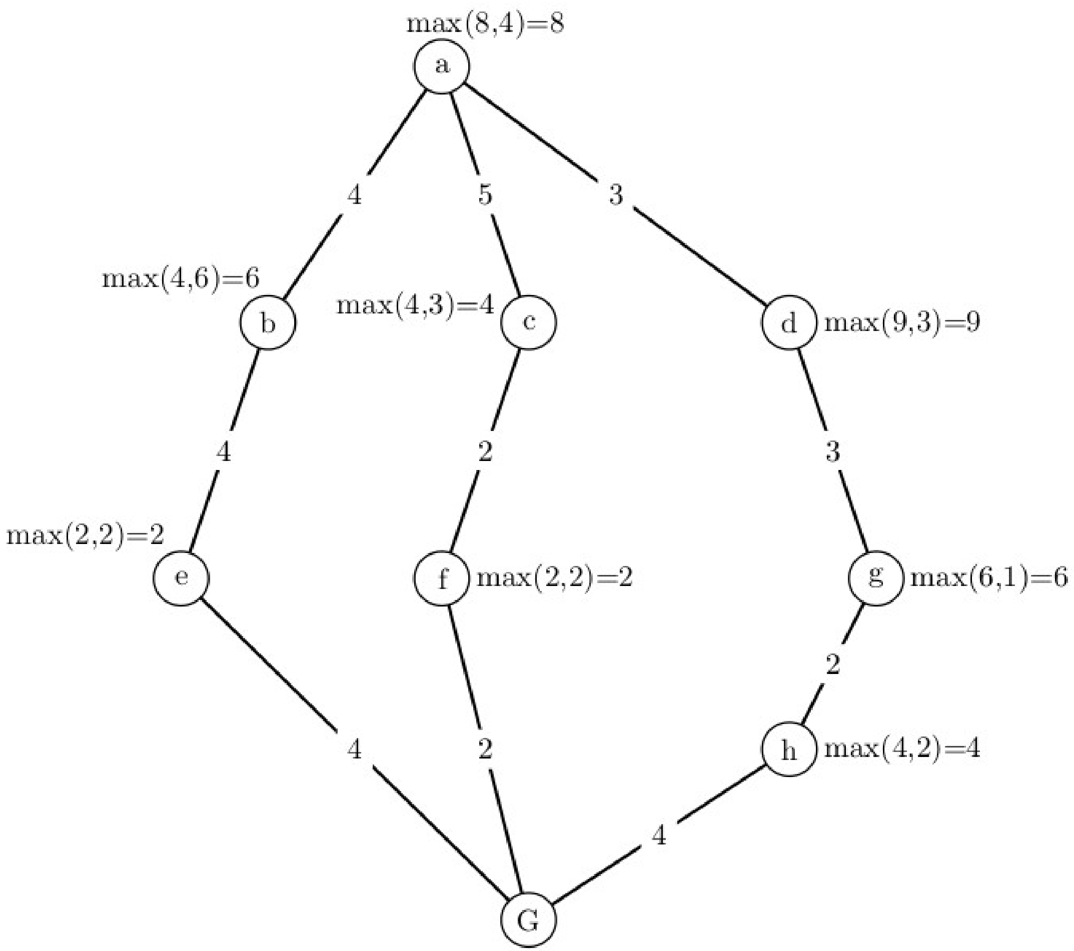
A*+h2



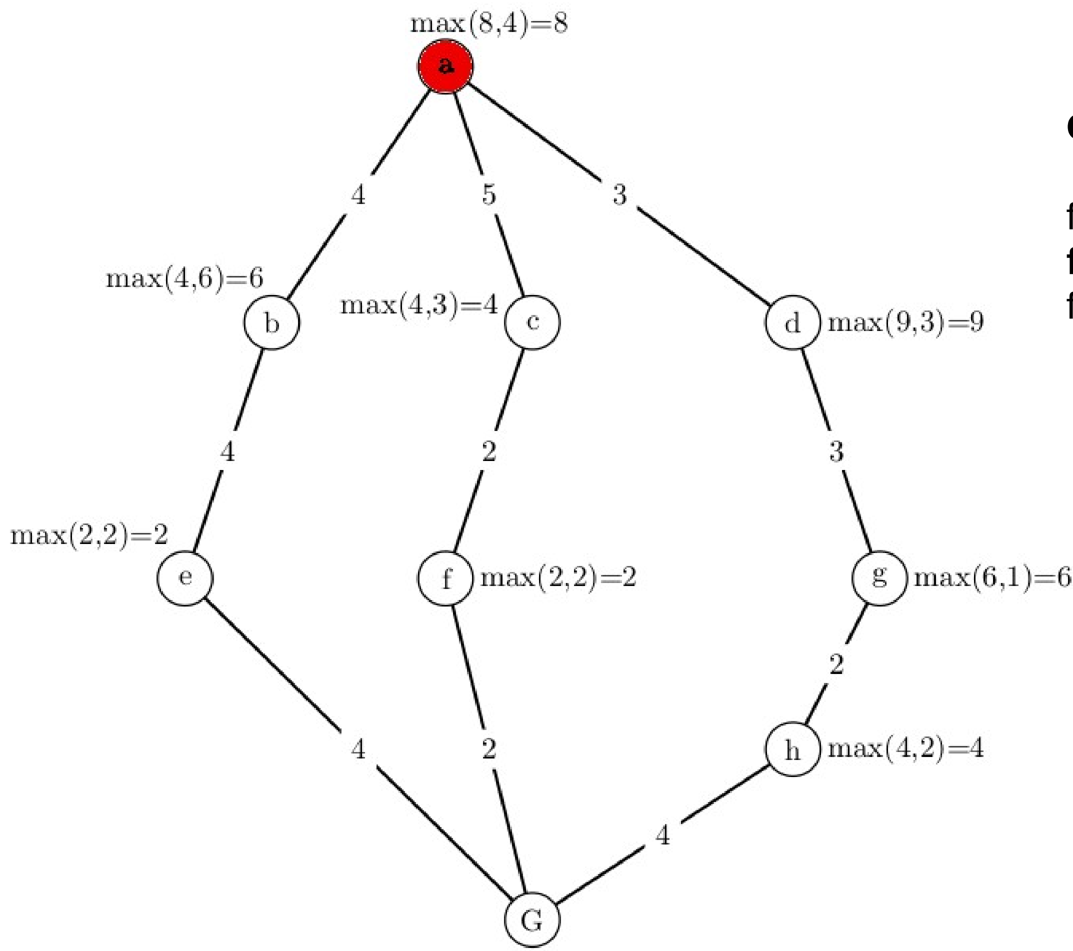
OPEN

f(b)=4+6
f(h)=8+2

A*+max(h1,h2)


max(8,4)=8
a
4
5
3
max(4,6)=6
b
max(4,3)=4
c
d
max(9,3)=9
4
2
3
max(2,2)=2
e
f
max(2,2)=2
g
max(6,1)=6
2
h
max(4,2)=4
4
2
4
G

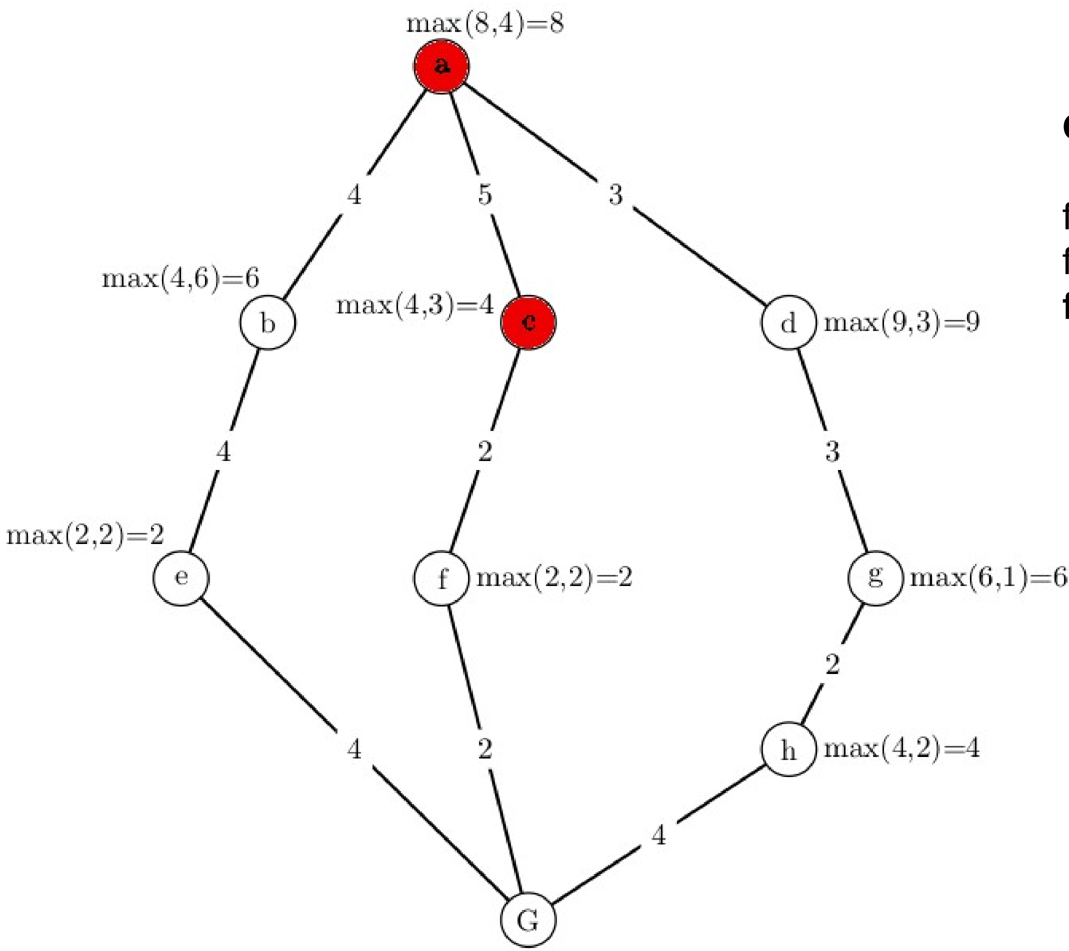
A*+max(h1,h2)



OPEN

f(b)=4+6
f(c)=5+4
f(d)=3+9

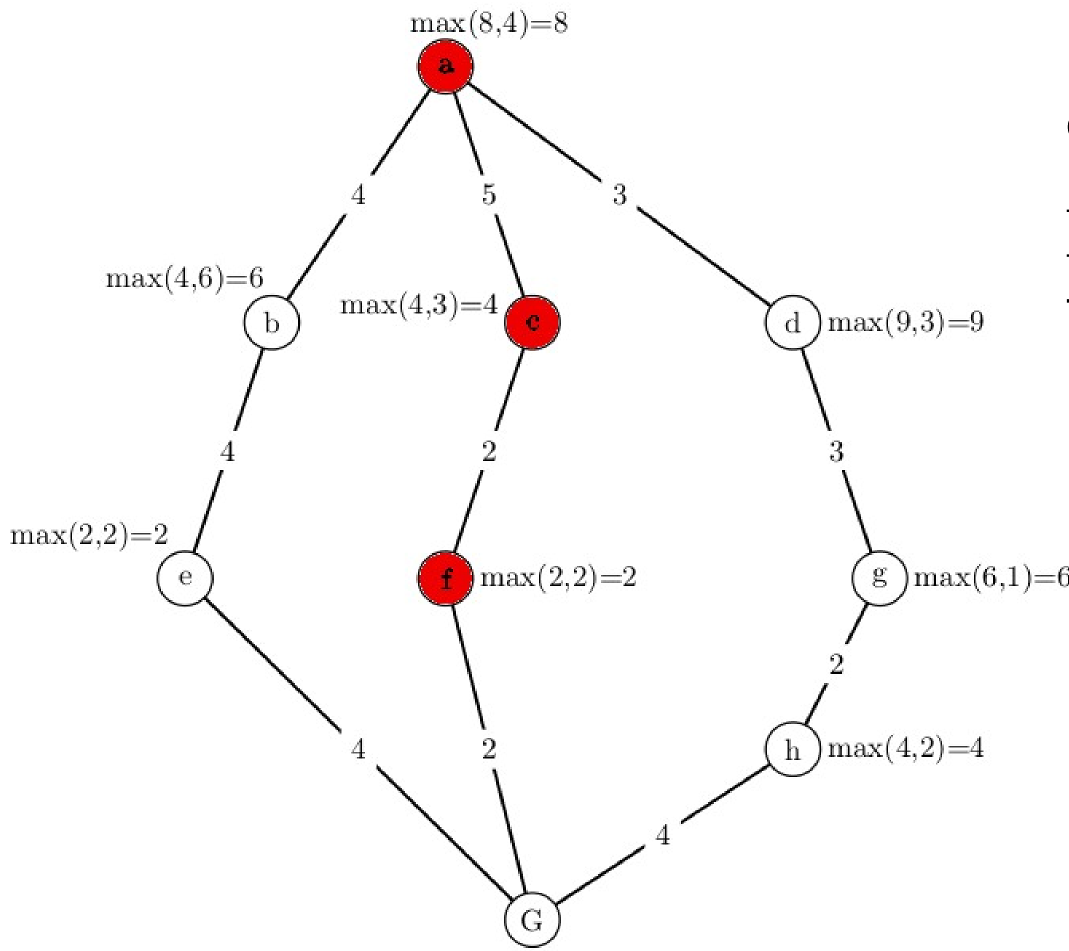
A*+max(h1,h2)



OPEN

f(b)=4+6
f(d)=3+9
f(f)=7+2

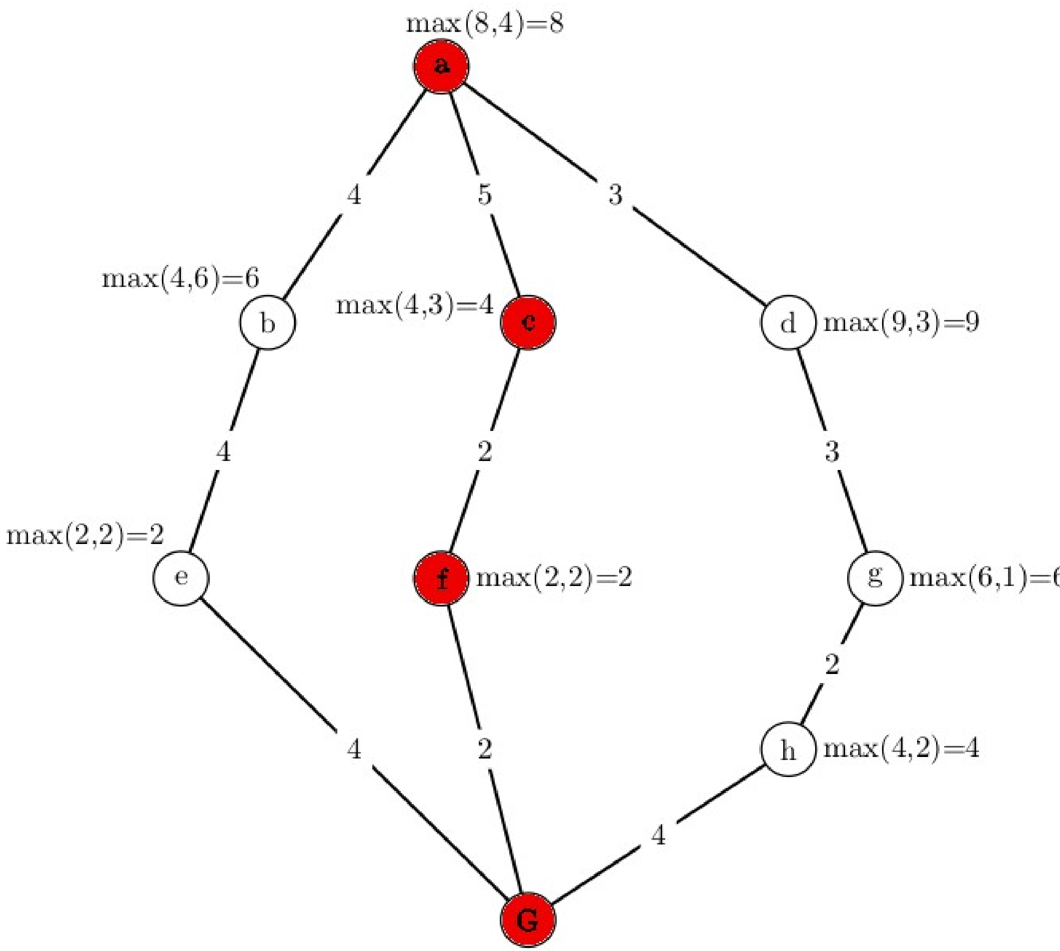
A*+max(h1,h2)



OPEN

f(b)=4+6
f(d)=3+9
f(G)=9+0

A*+max(h1,h2)



OPEN

f(b)=4+6
f(d)=3+9

Greedy Heuristic Selection(GHS)

Greedy Heuristic Selection(GHS)

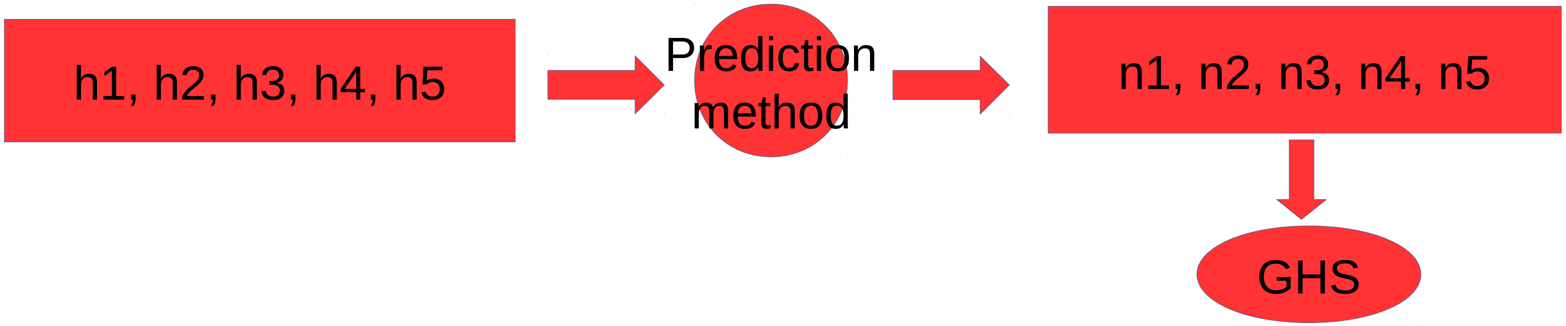
h1, h2, h3, h4, h5

Greedy Heuristic Selection(GHS)

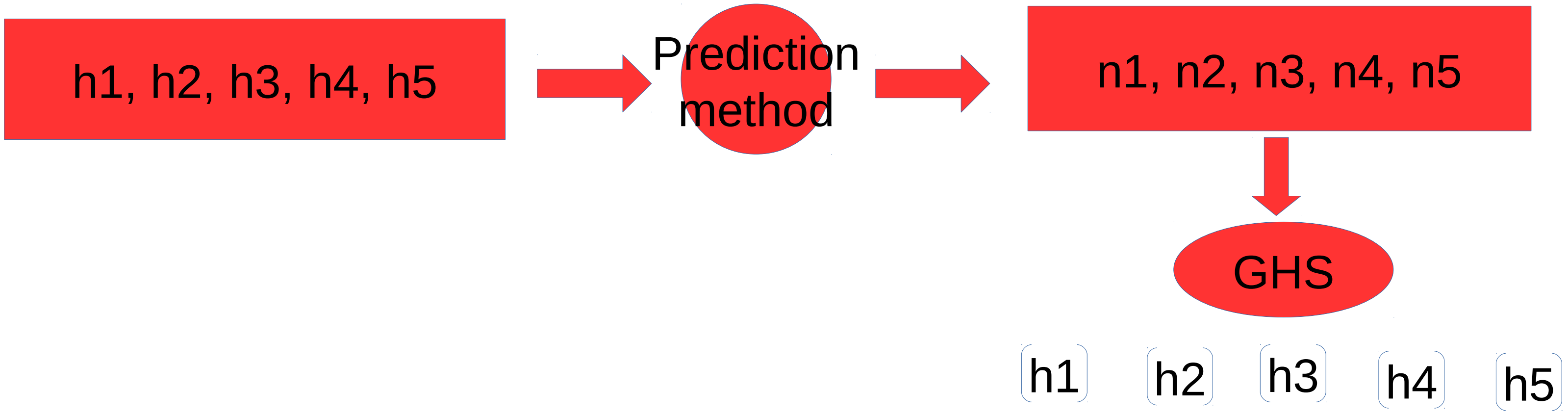
h1, h2, h3, h4, h5 → Prediction method

Greedy Heuristic Selection(GHS)

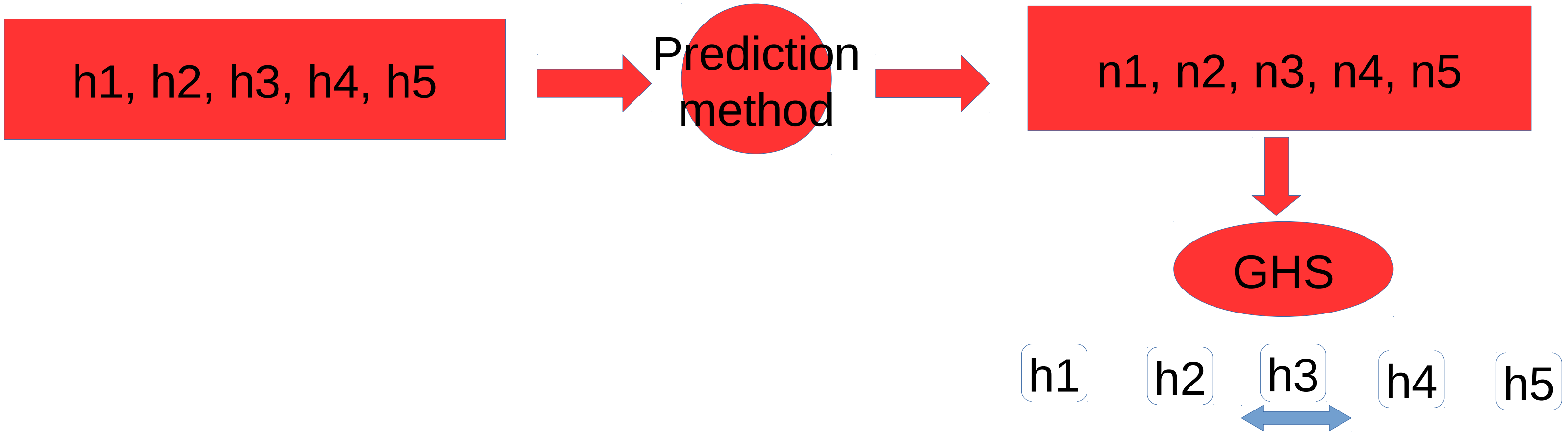
Greedy Heuristic Selection(GHS)



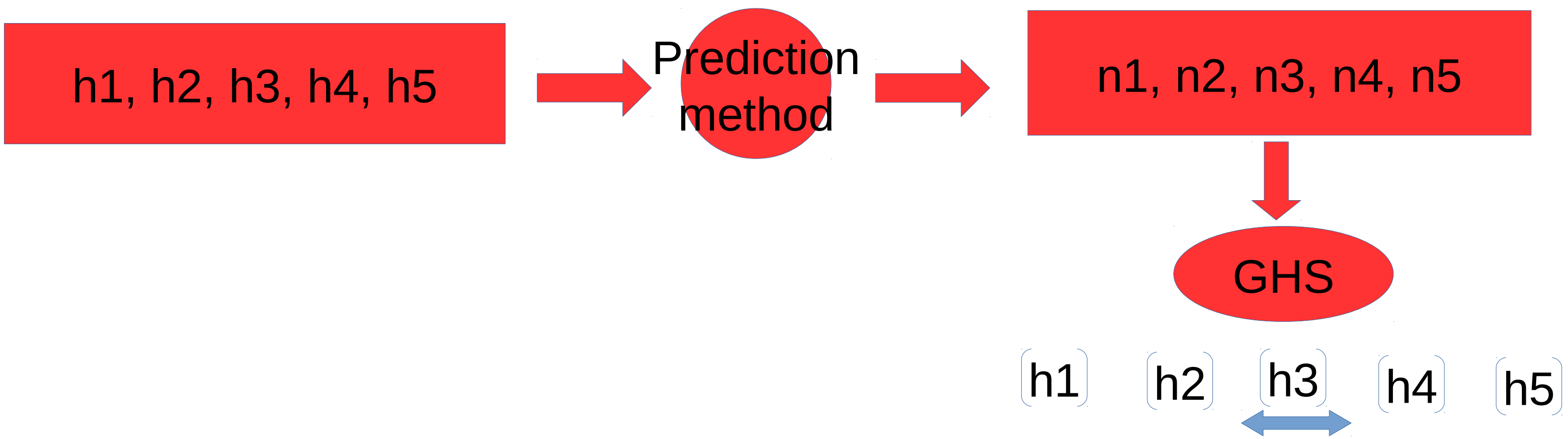
Greedy Heuristic Selection(GHS)



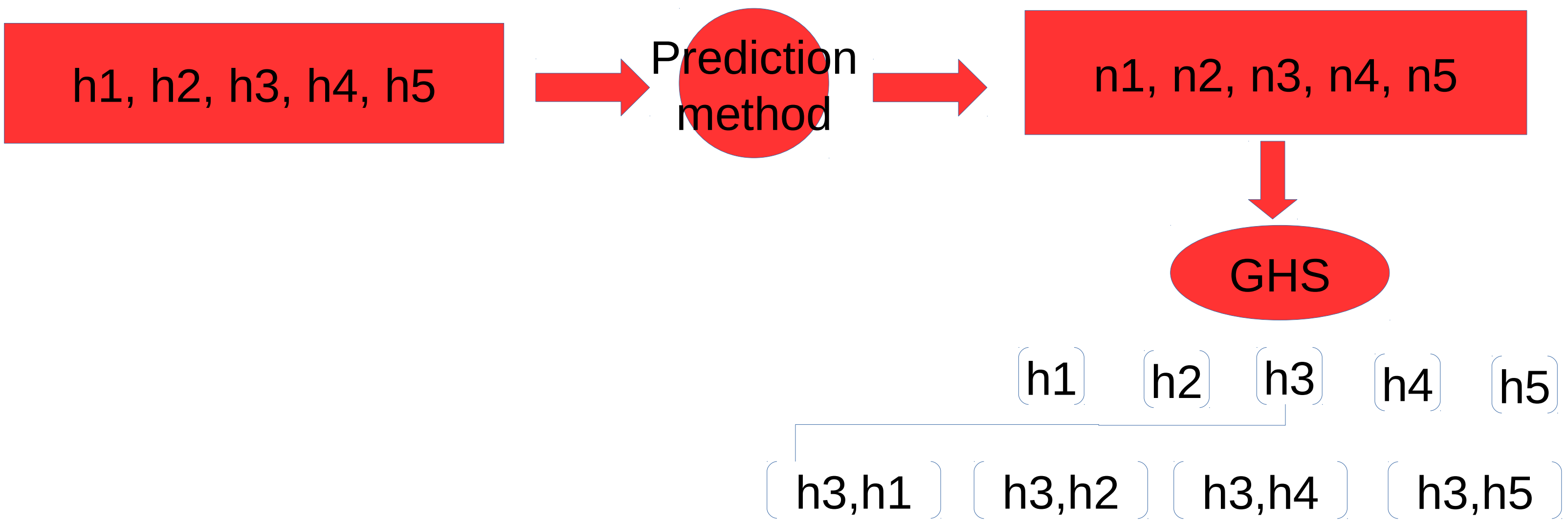
Greedy Heuristic Selection(GHS)



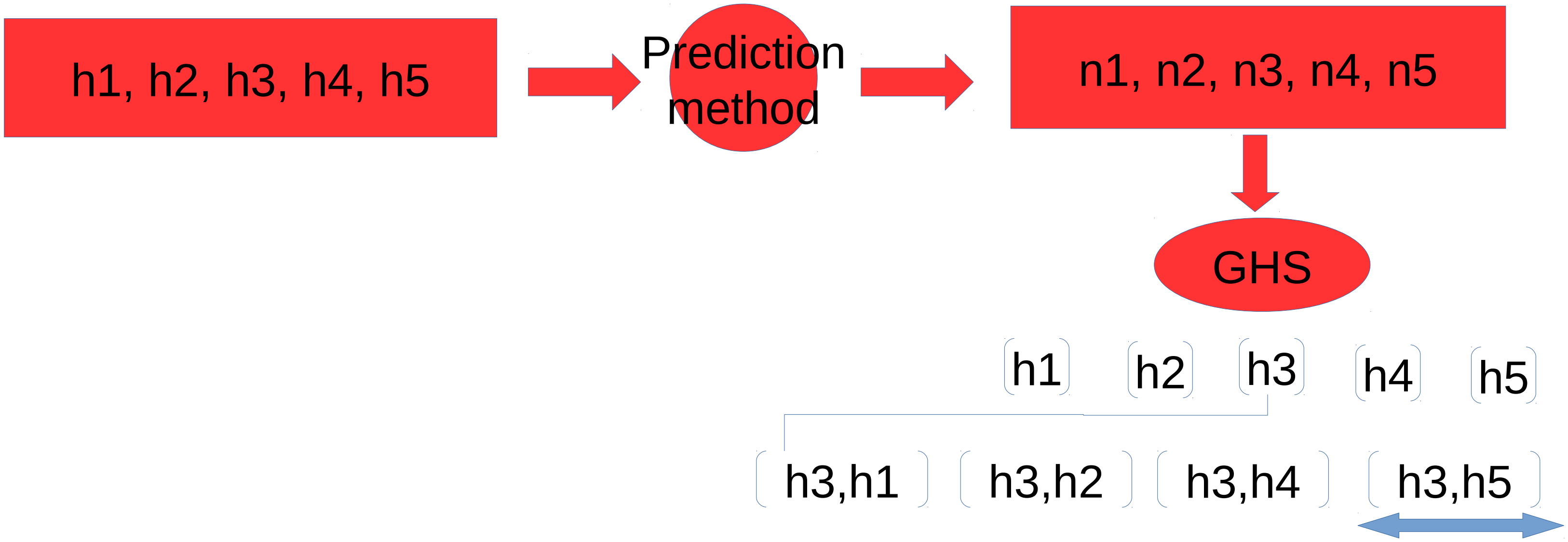
Greedy Heuristic Selection(GHS)



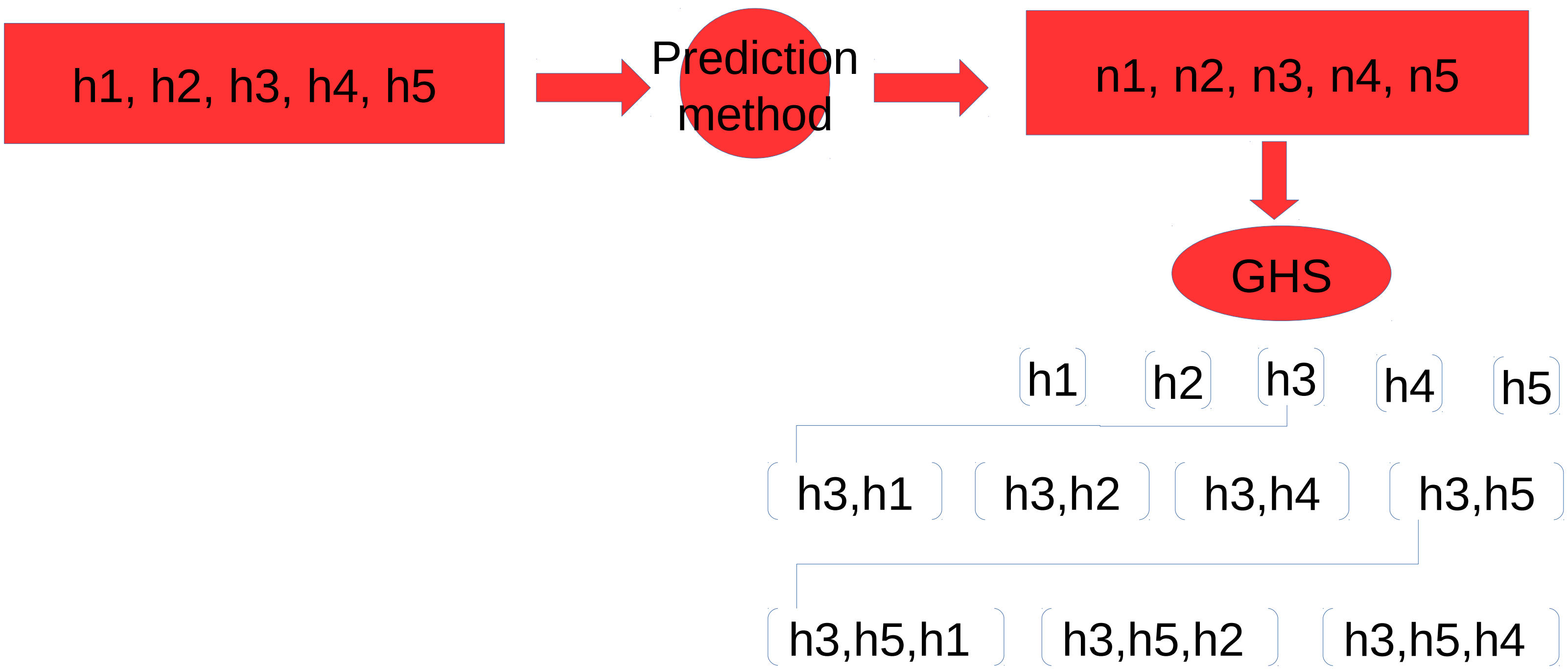
Greedy Heuristic Selection(GHS)



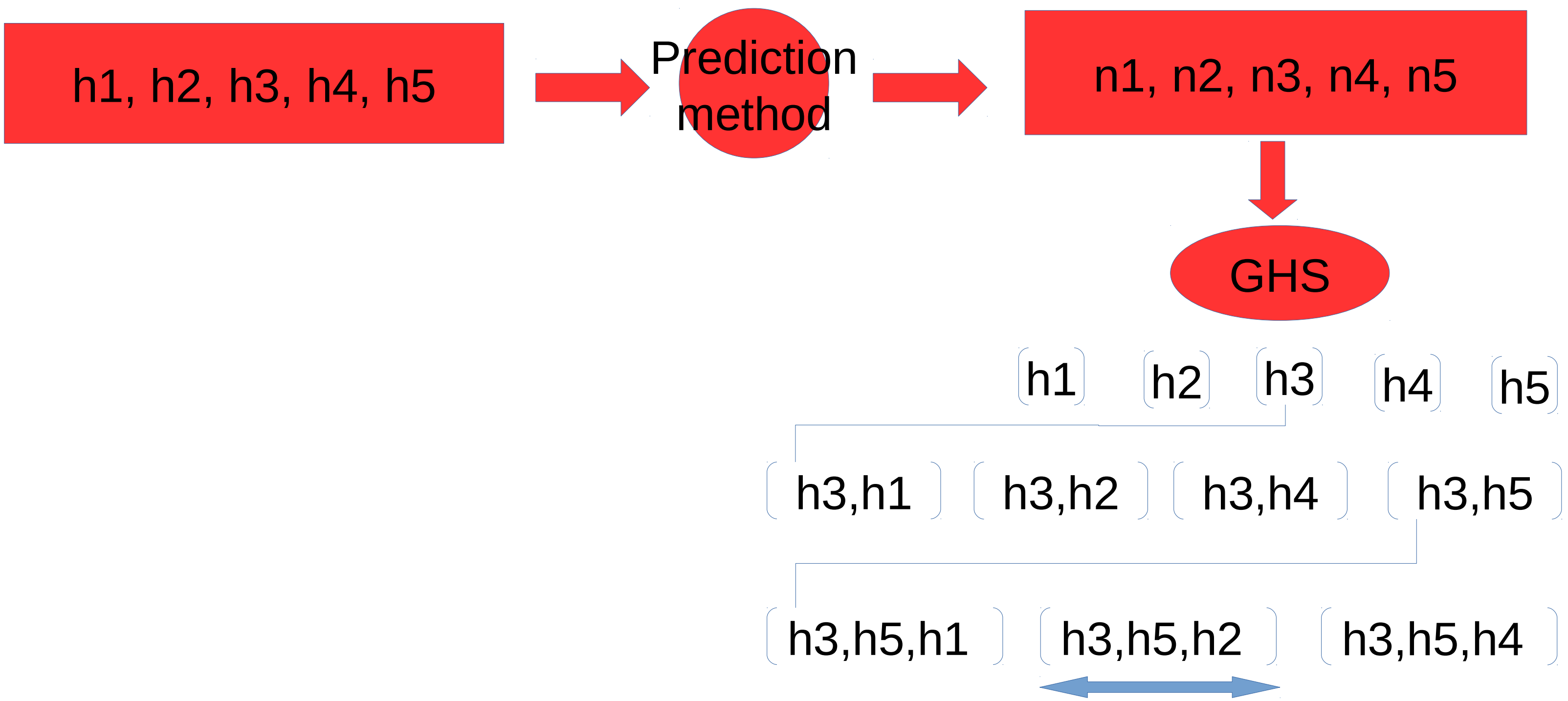
Greedy Heuristic Selection(GHS)



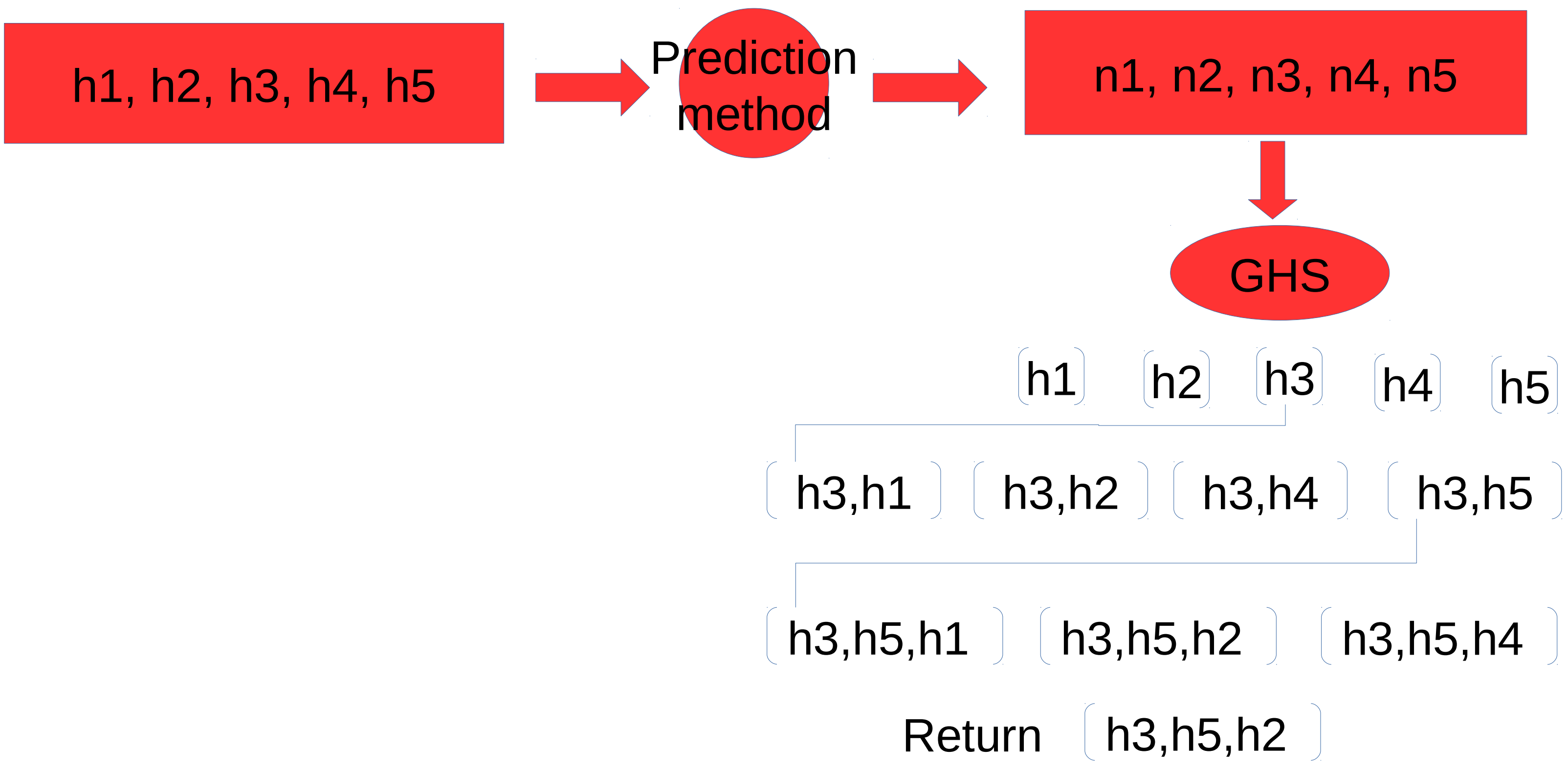
Greedy Heuristic Selection(GHS)



Greedy Heuristic Selection(GHS)



Greedy Heuristic Selection(GHS)



Greedy Heuristic Selection(GHS)

- **The problem formulation**
  - We present a greedy algorithm for approximately solving the following optimization problem,

$$\textbf{Minimize}_{\zeta' \in \zeta} \quad \Psi(\zeta', \nabla)$$

  - Ψ can be any objective function (J, T)
    - J is the approximation of the A* search tree size
    - T is the approximation of the A* running time
  - Run up to no more utility gain.

Greedy Heuristic Selection(GHS)

- **Ψ models the A* search tree size (J)**

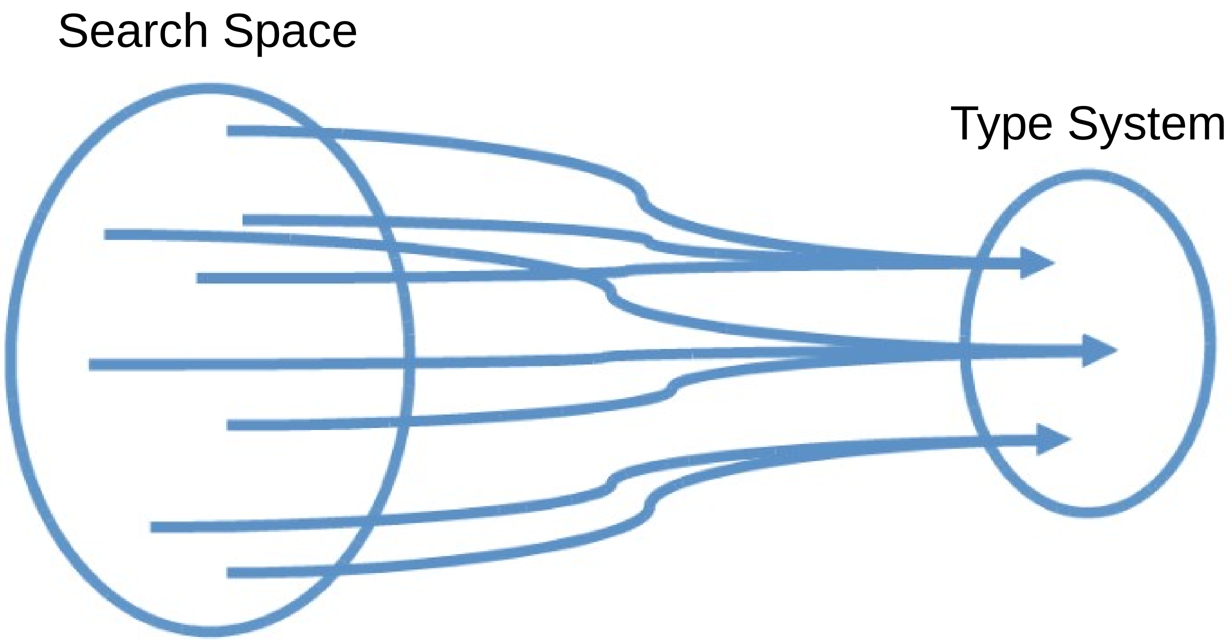
$$J(\zeta', \nabla) = |\{children(s) \in V \mid f_{max}(s, \zeta') \leq C^*\}|$$

Greedy Heuristic Selection(GHS)

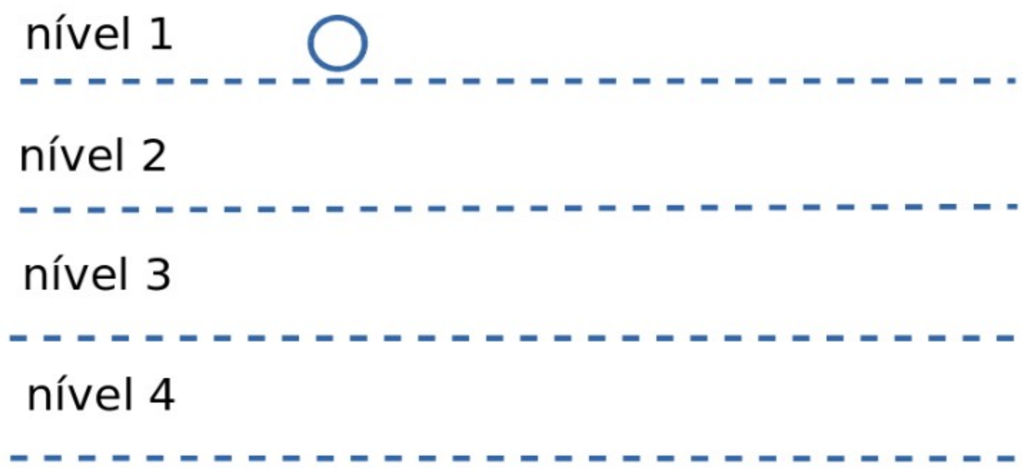
- **Ψ models the A* running time (T)**

$$T\ (\zeta', \nabla) = J(\zeta', \nabla) \cdot (t_{h_{max(\zeta')}} + t_{gen})$$

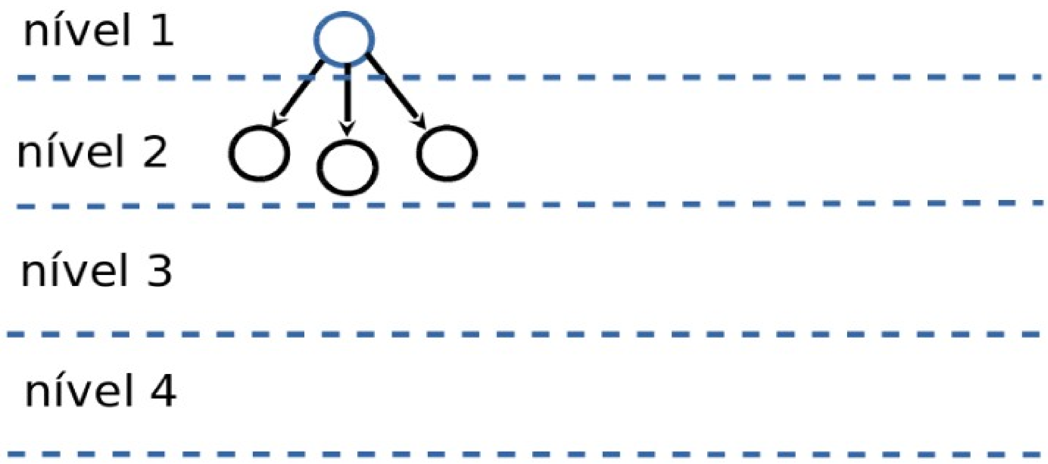
Type System



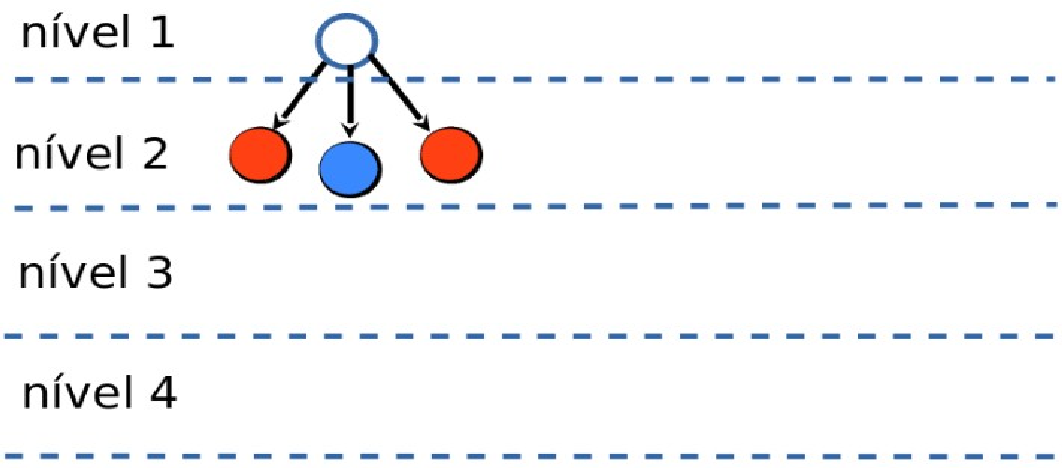
Stratified Sampling



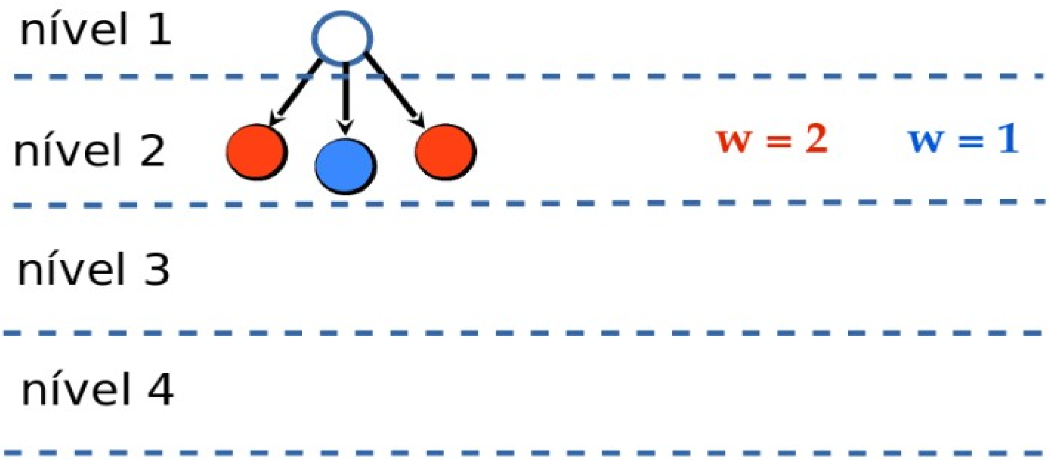
Stratified Sampling



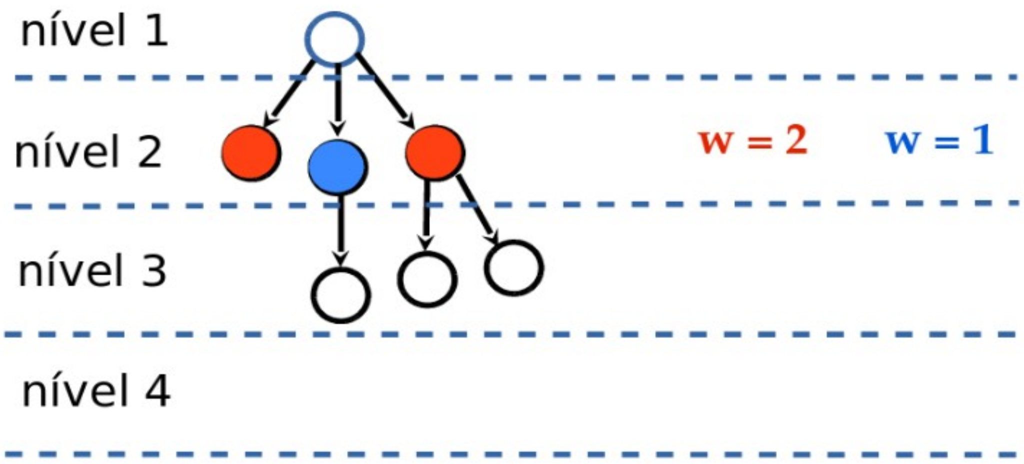
Stratified Sampling



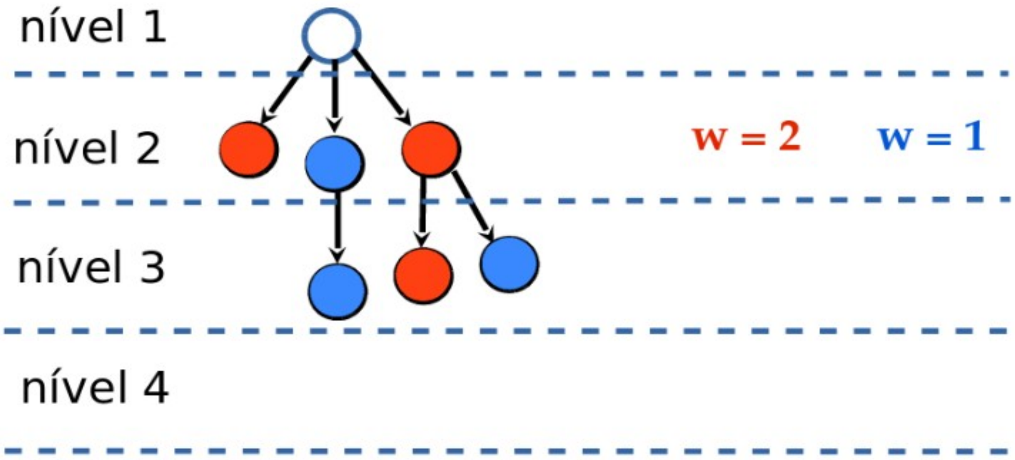
Stratified Sampling



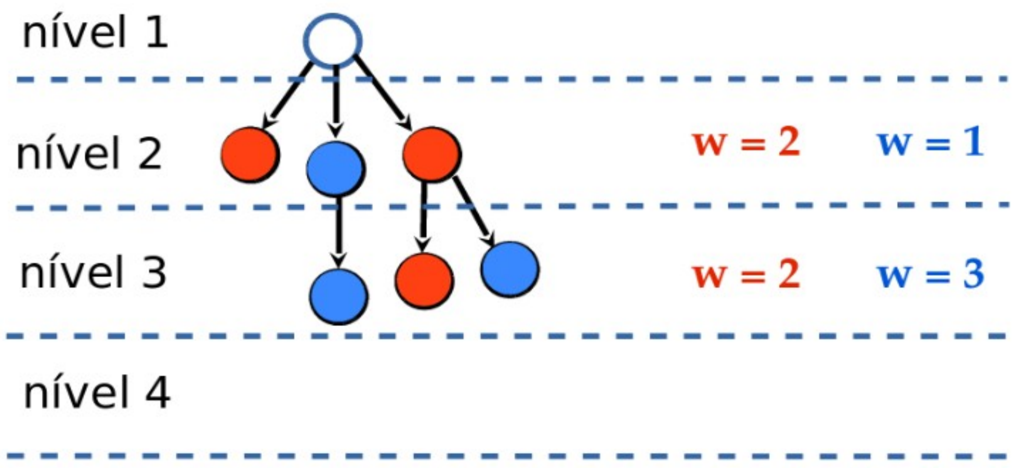
Stratified Sampling



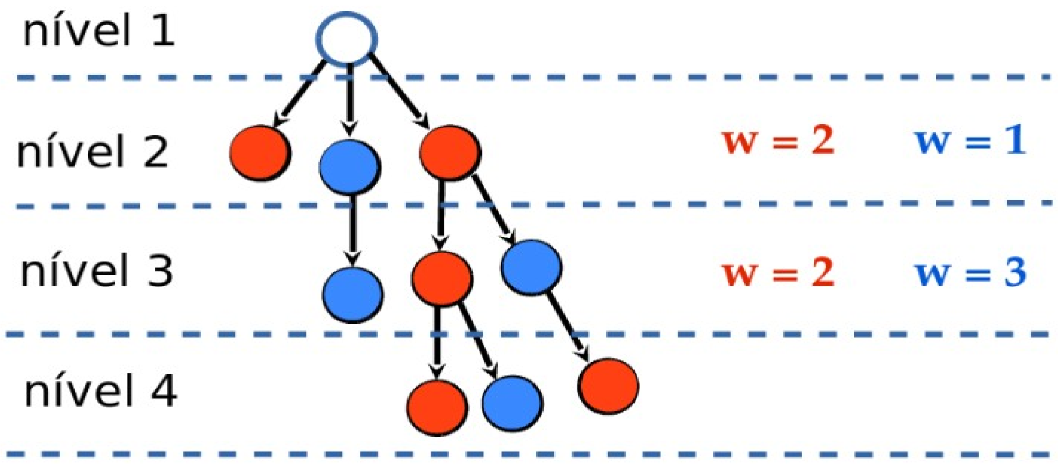
Stratified Sampling



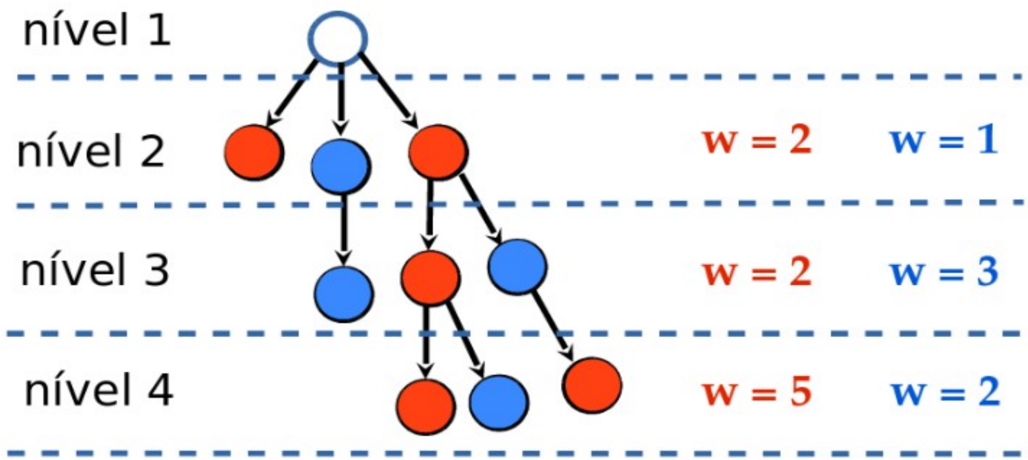
Stratified Sampling



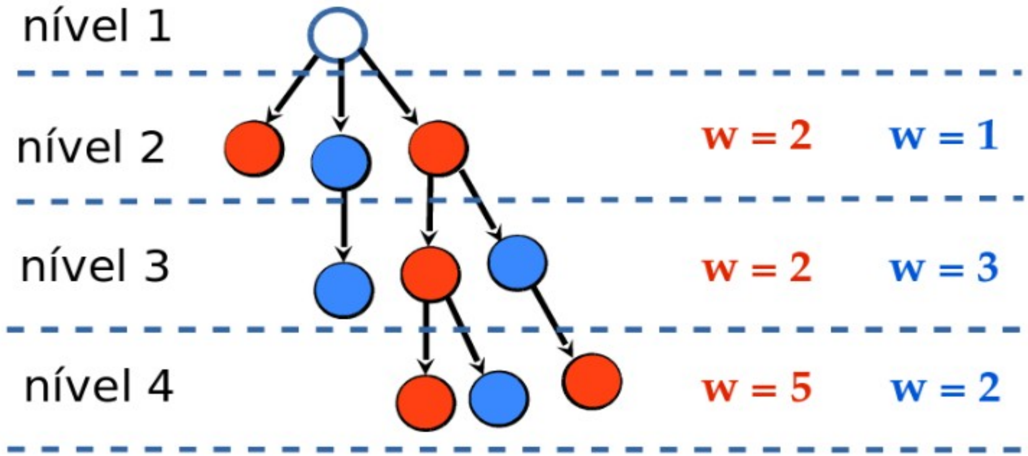
Stratified Sampling



Stratified Sampling



Stratified Sampling

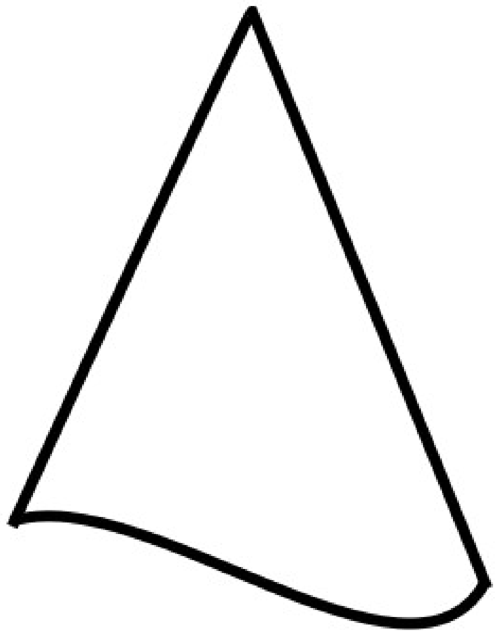


Prediction: 16 nodes

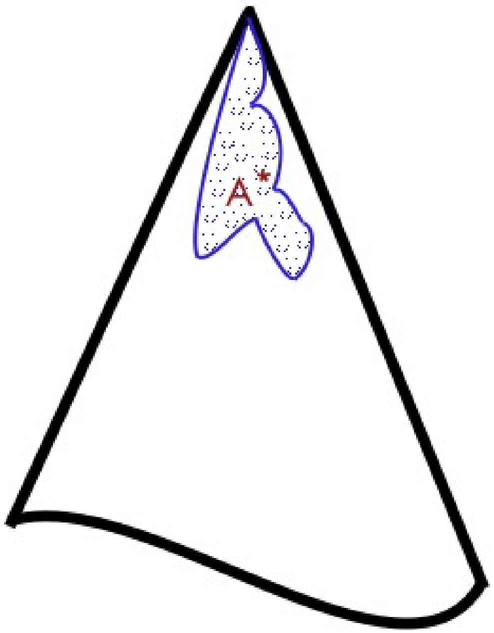
Culprit Sampling (Barley et al., 2014)

- $f_c(n) = \{f_1(n), f_2(n), \ldots, f_M(n)\}$, where $f_i(n) = g(n) + h_i(n)$ and set of heuristics $h_1, h_2, \ldots h_M$
- $bc(n) = \{y_1(n), y_2(n), \ldots y_M(n)\}$, where $y_i(n) = 1$ if $g(n) + h_i(n) \leq b$ and $y_i(n) = 0$, otherwise.

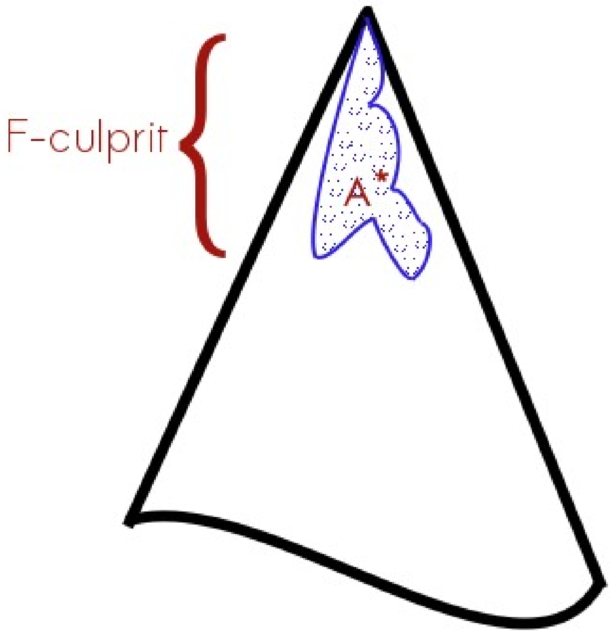
Culprit Sampling (Barley et al., 2014)



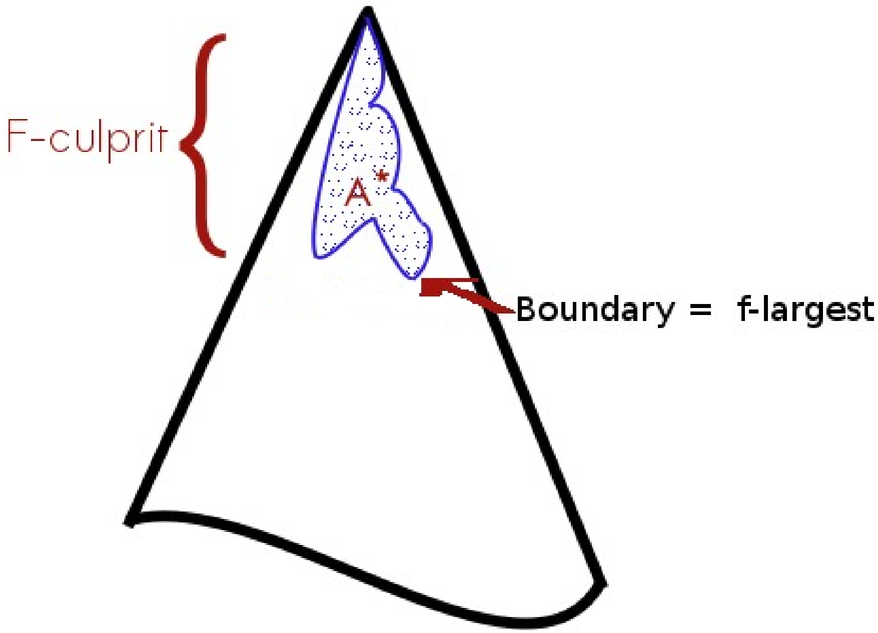
Culprit Sampling (Barley et al., 2014)



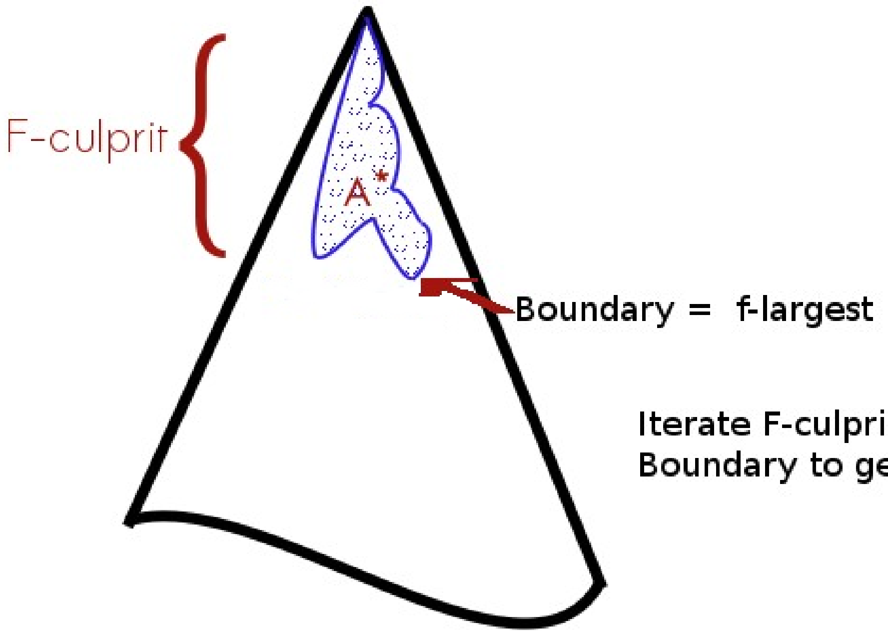
Culprit Sampling (Barley et al., 2014)



Culprit Sampling (Barley et al., 2014)

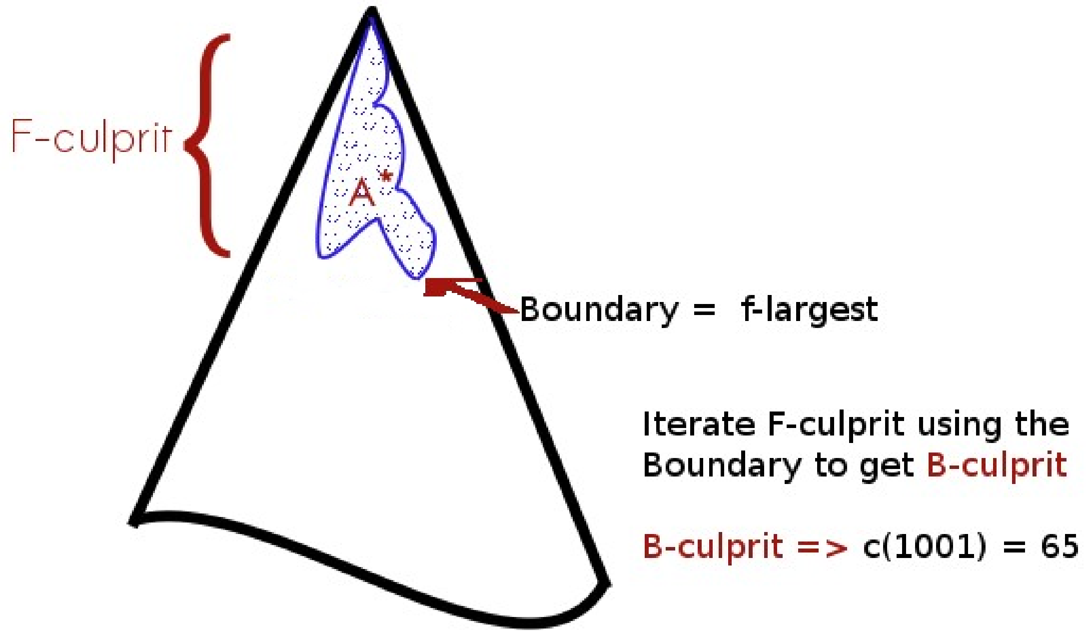


Culprit Sampling (Barley et al., 2014)



Iterate F-culprit using the Boundary to get B-culprit

Culprit Sampling (Barley et al., 2014)



Experiments

- Relative error

$$\frac{\sum_{s\in PI}\frac{Pred(s,d)-R(s,d)}{R(s,d)}}{|PI|}$$

Experiments

- SS for Predicting the IDA* Search Tree Size

Domain	hmax							
	IDA*	time	Relative-error					n
			1	10	100	1000	5000	
Barman	8835990.00	6016.38	0.60	0.45	0.20	0.07	0.04	20
Elevators	1012570.00	4987.57	0.84	0.42	0.23	0.13	0.10	2
Floortile	30522300.00	3919.72	2.02	0.62	0.40	0.14	0.11	20
Nomystery	6565740.00	3256.86	0.53	0.26	0.07	0.03	0.01	20
Openstacks	80108.50	4017.19	0.03	0.03	0.03	0.03	0.03	20
Parcprinter	1.00	0.00	0.00	0.00	0.00	0.00	0.00	20
Parking	374925.00	5607.50	0.17	0.04	0.01	0.00	0.00	20
Pegsol	68763.70	5.00	0.17	0.04	0.02	0.01	0.00	20
Scanalyzer	8449890.00	4920.58	0.43	0.25	18.63	0.02	0.01	20
Sokoban	3118530.00	3932.69	0.41	0.26	0.11	0.05	0.04	20
Tidybot	444473.00	5632.08	300.86	1072.40	5.88	0.01	0.01	20
Transport	2622880.00	2253.51	0.63	0.54	0.24	0.15	0.11	20
Visitall	71032400.00	3704.78	0.12	0.04	0.04	0.00	0.00	20
Woodworking	5139070.00	4944.76	1.28	0.69	0.69	0.17	0.07	20

Experiments

- SS for Predicting the IDA* Search Tree Size

Domain	hmax							
	IDA*	time	time					n
			1	10	100	1000	5000	
Barman	8835990.00	6016.38	0.06	0.32	3.21	32.57	214.59	20
Elevators	1012570.00	4987.57	1.40	9.85	96.37	994.33	4425.93	2
Floortile	30522300.00	3919.72	0.01	0.07	0.69	6.93	36.60	20
Nomystery	6565740.00	3256.86	0.07	0.38	3.63	36.35	181.03	20
Openstacks	80108.50	4017.19	94.79	774.86	1067.84	10929.00	11174.30	20
Parcprinter	1.00	0.00	0.01	0.04	0.35	3.48	17.29	20
Parking	374925.00	5607.50	1.79	11.36	114.28	1196.83	5835.03	20
Pegsol	68763.70	5.00	0.01	0.04	0.37	3.69	17.88	20
Scanalyzer	8449890.00	4920.58	3.13	28.79	273.74	3033.06	10254.00	20
Sokoban	3118530.00	3932.69	0.31	2.00	21.42	222.47	1056.61	20
Tidybot	444473.00	5632.08	4.40	26.48	238.76	2747.10	11925.40	20
Transport	2622880.00	2253.51	0.09	0.61	5.89	59.37	290.31	20
Visitall	71032400.00	3704.78	0.00	0.05	0.56	5.77	28.07	20
Woodworking	5139070.00	4944.76	0.15	1.33	13.21	130.82	664.08	20

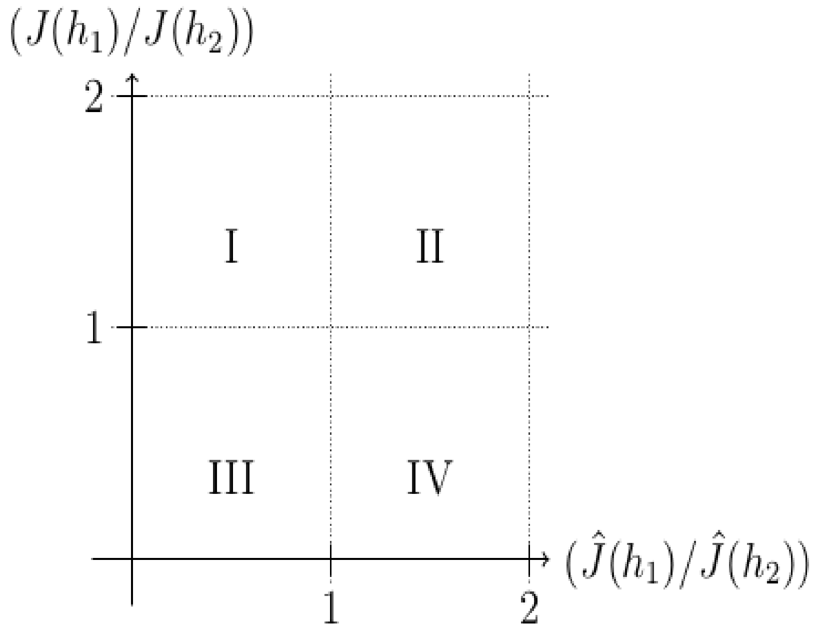
Experiments

- SS for Predicting the A* Search Tree size

	ipdb		LM-Cut		M&S		n
Domain	A*	Relative-error	A*	Relative-error	A*	Relative-error	
Barman	1.72×10^{7}	8.68×10^{31}	7.45×10^{6}	2.21×10^{30}	6.67×10^{6}	1.26×10^{36}	4
Floortile	1.40×10^{7}	1.74×10^{18}	702435	4.68×10^{14}	4.46×10^{6}	1.90×10^{12}	4
Nomystery	40169.7	6.71×10^{32}	267100	6.14×10^{19}	8236	1.20×10^{20}	9
Openstacks	570099	0.61884	570099	0.677425	569984	0.672143	4
Parcprinter	1157	2.56×10^{22}	1363.67	2.33×10^{21}	766.333	6.36×10^{20}	3
Pegsol	841693	2901.39	398221	6859.86	933430	779.017	16
Scanalyzer	337894	3.94×10^{33}	334747	7.58×10^{31}	337833	2.42×10^{31}	3
Sokoban	376755	1.04×10^{7}	45374	2.74×10^{6}	739775	5.60×10^{8}	9
Transport	1.89×10^{6}	2.91×10^{38}	1.49×10^{6}	1.15×10^{25}	1.73×10^{6}	1.50×10^{29}	2
Visitall	253710	1.69×10^{46}	253195	1.69×10^{46}	253521	1.71×10^{46}	8
Woodworking	3.21×10^{6}	2.53×10^{18}	3.20×10^{6}	2.76×10^{18}	3.21×10^{6}	2.48×10^{18}	3

Experiments

- Approximation analysis for SS and A*



Experiments

- Approximation analysis for SS and A*

Domain	II and III (%)
Elevators	78.57
Floortile	96.08
Parking	71.82
Parcprinter	70.50
Pegsol	96.83
Scanalyzer	100.00
Sokoban	89.31
Tidybot	100.00
Transport	51.78
Visitall	98.05
Woodworking	100.00

Experiments

- Ratios of the number of nodes expanded using $h_{max}(\zeta')$ to the number of nodes expanded using $h_{max}(\zeta)$

	SS		CS		$\|\zeta\|$
Domain	Ratio	$\|\zeta'\|$	Ratio	$\|\zeta'\|$	
Barman	1.11	17.70	1.50	30.25	5168.50
Elevators	11.50	2.00	1.03	21.00	168.00
Floortile	1.02	43.07	1.01	42.35	151.28
Openstacks	1.00	1.00	1.00	1.00	390.69
Parking	1.00	5.53	1.01	7.26	21.73
Parcprinter	3.62	1.00	2.21	13.00	1189.00
Pegsol	1.00	31.00	1.00	57.00	90.00
Scanalyzer	1.23	30.57	1.57	19.43	72.86
Sokoban	1.32	7.00	1.01	24.00	341.00
Tidybot	1.00	2.35	1.00	8.59	3400.18
Transport	1.00	14.70	1.03	14.30	171.17
Visitall	1.03	99.33	1.19	48.67	256.33
Woodworking	32.42	3.00	199.65	5.00	1289.00

Experiments

- Coverage of different planning systems on the 2011 IPC benchmarks.

Domains	HYBRID	CS		SS		Sum	Max	RIDA*	SY1	SY2	StSp1	StSp2
		Time	Size	Time	Size							
Barman	7	5	4	4	4	4	4	4	10	11	4	4
Elevators	19	19	19	19	19	19	19	19	20	20	18	18
Floortile	15	14	14	14	14	14	14	14	14	14	14	14
Nomystery	20	20	20	19	19	20	20	20	16	16	20	20
Openstacks	17	17	15	17	15	15	11	15	20	20	17	17
Parcprinter	18	18	18	16	15	19	18	18	17	17	18	18
Parking	7	7	2	7	2	2	2	7	2	1	5	5
Pegsol	18	18	19	19	19	19	19	19	19	20	19	19
Scanalyzer	13	14	12	11	14	14	14	14	9	9	14	14
Sokoban	20	20	20	20	20	20	20	20	20	20	20	20
Tidybot	17	16	16	16	16	16	15	17	15	17	16	16
Transport	14	13	10	11	13	11	9	10	10	11	7	8
Visitall	18	18	18	15	18	18	18	18	12	12	16	16
Woodworking	16	15	15	12	16	16	16	15	20	20	15	15
Total	219	214	202	200	204	207	199	210	204	208	203	204

Conclusions

- GHS selects good subset of heuristics from ζ with respect to the A* search tree size and A* running time.
- We also experimented with an objective function that accounts for the sum of heuristic values in the state-space (Rayner et al., 2013)
- We tested two prediction algorithms, CS and SS
- SS is helpful for our utility function in the greedy heuristic selection for some domain/instances.
- GHS substantially outperforms other systems designed for using multiple heuristic functions.

References

Michael W. Barley, Santiago Franco, and Patricia J. Riddle. Overcoming the utility problem in heuristic generation: Why time matters. Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014, Portsmouth, New Hampshire, USA, June 21-26, 2014, 2014.

Haslum, P.; Botea, A.; Helmert, M.; Bonet, B.; and Koenig, S. 2007. Domain-independent construction of pattern database heuristics for cost-optimal planning. In AAAI.

Haslum, P.; Bonet, B.; and Geffner, H. 2005. New admissible heuristics for domain-independent planning. In AAAI

Edelkamp, S. 2007. Automated creation of pattern database search heuristics. In Model Checking and Artificial Intelligence, volume 4428 of LNCS. 35–50

L. H. S. Lelis, S. Zilles, and R. C. Holte. Predicting the Size of IDA*'s Search Tree. Artificial Intelligence, pages 53–76, 2013.

Christer B¨ackstr¨om and Bernhard Nebel. Complexity results for sas+ planning. Computational Intelligence, 11(4):625–655, 1995.

References

Haifeng Xu, Fei Fang, Albert Xin Jiang, Vincent Conitzer, Shaddin Dughmi, and Milind Tambe. Solving zero-sum security games in discretized spatio-temporal domains. Proceedings of the 28th Conference on Artificial Intelligence (AAAI 2014), Qubec, Canada, 2014.

Andreas Krause and Daniel Golovin. Submodular function maximization. Tractability: Practical
Approaches to Hard Problems, 3:19, 2012.

Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. AAAI, 7:1650–1654, 2007.

Knuth, D. E. 1975. Estimating the efficiency of backtrack programs. Math. Comp. 29:121–136.

Chen, P.-C. 1992. Heuristic sampling: A method for predicting the performance of tree searching programs. SIAM Journal on Computing 21:295–315.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maxi-mizing submodular set functions. Mathematical Programming, 14(1):265–294, 1978.

Thanks

Questions