

Cutting the Size of Compressed Path Databases with Wildcards and Redundant Symbols

Mattia Chiari, Shizhe Zhao, Adi Botea, Alfonso E. Gerevini, Daniel Harabor, Alessandro Saetti, Matteo Salvetti, Peter J. Stuckey

ICAPS 2019

Outline

Introduction

Redundant Symbols

Proximity Wildcards

Experimental Results

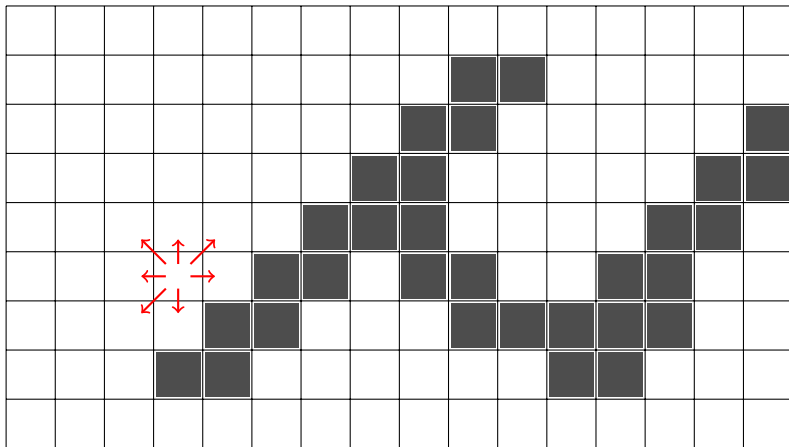
Conclusion

The shortest path problem

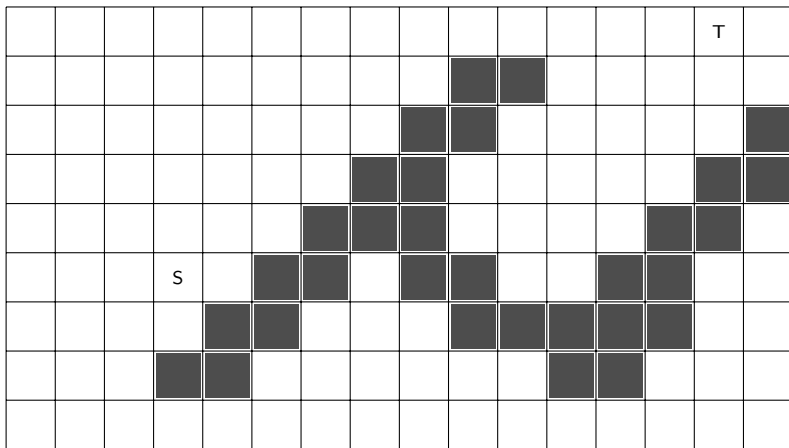
Shortest paths in a graph have many applications

- Robot navigation
- Games
 - Character navigation on a game map
 - Pathfinding used to be the only AI present in commercial video games

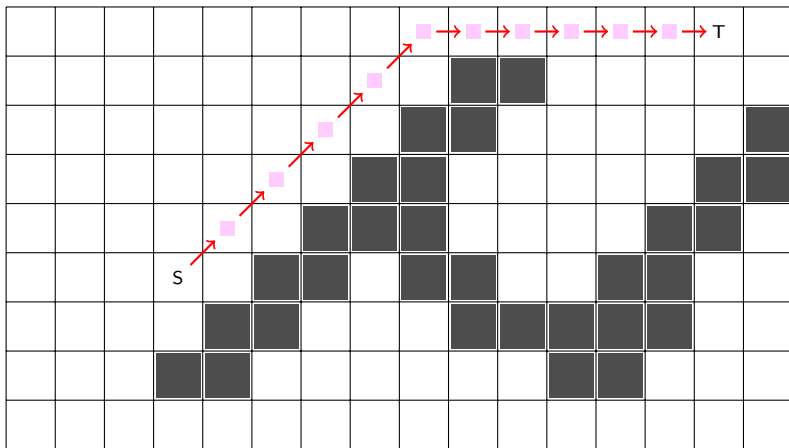
Example



Example



Example



The Take-Home Message

Our contribution

Significant size reduction for Compressed Path Databases (CPDs)

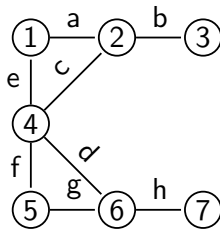
CPD [Botea, 2011; Strasser *et al.*, 2015]

- Encoding optimal moves on a map/graph
- Useful e.g., in optimal pathfinding on gridmaps
- State-of-the-art speed
 - Top performer in Grid-based Path Planning Competitions, 2012 and 2014
 - Subsequent improvements [Salvetti *et al.*, 2017; Salvetti *et al.*, 2018]
 - Moving target search [Botea *et al.*, 2013; Baier *et al.*, 2014; Xie *et al.*, 2017]
- Preprocessing time and memory required

What Is a Compressed Path Database (CPD)?

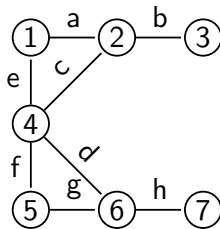
What Is a Compressed Path Database (CPD)?

- Given a graph...



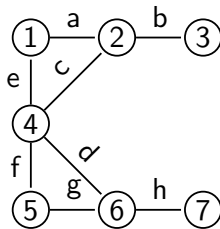
What Is a Compressed Path Database (CPD)?

- Given a graph...
- and *any* two nodes s and t ...



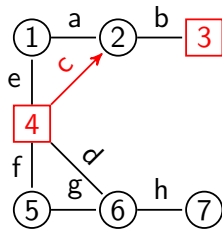
What Is a Compressed Path Database (CPD)?

- Given a graph...
- and *any* two nodes s and t ...
- A CPD provides the **first edge** of a shortest path from s to t



What Is a Compressed Path Database (CPD)?

- Given a graph...
- and *any* two nodes s and t ...
- A CPD provides the **first edge** of a shortest path from s to t
- E.g., $\text{CPD}[4, 3] = c$

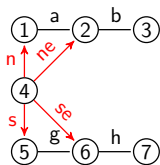


First-move matrix

1	2	3
4		
5	6	7

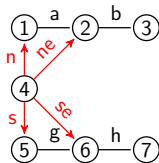
First-move matrix

1	2	3
4		
5	6	7



First-move matrix

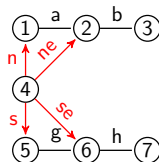
1	2	3
4		
5	6	7



s \ t	1	2	3	4	5	6	7
1							
2							
3							
4	n	ne	ne	*	s	se	se
5							
6							
7							

First-move matrix

1	2	3
4		
5	6	7



s \ t		t						
		1	2	3	4	5	6	7
1		*	e	e	s	s	s	s
2		w	*	e	sw	sw	sw	sw
3		w	w	*	w	w	w	w
4		n	ne	ne	*	s	se	se
5		n	n	n	n	*	e	e
6		nw	nw	nw	nw	w	*	e
7		w	w	w	w	w	w	*

Compressing first-matrix rows

- First-matrix rows are compressed with run-length encoding (RLE)
- *Runs* are repetitions of the same token
 - E.g., the string **ssssssnnnnnnssssssss** has three runs: **1/s**; **7/n**; **12/s**

Compressing first-matrix rows

- First-matrix rows are compressed with run-length encoding (RLE)
- *Runs* are repetitions of the same token
 - E.g., the string **ssssssnnnnnnssssssss** has three runs: **1/s; 7/n; 12/s**

s \ t	t							
		1	2	3	4	5	6	7
	1	*	e	e	s	s	s	s
	2	w	*	e	sw	sw	sw	sw
	3	w	w	*	w	w	w	w
	4	n	ne	ne	*	s	se	se
	5	n	n	n	n	*	e	e
	6	nw	nw	nw	nw	w	*	e
	7	w	w	w	w	w	w	*

Uncompressed first move matrix has 49 tokens

Compressing first-matrix rows

- First-matrix rows are compressed with run-length encoding (RLE)
- Runs* are repetitions of the same token
 - E.g., the string **ssssssnnnnnnssssssss** has three runs: **1/s; 7/n; 12/s**

s \ t	1	2	3	4	5	6	7
1	*	e	e	s	s	s	s
2	w	*	e	sw	sw	sw	sw
3	w	w	*	w	w	w	w
4	n	ne	ne	*	s	se	se
5	n	n	n	n	*	e	e
6	nw	nw	nw	nw	w	*	e
7	w	w	w	w	w	w	*



1	1/e	4/s		
2	1/w	3/e	4/sw	
3	1/w			
4	1/n	2/ne	5/s	6/se
5	1/n	6/e		
6	1/nw	5/w	7/e	
7	1/w			

CPD has 16 runs

Uncompressed first move matrix has 49 tokens

Outline

Introduction

Redundant Symbols

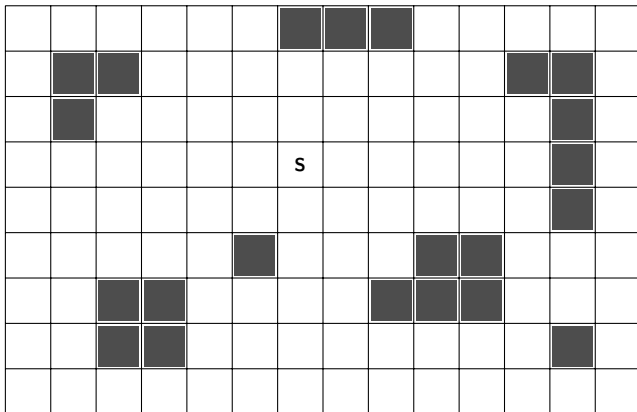
Proximity Wildcards

Experimental Results

Conclusion

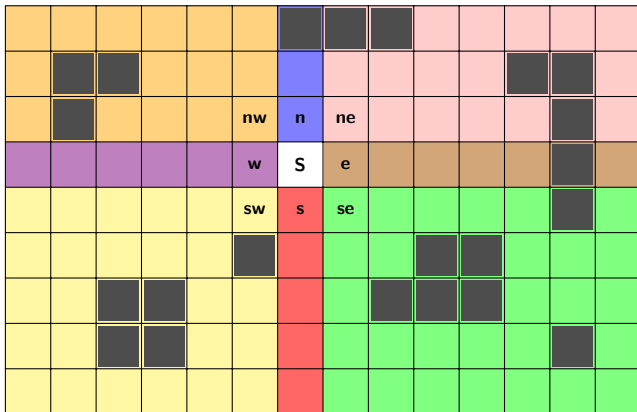
Default Moves

- Heuristic moves (e.g., based on the Octile heuristic)
- Easy to compute in constant time

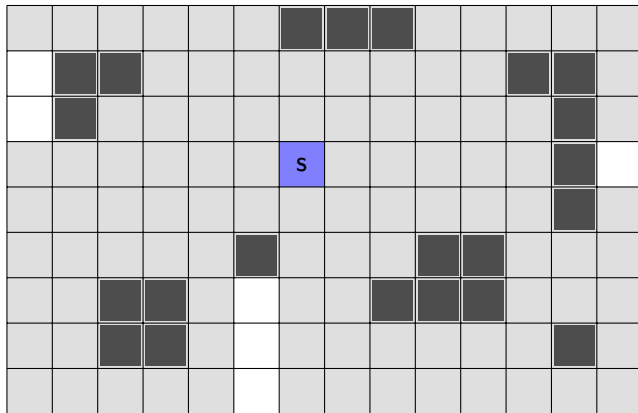


Default Moves

- Heuristic moves (e.g., based on the Octile heuristic)
- Easy to compute in constant time



Default Moves vs First Optimal Moves



= source node

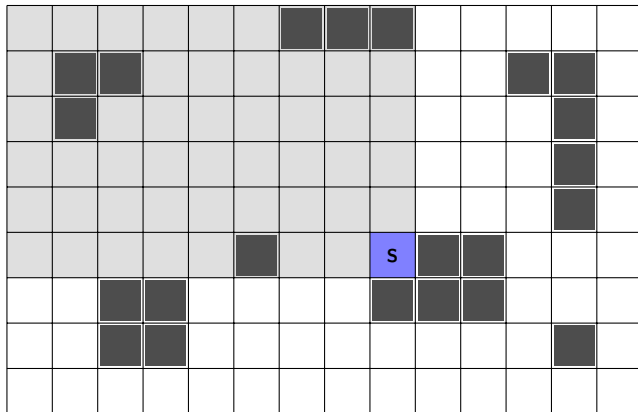


= default move which is also a first optimal move from S



= default move which is not a first optimal move from S

Default-Move Enhancement Around Obstacles



= source node

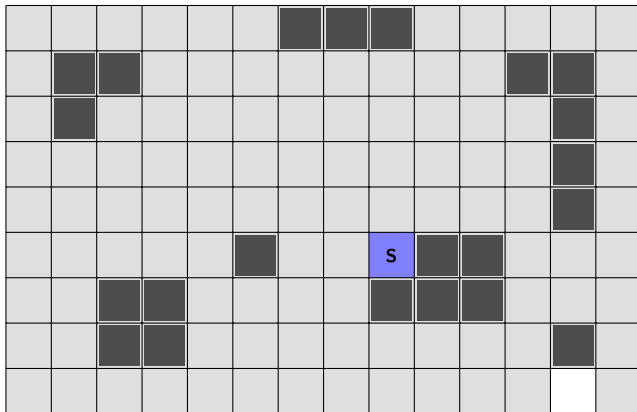





= default move which is also a first optimal move from S



= default move which is not a first optimal move from S

Default-Move Enhancement Around Obstacles



-  = source node
-  = enhanced default move which is also a first optimal move from S
-  = enhanced default move which is not a first optimal move from S

Compression with Redundant Symbols

Vanilla CPD compression

n n n w * se sw sw \rightarrow 1/n; 4/w; 5/se; 7/sw

Compression with redundant symbols

Add a redundant symbol \textcircled{h} when default move = optimal move:

n n n w * se sw sw
 \textcircled{h} \textcircled{h} \textcircled{h} \textcircled{h}

Compression with Redundant Symbols

Vanilla CPD compression

n n n w * se sw sw \rightarrow 1/n; 4/w; 5/se; 7/sw

Compression with redundant symbols

Chose symbol that compresses better:

n n n w * se sw sw \rightarrow 1/n; 4/Ⓜ; 7/sw
Ⓜ Ⓜ Ⓜ Ⓜ Ⓜ

Outline

Introduction

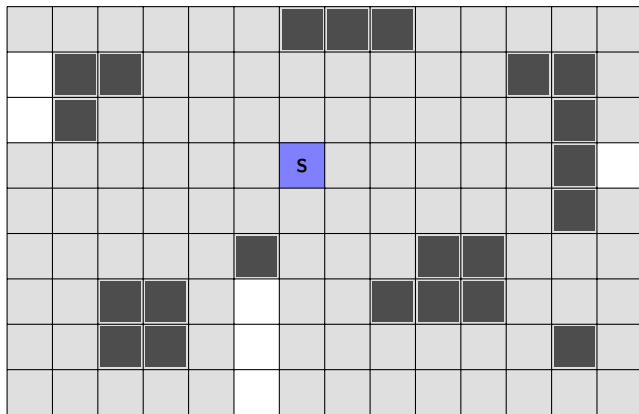
Redundant Symbols

Proximity Wildcards

Experimental Results

Conclusion

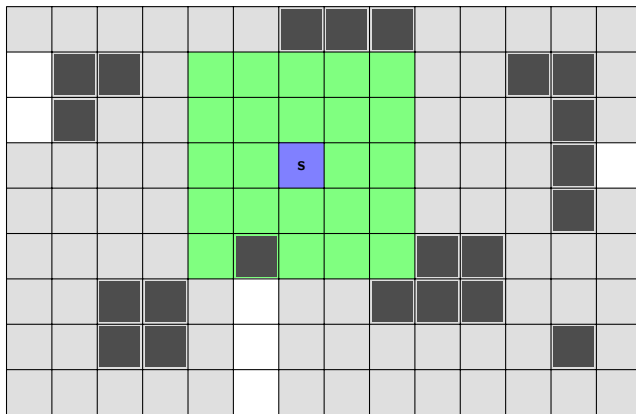
Wildcard Square







=

source node

Wildcard Square



-  = source node
-  = the heuristic move from s to this cell belongs to $T(t)$
-  = the heuristic move from s to this cell does not belong to $T(t)$
-  = the move is from s to this cell is replaced with a Proximity Wildcard

Compression with Proximity Wildcards

Proximity Wildcards

“Don’t care” symbols [Salveti et al., 2017] for the positions corresponding to a wildcard square.

n n n w * se sw sw

Compression with Proximity Wildcards

Proximity Wildcards

“Don’t care” symbols [Salvetti et al., 2017] for the positions corresponding to a wildcard square.

n n n w * se sw sw



n * n * * * sw *



1/n 7/sw

Outline

Introduction

Redundant Symbols

Proximity Wildcards

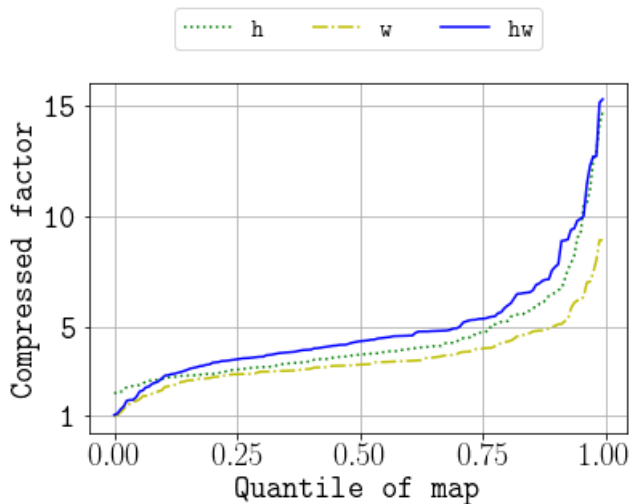
Experimental Results

Conclusion

Experimental Setup

- Developed on top of SRC Wildcard 2017 [Salvetti *et al.*, 2017]
- 8-connected gridmaps
- 155 maps from Dragon Age: Origins (from about 2,000 to 150,000 nodes)
- 9 maps from Dragon Age: Origins, Baldur's Gate II and Starcraft (from about 105,000 to 290,000 nodes)

DAO Dataset Results



h = Heuristic Symbol Enhancement
w = Proximity Wildcards
hw = Combination of the two

DAO Dataset Results

	Megabytes required by			
	Mean	std	Min	Max
SRC	6.52	11.2	0.011	68.6
Heuristic Symbol Enhancement (h)	1.85	3.82	0.003	29.6
Proximity Wildcards (w)	1.93	3.54	0.005	24.3
Combination of the two (hw)	1.48	2.82	0.004	21.7

Big Maps Results

Map	# Nodes	SRC	Megabytes required by		
			SRC + h	SRC + w	SRC + hw
AR0044SR	231469	507.1	14.3	38.4	9.1
AR0605SR	140922	179.4	25.4	23.0	14.2
AR0700SR	131852	69.0	37.8	23.3	20.3
Aftershock	166076	88.0	8.9	14.7	7.6
DarkContinent	285669	213.8	70.5	50.6	40.4
TheatreofWar	220816	170.8	53.9	42.2	36.6
hrt000d	106608	60.8	11.0	11.4	6.8
ost000a	130478	44.1	15.9	13.4	9.7
ost000t	105707	38.0	13.4	11.2	8.0

Big Maps Results

Map	# Nodes	SRC	Megabytes required by		
			SRC + h	SRC + w	SRC + hw
AR0044SR	231469	507.1	14.3	38.4	9.1
AR0605SR	140922	179.4	25.4	23.0	14.2
AR0700SR	131852	69.0	37.8	23.3	20.3
Aftershock	166076	88.0	8.9	14.7	7.6
DarkContinent	285669	213.8	70.5	50.6	40.4
TheatreofWar	220816	170.8	53.9	42.2	36.6
hrt000d	106608	60.8	11.0	11.4	6.8
ost000a	130478	44.1	15.9	13.4	9.7
ost000t	105707	38.0	13.4	11.2	8.0

AR0044SR compression factor is 55.7

Outline

Introduction

Redundant Symbols

Proximity Wildcards

Experimental Results

Conclusion

- We significantly reduced the size of CPDs
- With redundant symbols and proximity wildcards
- We define a new state of the art for CPDs

Future work

- Extending the area of the proximity wildcards
- Evaluating the proposed techniques on road graphs

References I



J. Baier, A Botea, D. Harabor, and C. Hernández.

A fast algorithm for catching a prey quickly in known and partially known game maps.
Computational Intelligence and AI in Games, IEEE Transactions on, PP(99), 2014.



Adi Botea, Jorge A. Baier, Daniel Harabor, and Carlos Hernández.

Moving target search with compressed path databases.

In *Proceedings of the International Conference on Automated Planning and Scheduling ICAPS*, 2013.



Adi Botea.

Ultra-fast optimal pathfinding without runtime search.

In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE'11)*, pages 122–127. AAAI Press, 2011.



Matteo Salvetti, Adi Botea, Alessandro Saetti, and Alfonso Gerevini.

Compressed Path Databases with Ordered Wildcard Substitutions.

In *Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS-17*, 2017.



Matteo Salvetti, Adi Botea, Alfonso Emilio Gerevini, Daniel Harabor, and Alessandro Saetti.

Two-oracle optimal path planning on grid maps.

In *ICAPS*, pages 227–231. AAAI Press, 2018.



Ben Strasser, Adi Botea, and Daniel Harabor.

Compressing optimal paths with run length encoding.

J. Artif. Intell. Res. (JAIR), 54:593–629, 2015.



Fan Xie, Adi Botea, and Akihiro Kishimoto.

A Scalable Approach to Chasing Multiple Moving Targets with Multiple Agents.

In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017.