

Overflow: Springertouren

Lutz Prechelt (prechelt@ira.uka.de)
Institut für Programmstrukturen und Datenorganisation
Universität Karlsruhe
7500 Karlsruhe 1

Erschienen in *Informatik-Spektrum* Juni 1992

1 Springertouren

Als eine *Springertour* bezeichnet man eine Folge von Springerzügen auf einem 8×8 Schachbrett, die auf irgendeinem Feld beginnt und dann mit 63 Sprüngen jedes Feld genau einmal besucht. Gibt es einen 64. Sprung, der dann wieder auf das Ausgangsfeld zurückführt, so heißt die Springer-tour *geschlossen*. Eine solche geschlossene Springer-tour nennt man in der Graphentheorie einen *Hamiltonian Cycle* in dem von allen erlaubten Sprüngen des Schachbretts aufgespannten Graphen. Eine geschlossene Springertour lässt sich als Hamiltonian Cycle eines anderen Graphen beschreiben. In der Durchlaufrichtung 127 definieren wir eine *kanonische Springertour* (im folgenden *kanonische Springertour*) als eine geschlossene Springer-tour, die von Verbindungspunkten besteht, die wir als *Werte* bezeichnen. Diese Werte sind die Werte der Springertouren, deren Werte nach allen

suchen diese Verfahren in der Regel nur einen einzigen Hamiltonkreis. Insgesamt ist diese Literatur für das Springertourenproblem nicht hilfreich.

Zum Suchen nach Springertouren fällt einem Informantikergerne natürlich als nächstes sofort die Tiefensuche ein:

Eine Teiltour mit i Sprüngen wird um einen Schritt verlängert, indem man die Nachbarn im Endfeld des aufzählt und dann den ersten freien als benutzt markiert, als nächsten Sprung mit der entstehenden Teiltour weiter versucht. Gibt es keinen freien so entfernt man den i -ten Sprung, friere Endfeld wieder als frei, bestehende kürzere Teiltour um Nachbarn zu verlängern.

3 das Feld f_7 er-

en; wenn man

uren ge-

ei-

n-

noch möglich sind, da hier die Gefahr,
das betreffende Feld nicht wieder zu er-
reichen und es somit ganz auszulassen,
verhältnismäßig am größten ist, während
natürlich diejenigen Felder, die noch mit
einer größeren Zahl von freien Feldern
sich rösseln, eher von einem dieser aus
später noch erreicht werden können.

haben diese Heuristik (nennen wir sie H_1)

Beispiel für heuristische Programmierung in

Lehrveranstaltung Informatik II vorgestellt

in kleinen Wettbewerb ausgeschrieben:

das Programm für einen Mini-

Computer schnellsten 10.000 Turen findet.

prinzipiell in der Lage sein,

es gibt. Der Sinn dieser

Teilnehmenden dazu

auf die Suche nach

ein bekanntes Pro-

gramm es auch dort

gefunden gibt.

ein vol-

ständigereich-

Auf-

von

gori thmischen Unterschie-
 e optimiert waren. Das
 nai vemund gerad-
 niertem Assem
 enstrukt u-
 glei chen

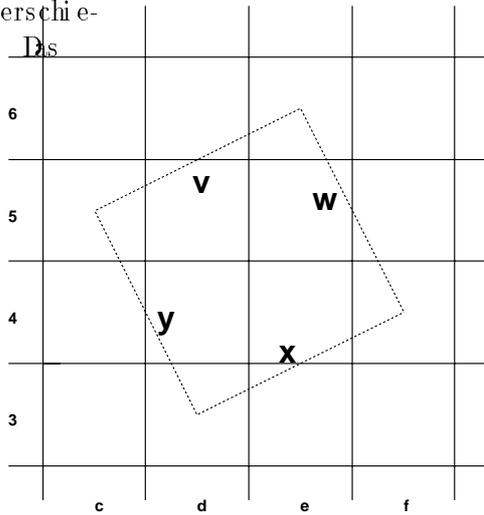


Abbildung 2:
Spruenge im Bereich der Brettmitte

schränken sich die Suchbäume in diesem Problem
 über die benutzten Felder auch noch gegenseitig.

Des Weiteren wären zum Beispiel folgende Verfah-
 ren denkbar:

H_6 : Die Heuristik H_4 kann noch verbessert wer-
 den, wenn jeweils zurückgesetzt wird, sobald sich
 herausstellt, daß die gefundene Teillösung nur
 Lösungen erzeugen kann, die in eine der erst durch
 Sprünge zu erzeugenden Klassen gehören.

nur offene Springertouren auf einem
 freien Anfangs- und Endfeld so
 dort aus in die andere Hälfte
 in mehrere solche Teiltou-
 ren in einem Anfangsfeld A_1
 mehrere Teiltouren
 der anderen
 Anfangs-
 A_1 aus
 nur von
 p

zum Fehlen bestimmter Lösungen führen kann,
veranschaulicht, daß Heuristiken stets darauf hin
zu untersuchen sind, ob sie z. B. suboptimale oder
falsche Lösungen finden können, wie oft und bei
welchen Lösungsklassen dies geschieht und ob der
Lösungsraum komplett abgedeckt wird oder wel-
davon nicht.

Vorbereitung eines Wettbewerbs hat sich
des Element für eine Vorlesung sehr
s. Bereicherung der Lehre zu be-
r. Nachahmung empfohlen; wir
in Zukunft ständig nach
ich ein Wettbewerb

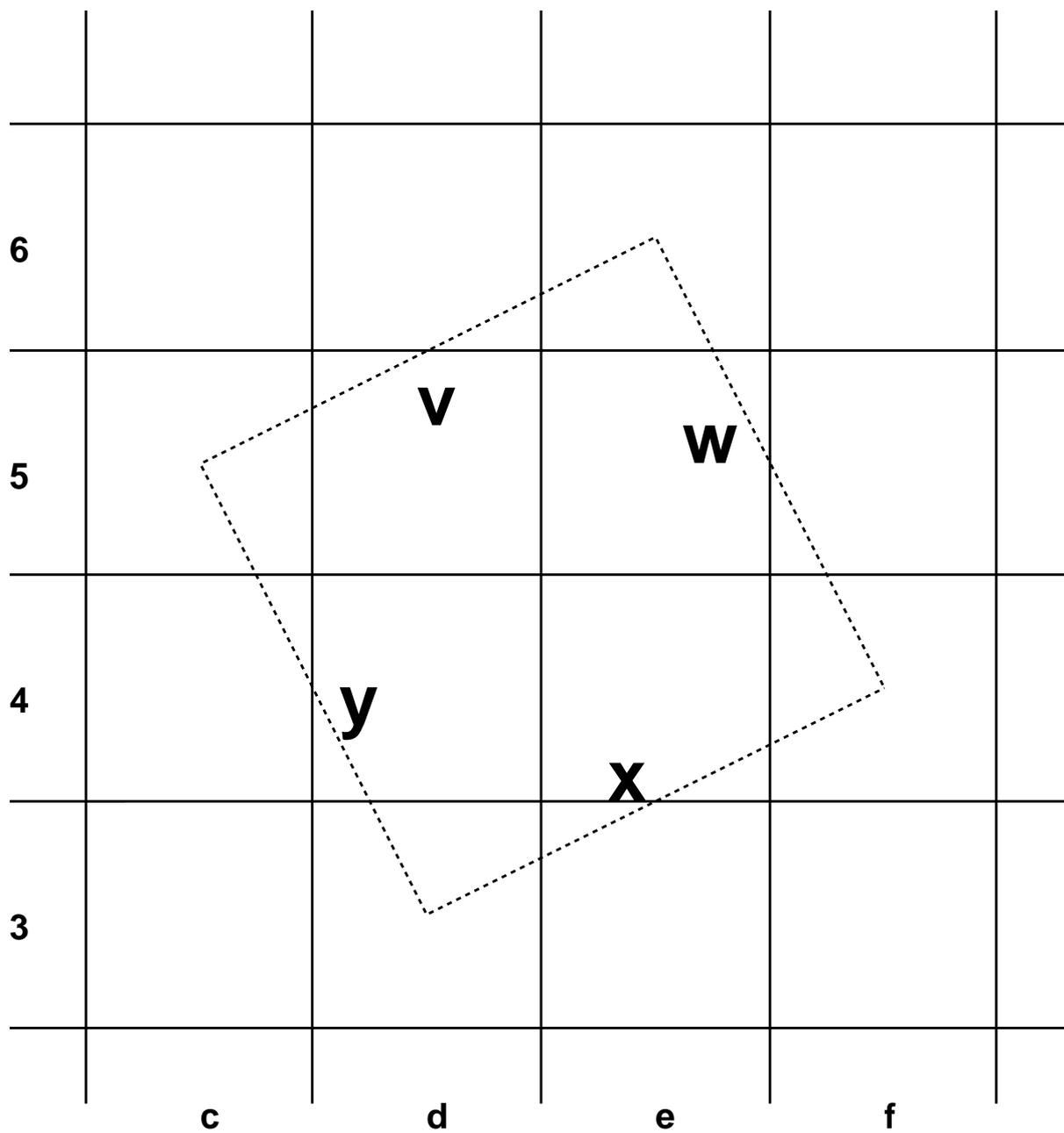
Beispiele. 5. Aufl.
t. Leipzig
Teub-

Ge-
an

4 <i>a4</i>	6 <i>b4</i>	8 <i>c4</i>	8 <i>d4</i>	
4 <i>a3</i>	6 <i>b3</i>	8 <i>c3</i>	8 <i>d3</i>	
3 <i>a2</i>	4 <i>b2</i>	6 <i>c2</i>	6 <i>d2</i>	
2 <i>a1</i>	3 <i>b1</i>	4 <i>c1</i>	4 <i>d1</i>	

Abbildung 1:

Nachbarn von a1; Grade; Feldnamen



**Abbildung 2:
Spruenge im Bereich der Brettmitte**