Overflow: Springertouren

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

Erschienen im Informatik–Spektrum Juni 1992

1 Springertouren

8 6 4 — c4 d4 c3 d3 c2 d2 c1 d1

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 einen 64. Sprung, der dann wieder auf das angsfeld zurückführt, so heißt die Springer- *eschlossen*. Eine solche geschlossene Sprin- r nennt man in der Graphentheorie einen *reis* in dem von allen erlaubten Sprin- dem Schachbrett aufgespannten Gra-

geschlossenen Springertour las-
hl eines anderen Anfangsfeldes
ler Durchlaufrichtung 127
b definieren wir eine *ka-*
en
ertour (im folgenden
hlossene Springer-
etz von Verbindun-
i nnt. We. wir
esteht darin, daß
h diese Ein-
e Touren) die-
en, deren
che nach al-
ui va-
nach allen

s

ouren

e Algorith-
suchen.
garan-
eine
so

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 Informatikergehirn natürlich als nächstes sofort die Tiefensuche ein:

Eine Teiltour mit i Sprüngen wird um einen
chritt verlängert, indem man die Nachbarn ih-
Endfeldes aufzählt und dann den ersten freien
ls benutzt markiert, als nächsten Sprung
mit der entstehenden Teiltour weitere
n versucht. Gibt es keinen freien
o entfernt man den i -ten Sprung,
Srige Endfeld wieder als frei,
tehende kürzere Teiltour um
hbarn zu verlängern.
3 das Feld f7 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)
eispiel für heuristische Programmierung in
ührungsvorlesung Informatik II vorgestellt
n kleinen Wettbewerb ausgeschrieben:
das Programm für einen Macin-
chnellsten 10.000 Touren findet.
nzipiell in der Lage sein,
es gibt. Der Sinn dieser
e Teilnehmenden dazu
auf die Suche nach
in bekanntes Pro-
es auch dort
rungen gibt.

ein vol-
gereich-
Auf-
von

en algorithmischen Unterschie-
ieden optimiert waren. Das
haus naiv vermund gerad-
optimiertem Assem-
gsdatenstruktu-
demgleichen

hervor-
ngs die
rten

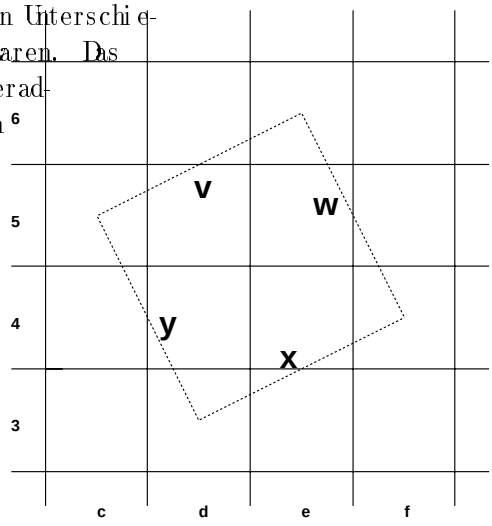

Abbildung 2:
Spruenge im Bereich der Brettmitte

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

Des Weiteren wären zum Beispiel folgende Verfahren denkbar:

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

nur offene Springertouren auf einem
eren Anfangs- und Endfelder so
dort aus in die andere Hälfte
un mehrere solche Teiltou-
ichem Anfangsfeld A_1
hrere Teiltouren
der anderen
Anfangs-
A_1 aus
r von
p

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

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

che Spiele. 5. Aufl.
eswelt. Leipzig
gart: Teub-

I, Wege-
tonian
ics.

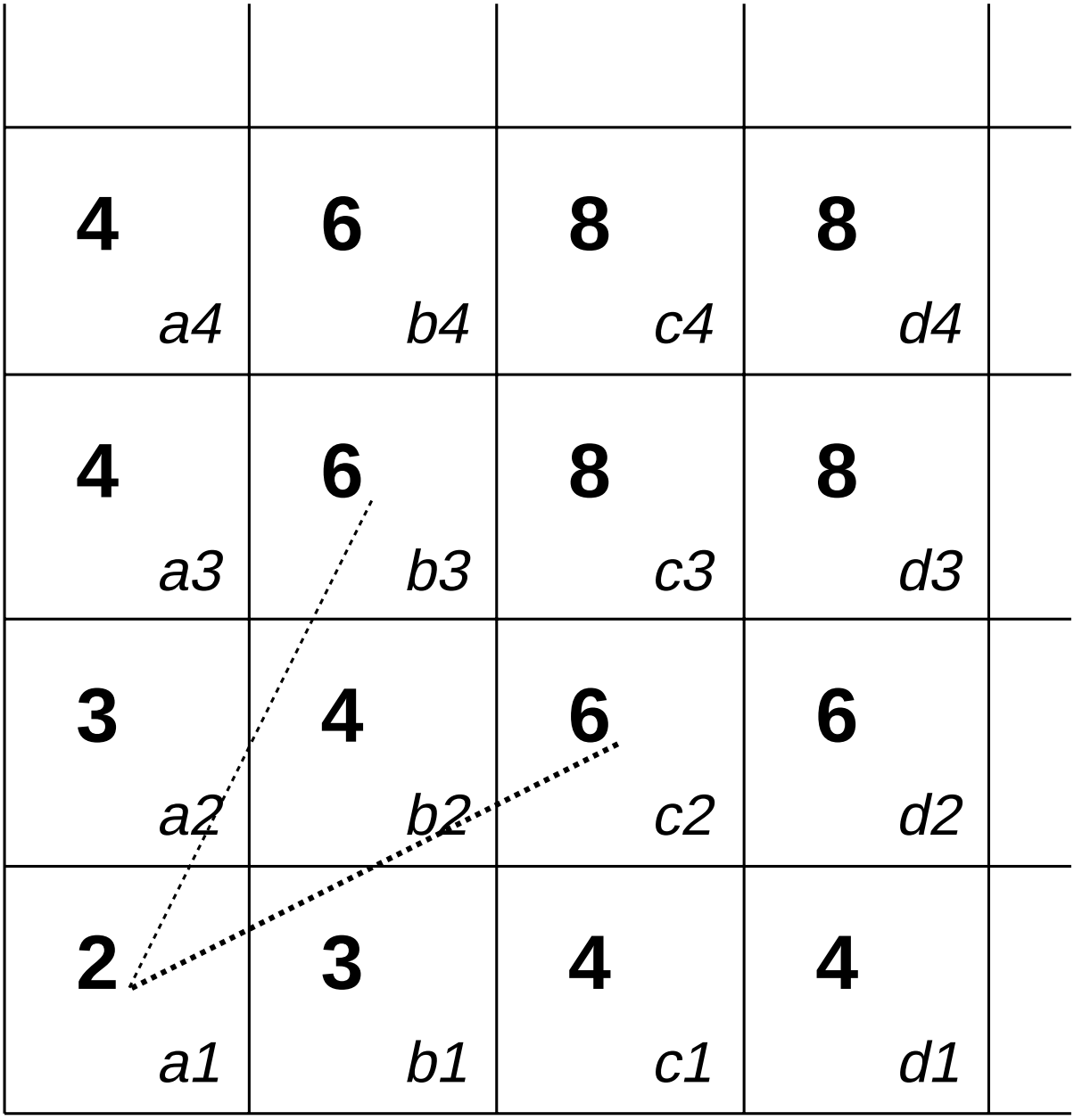

Abbildung 1:

Nachbarn von a1; Grade; Feldnamen

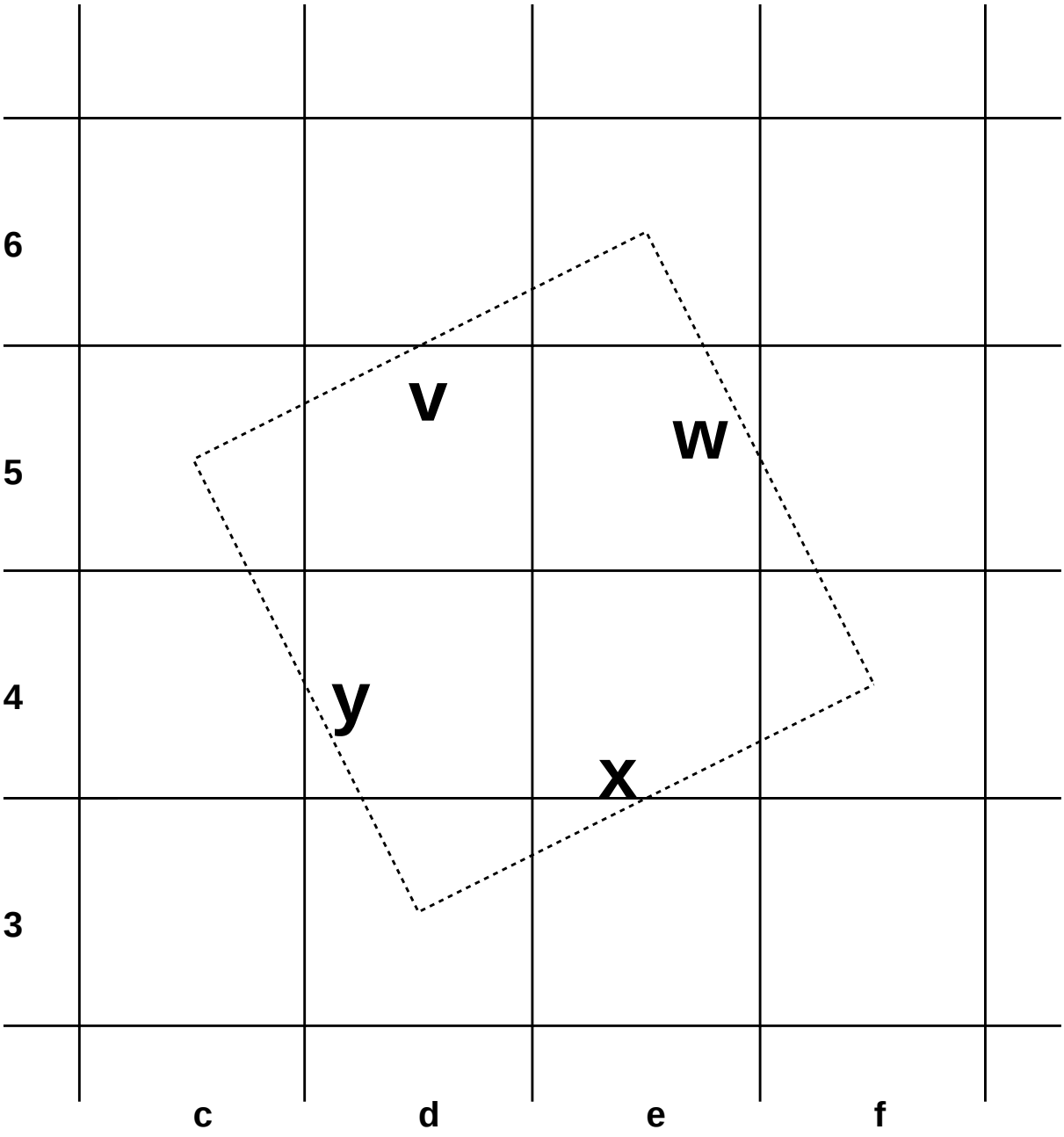

Abbildung 2:
Spruenge im Bereich der Brettmitte