
H. PROSSER & SONS
H. PROSSER & SONS

A talk with 3 titles

By

Patrick Prosser

Research … how not to do it

LDS revisited (aka Chinese whispers)

Yet Another Flawed Talk by Patrick Prosser

Chinese whispers - Wikipedia, the free encyclopedia - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Address http://en.wikipedia.org/wiki/Telephone_game Go

Your continued donations keep Wikipedia running! Sign in / create account

article | discussion | edit this page | history

WIKIPEDIA
The Free Encyclopedia

navigation
- Main page
- Contents
- Featured content
- Current events
- Random article

interaction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

search

Go Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

• Learn more about citing Wikipedia •

Chinese whispers

From Wikipedia, the free encyclopedia
(Redirected from Telephone game)

"Telephone game" redirects here. For the retired game from "The Price Is Right", see Telephone Game.

Chinese whispers	
Players:	3 or more
Age range:	5 and up
Setup time:	< 5 minutes
Playing time:	5–15 minutes
Random chance:	Low
Skills required:	listening, whispering

Chinese whispers or **Telephone** is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first. It is most often played by children as a party game or in the playground. It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours.[1]

The game has many other names, including the **telephone game**, **Broken Telephone**, **operator**, **grapevine**, **whisper down the lane** and **Pass It Down**. In the United States, "Telephone" is the most common name for the game.[1] The name "Chinese whispers" reflects the former stereotype in Europe of the Chinese language as being incomprehensible.[2] It is little-used in the United States and may be considered offensive.[3] However, it remains the common name in the United Kingdom and many British-influenced countries, where it is not generally considered politically incorrect.[4] In practice the game is also referred to by describing how to play, without giving it a specific name.

Contents [hide]

How to play [edit]

Internet

start papers Microsoft Po... korf96impro... harvey95limi... Chinese whi... 12:39

Send reinforcements.
We're going to advance.


Send three and fourpence.
We're going to a dance!

Quick Intro

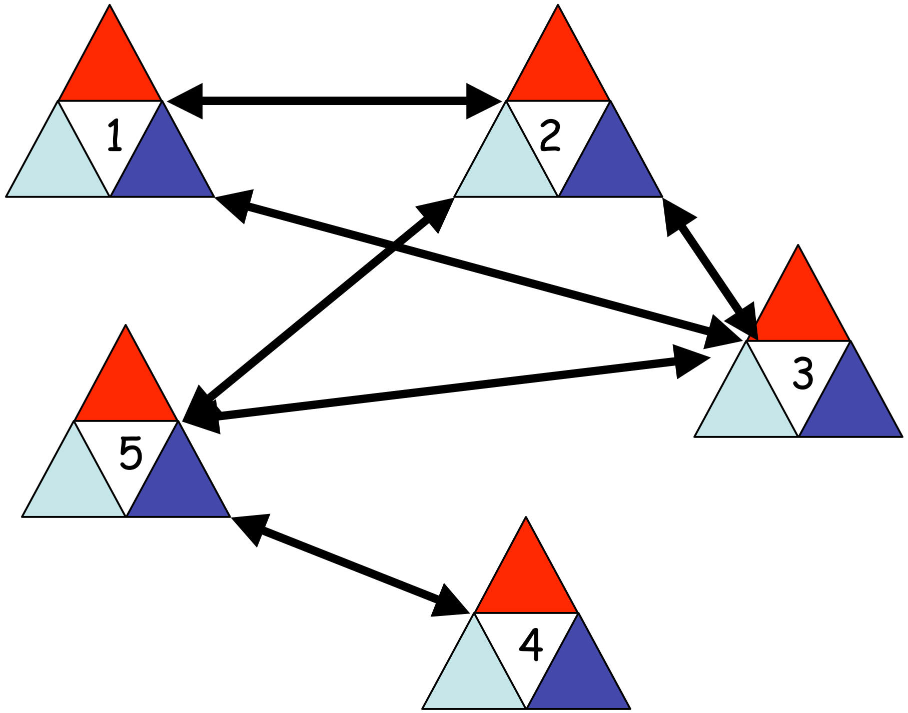
A refresher

- Chronological Backtracking (BT)
- what's that then?
- when/why do we need it?

Limited Discrepancy Search (lds)
- what's that then

Then the story … how not to do it

An example problem (to show chronological backtracking (BT))



Colour each of the 5 nodes, such that if they are adjacent, they take different colours

A Tree Trace of BT (assume domain ordered {R,B,G})

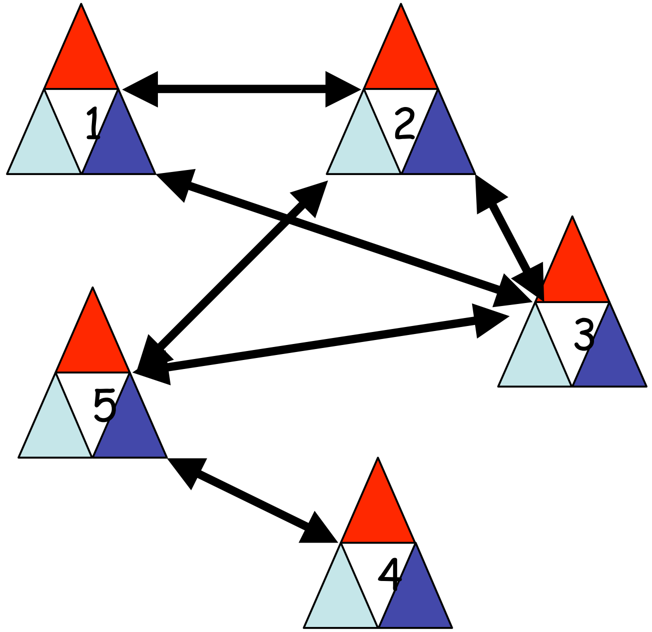
v1

v2

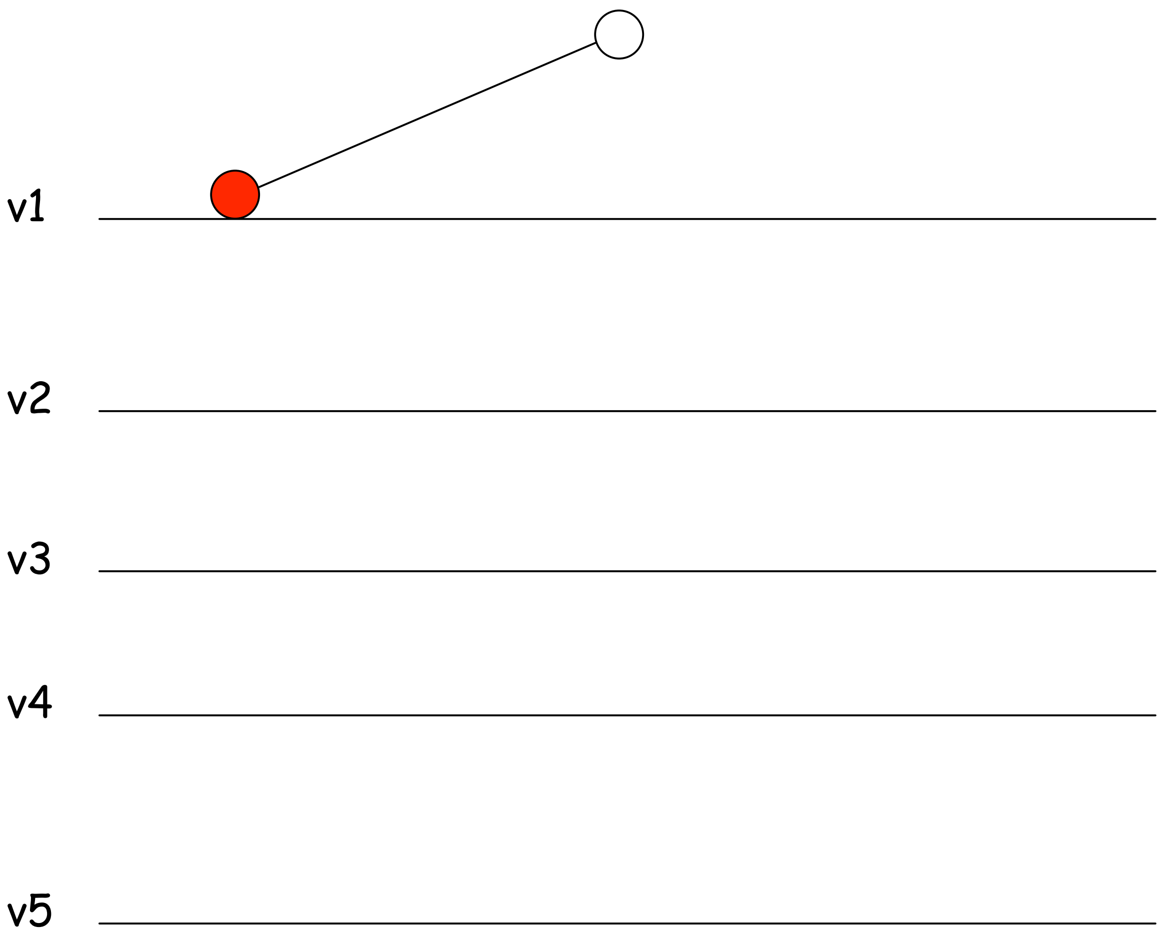
v3

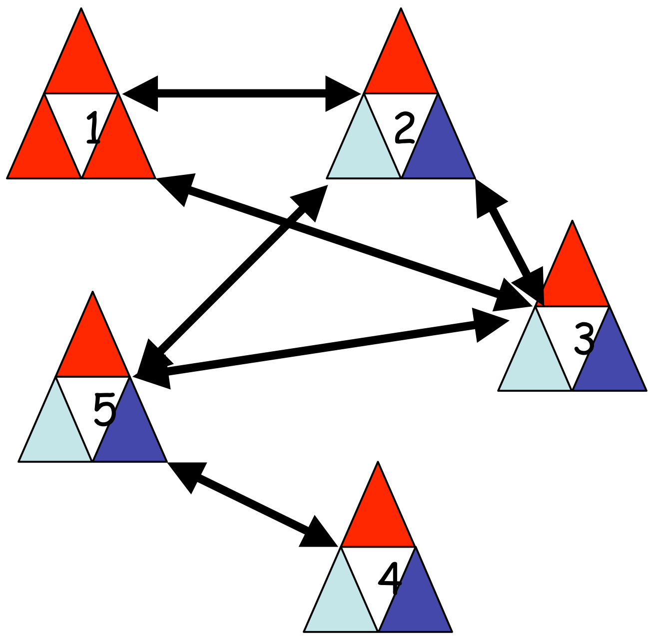
v4

v5

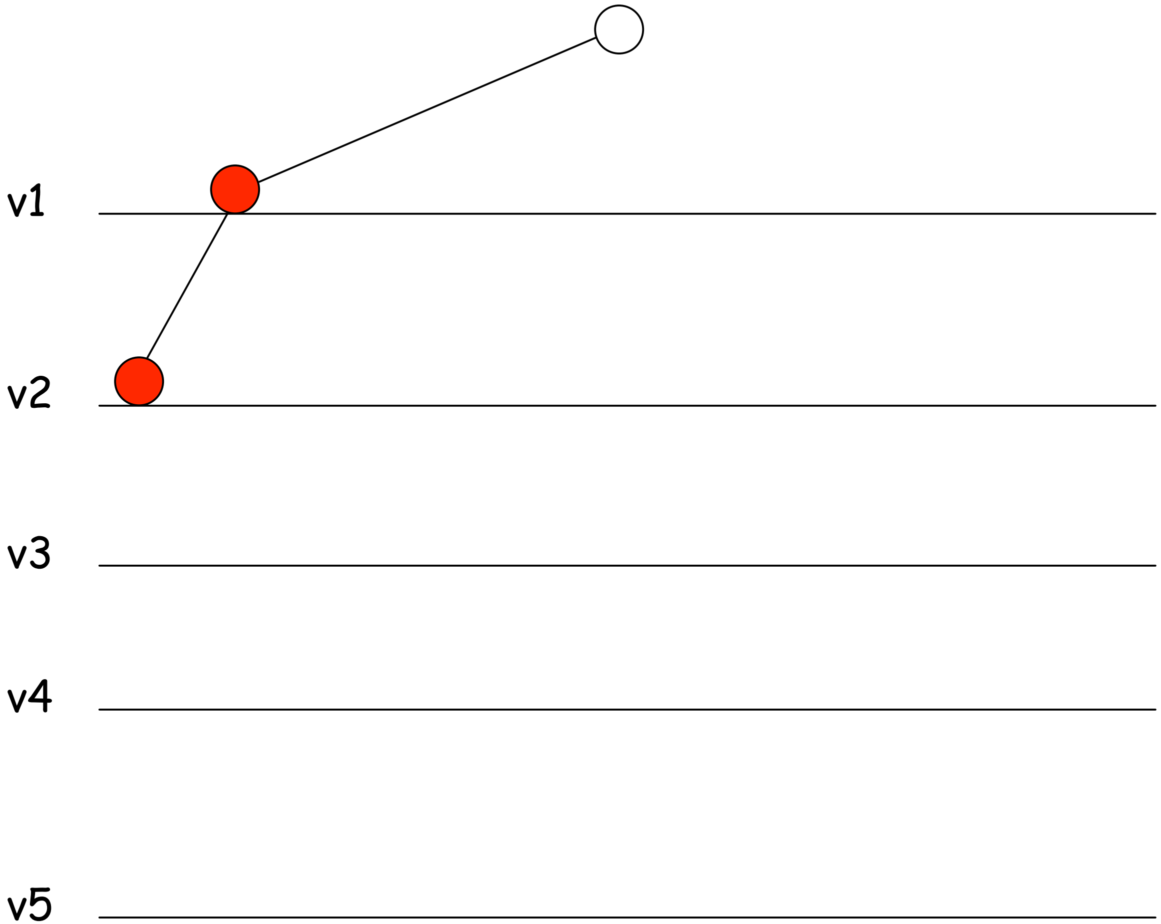


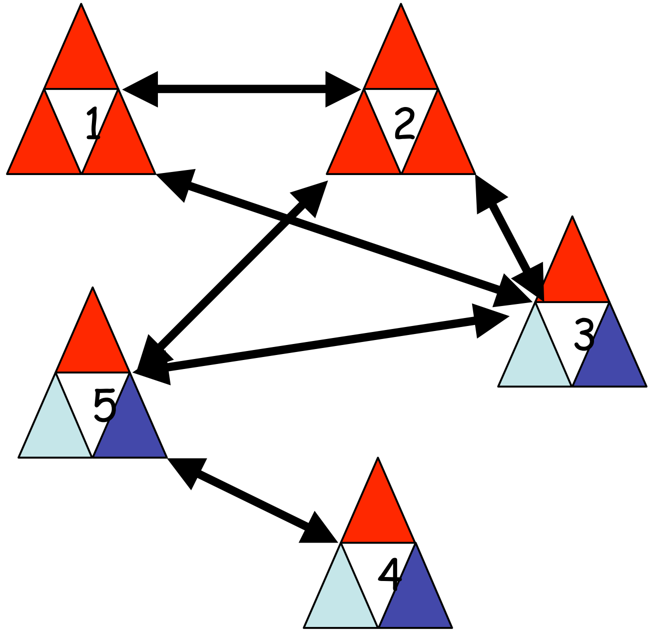
A Tree Trace of BT (assume domain ordered {R,B,G})



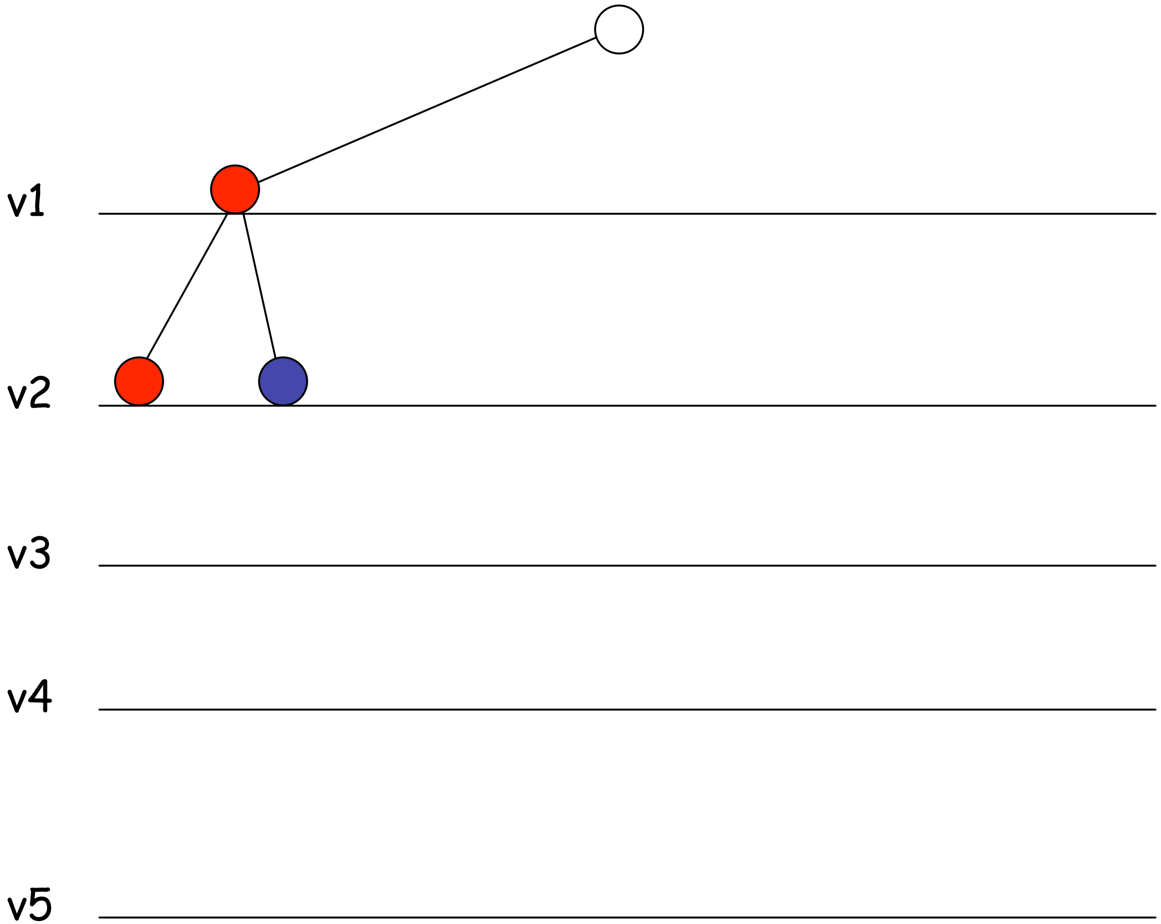


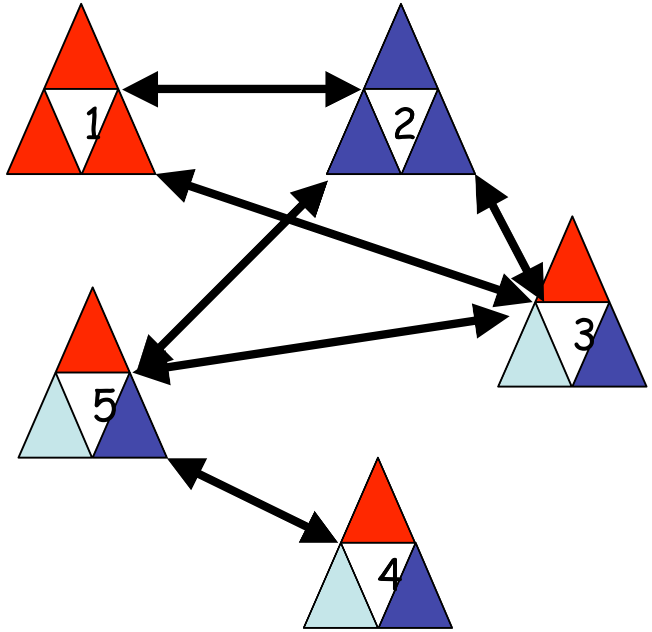
A Tree Trace of BT (assume domain ordered {R,B,G})



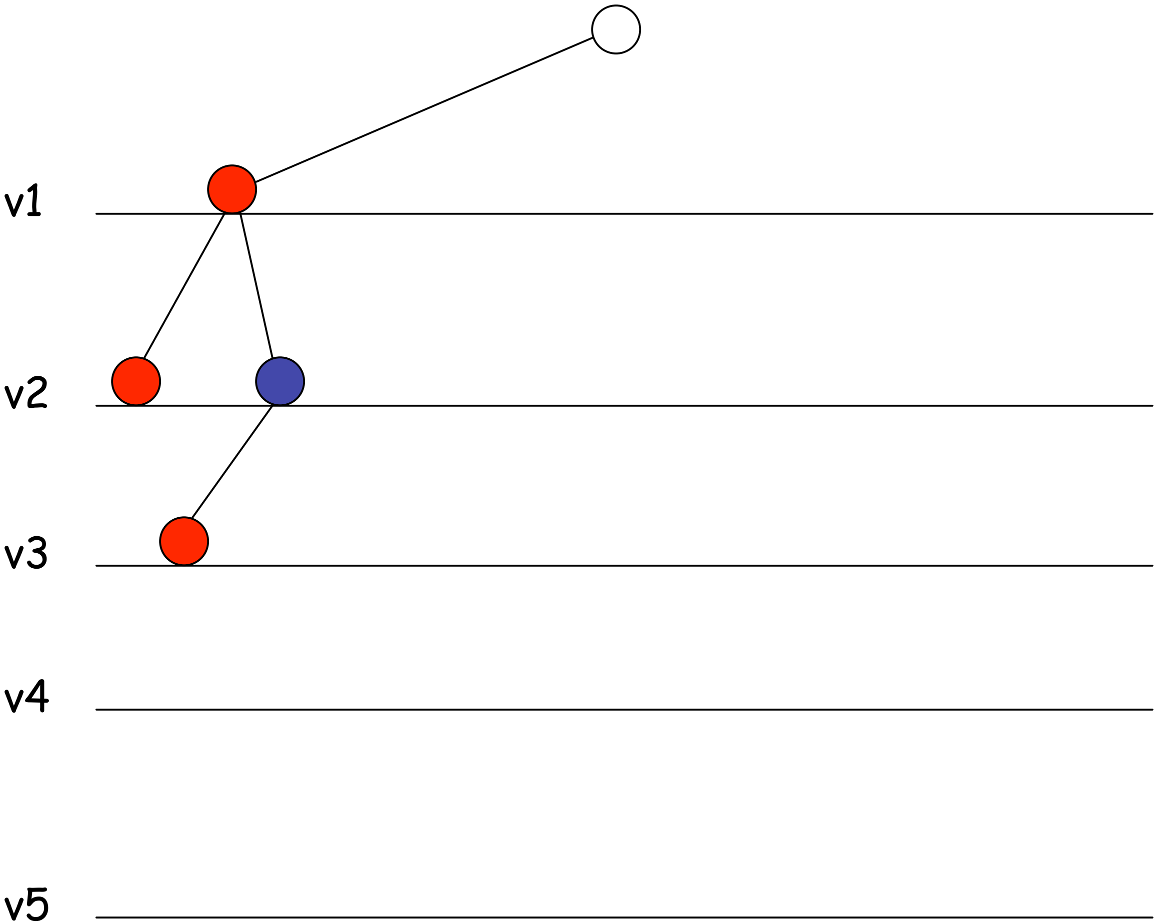


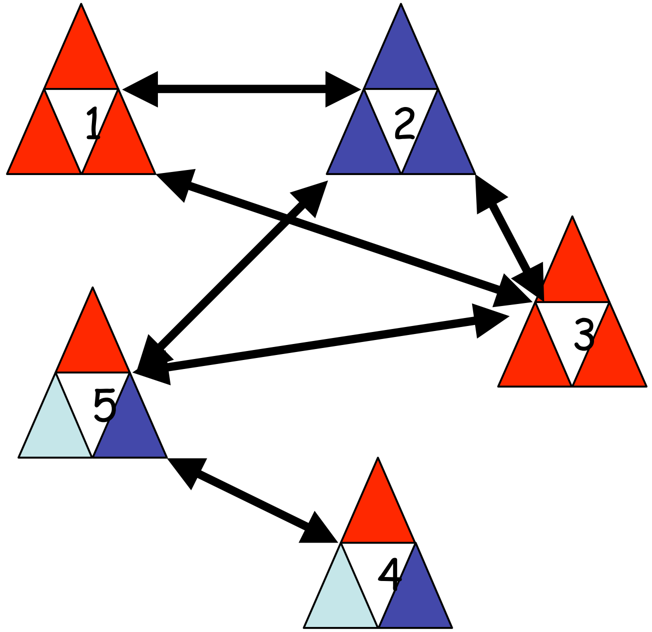
A Tree Trace of BT (assume domain ordered {R,B,G})



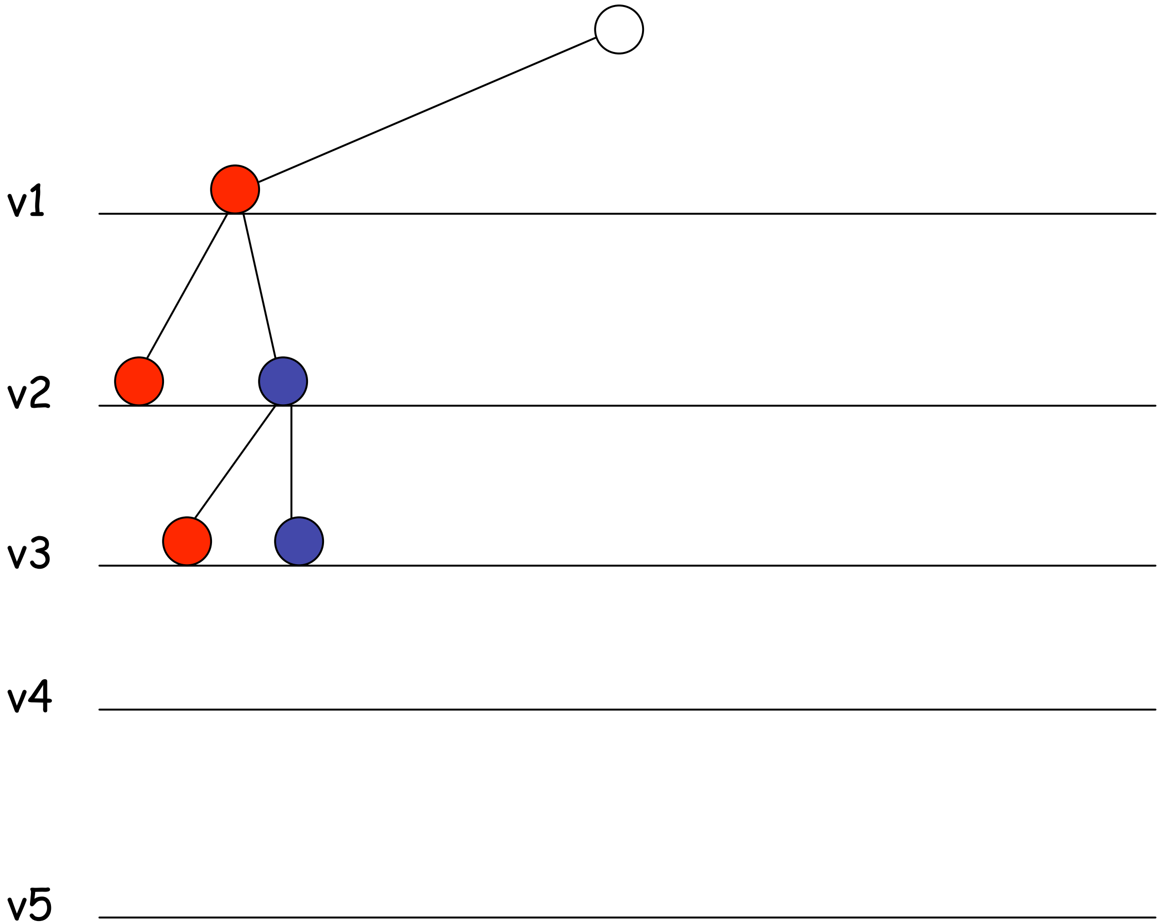


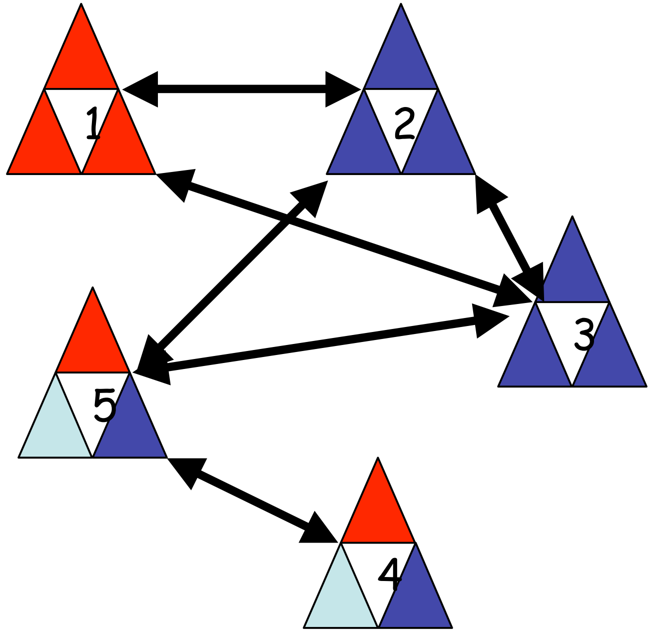
A Tree Trace of BT (assume domain ordered {R,B,G})



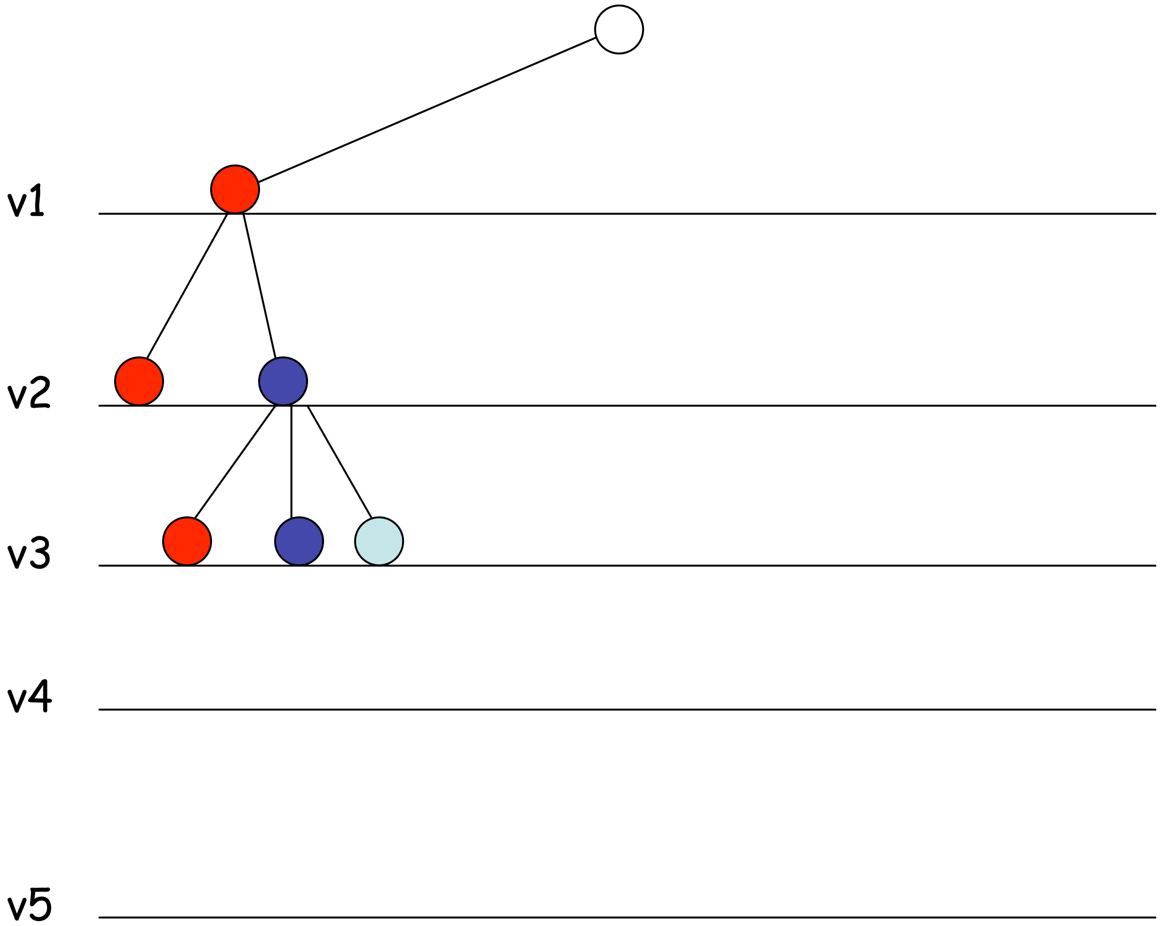


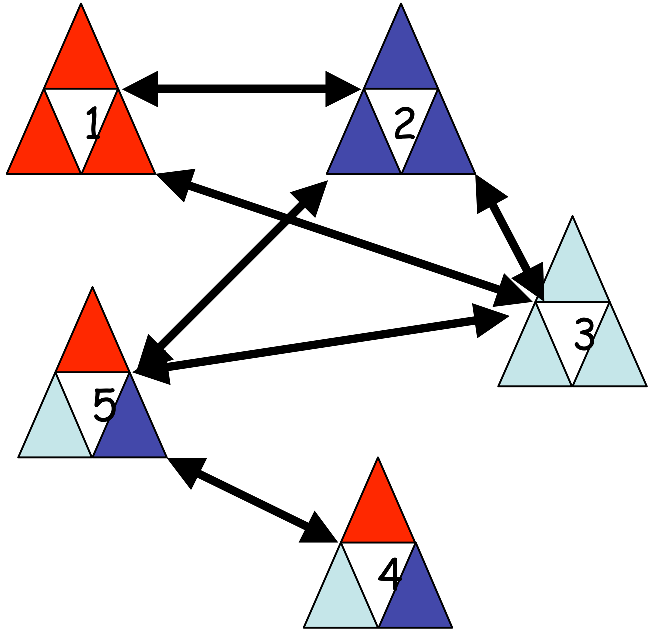
A Tree Trace of BT (assume domain ordered {R,B,G})



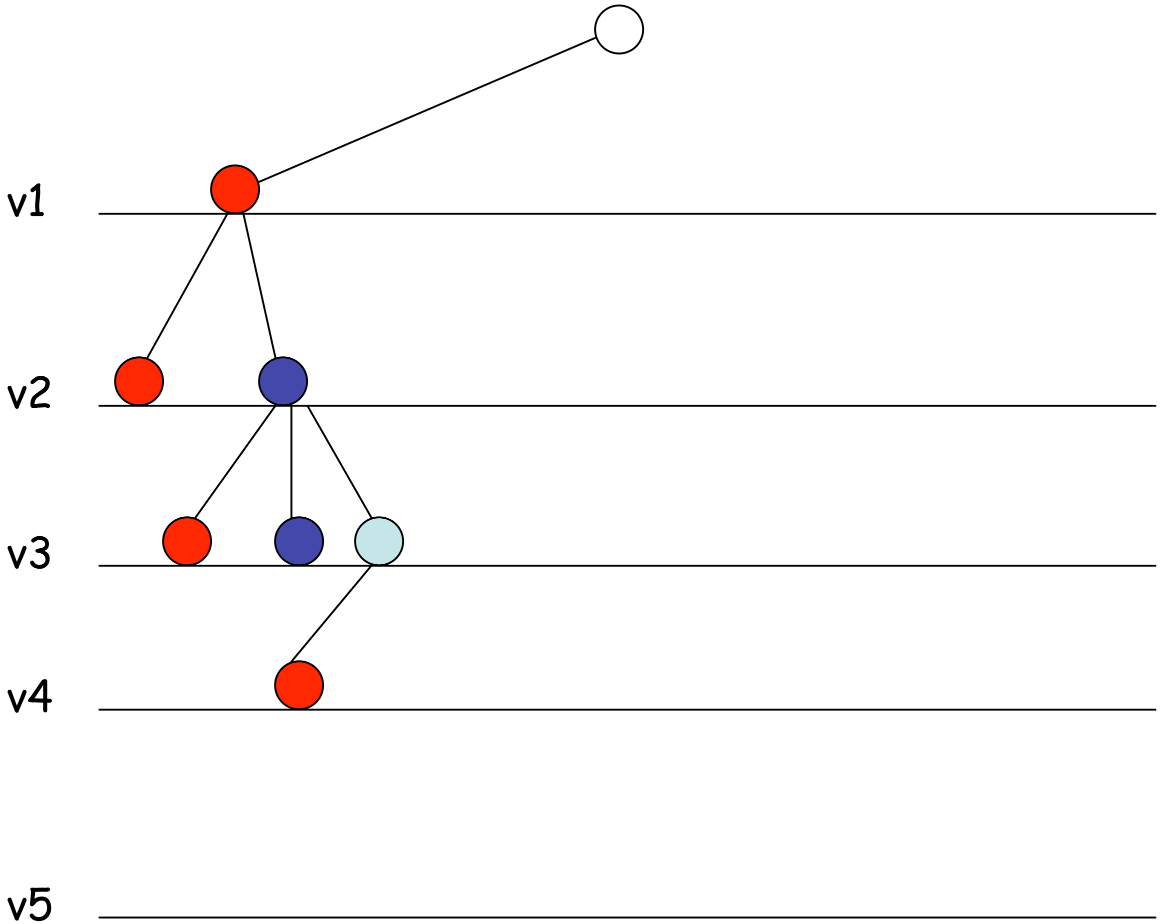


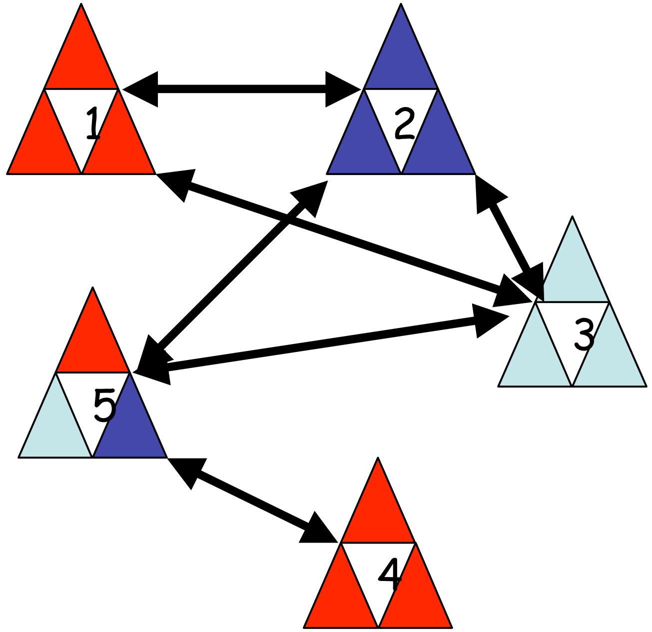
A Tree Trace of BT (assume domain ordered {R,B,G})



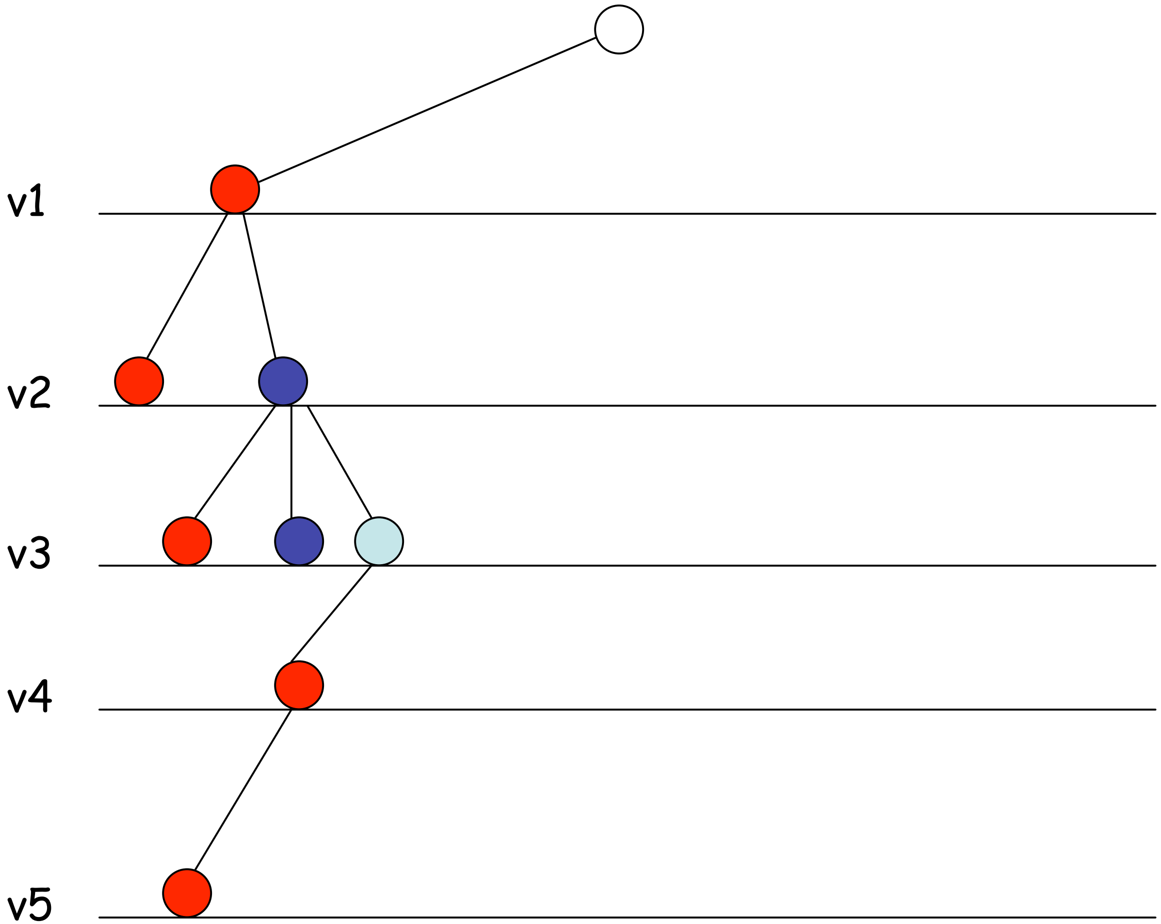


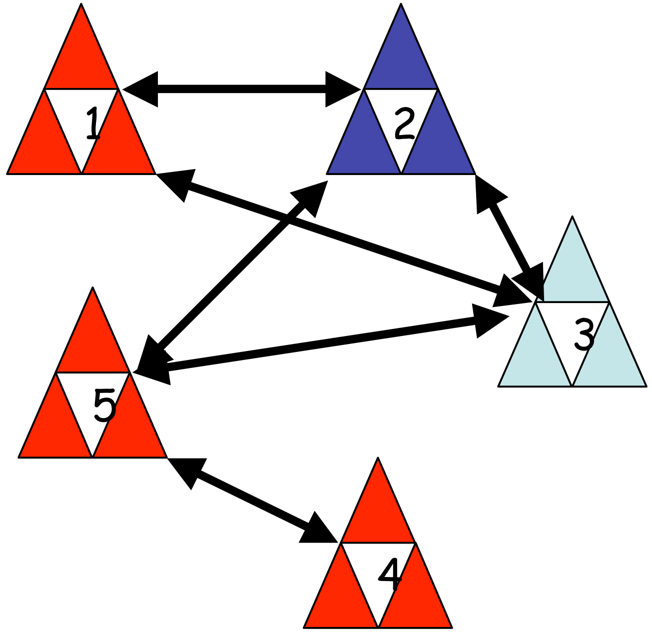
A Tree Trace of BT (assume domain ordered {R,B,G})



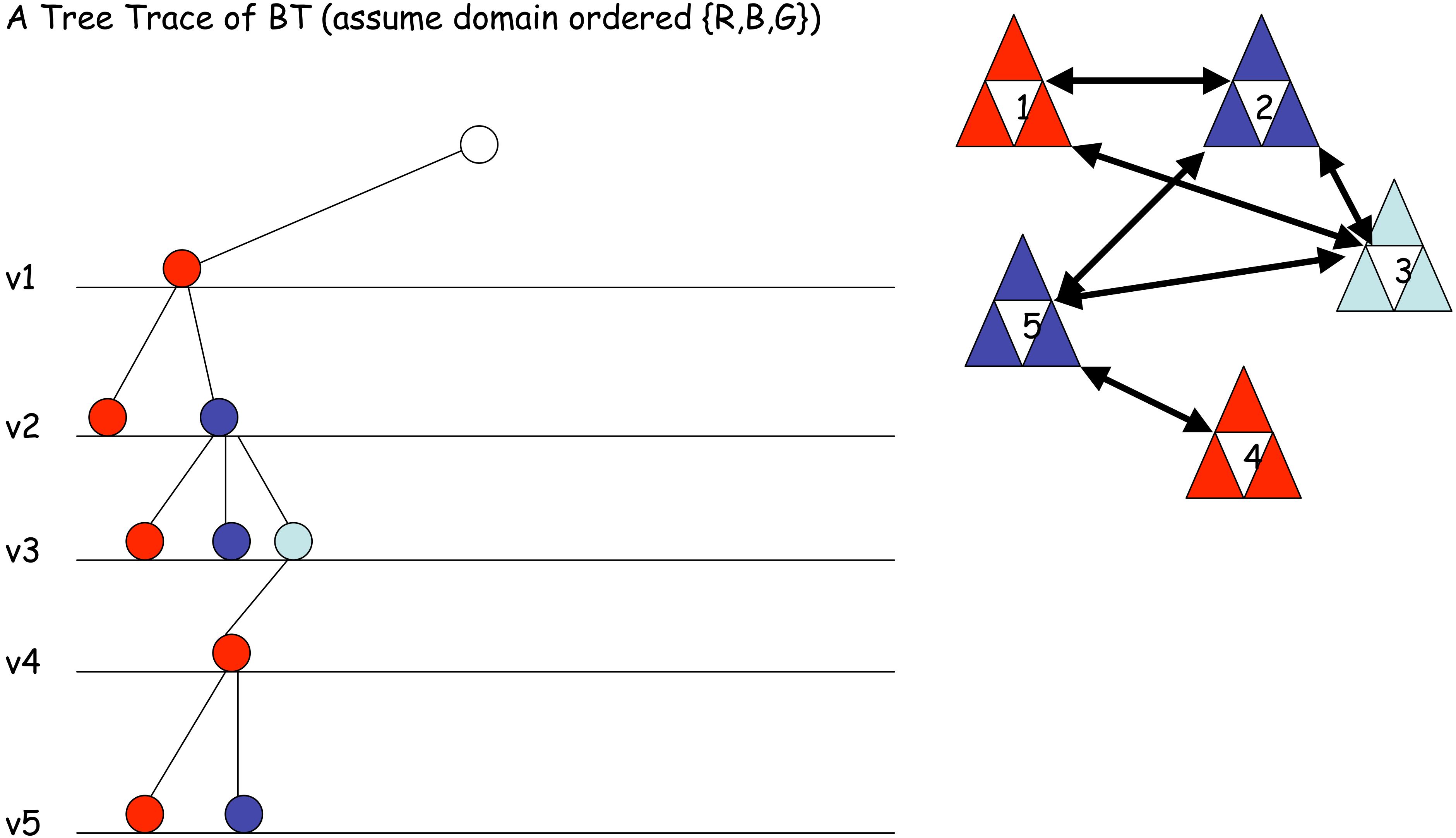


A Tree Trace of BT (assume domain ordered {R,B,G})

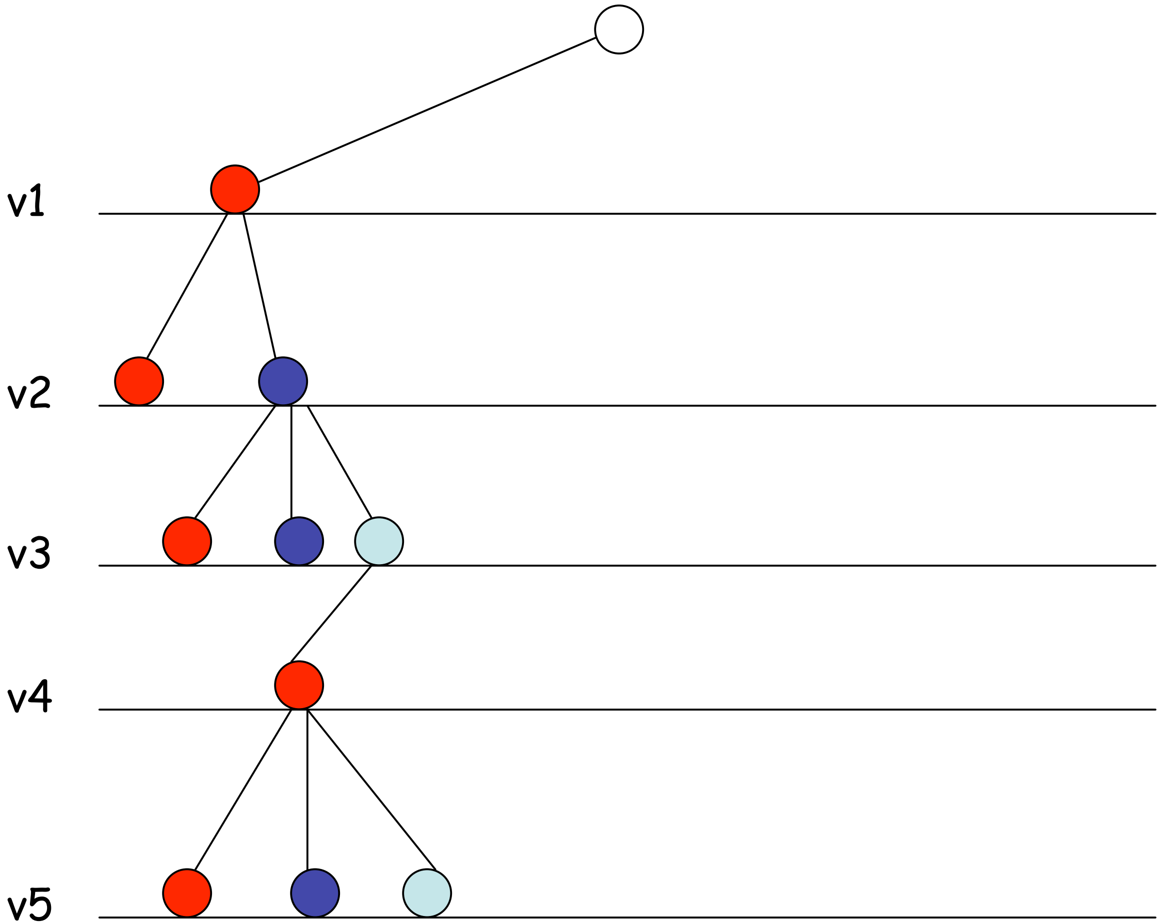


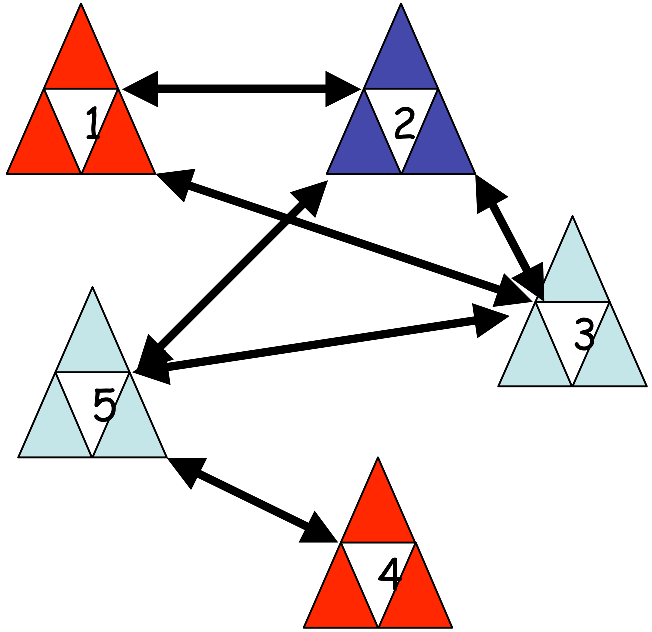


A Tree Trace of BT (assume domain ordered {R,B,G})

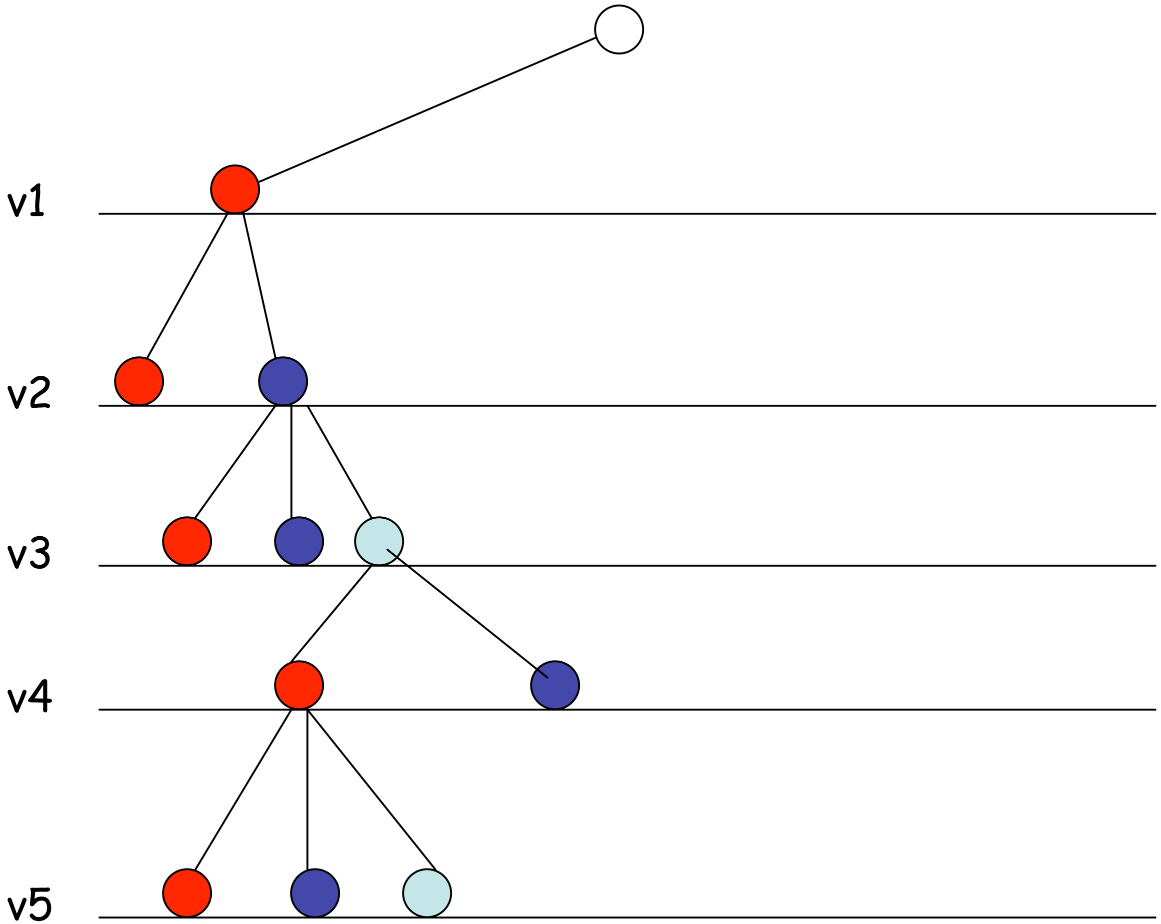


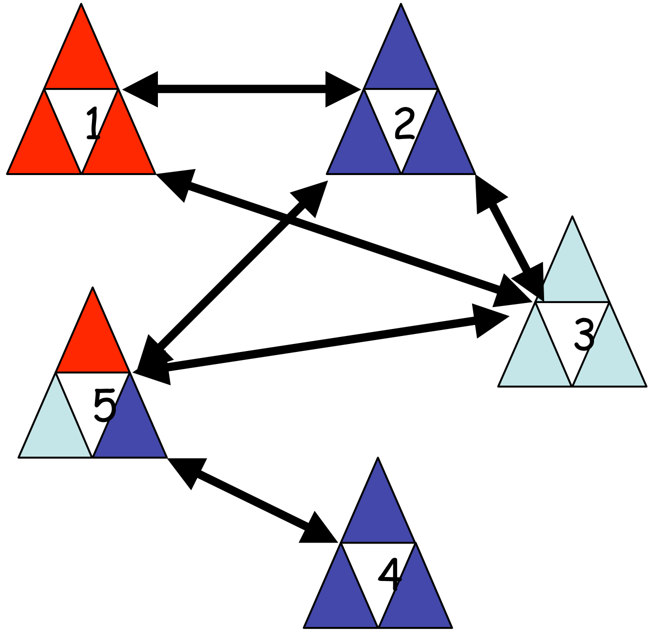
A Tree Trace of BT (assume domain ordered {R,B,G})



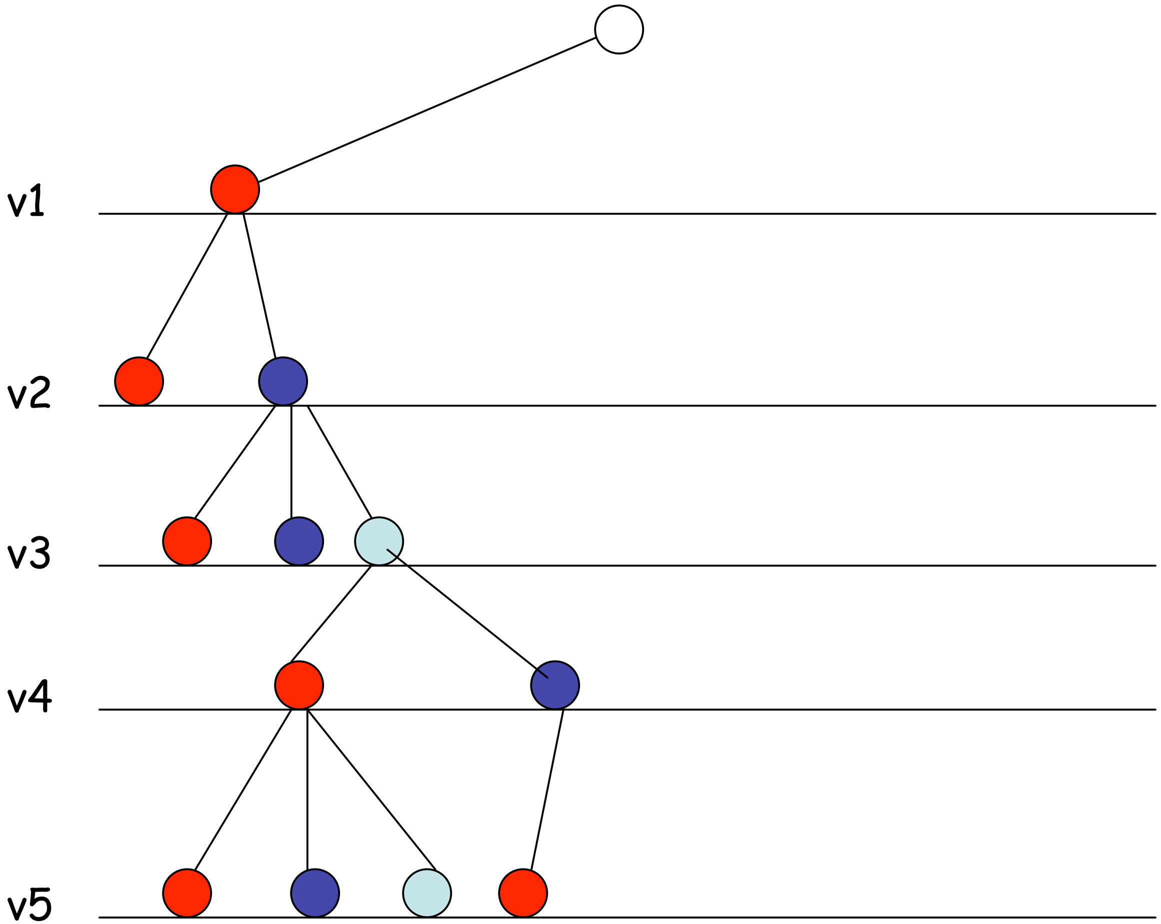


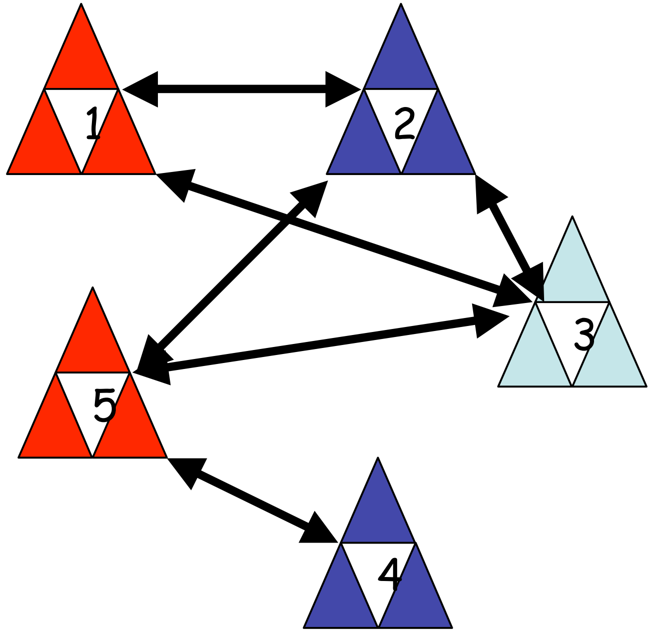
A Tree Trace of BT (assume domain ordered {R,B,G})





A Tree Trace of BT (assume domain ordered {R,B,G})





Could do better

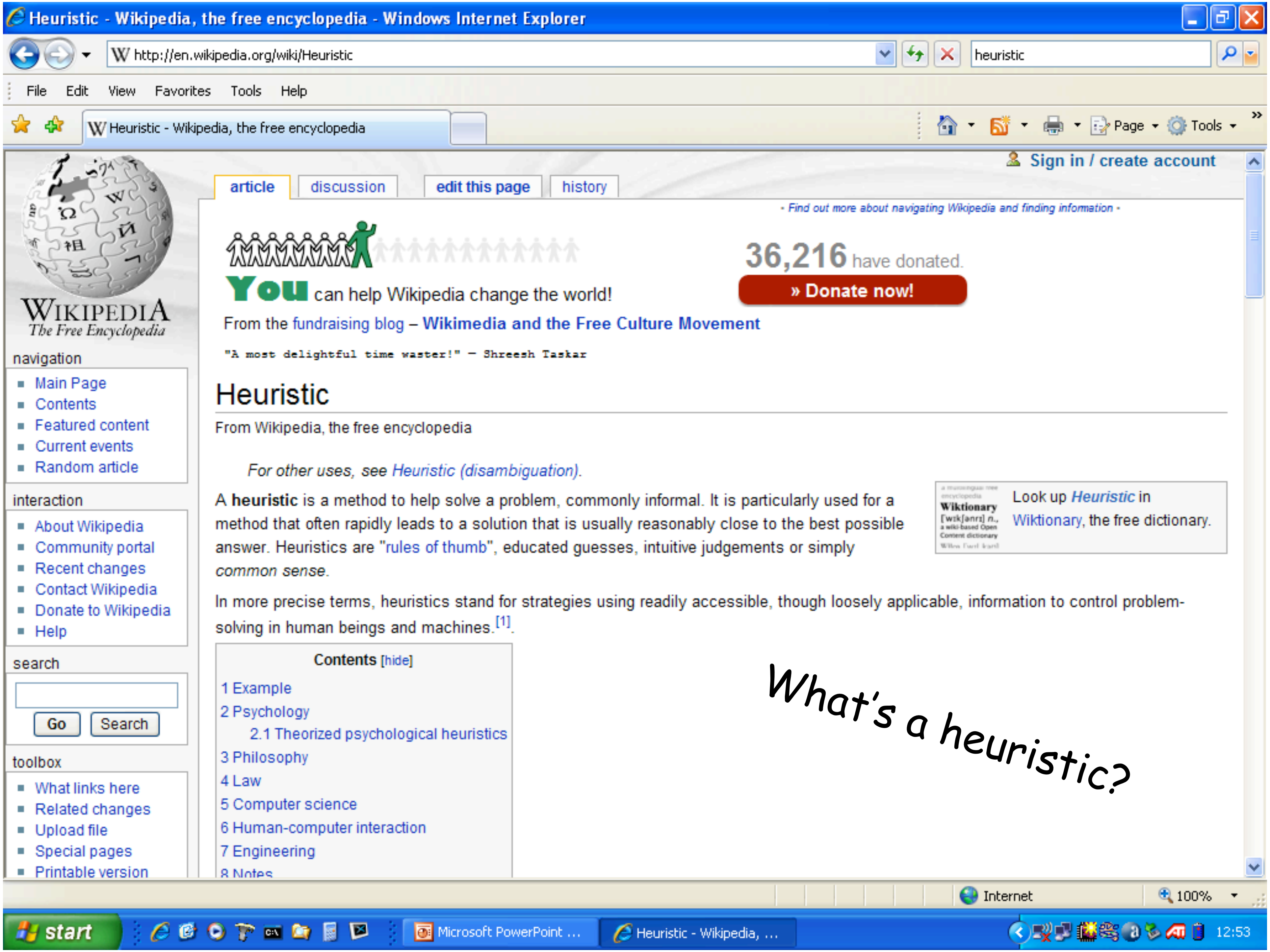
Improvements:

- when colouring a vertex with colour X
  - remove X from the palette of adjacent vertices
- when selecting a vertex to colour
  - choose the vertex with the smallest palette
  - tie break on adjacency with uncoloured vertices

An inferencing step

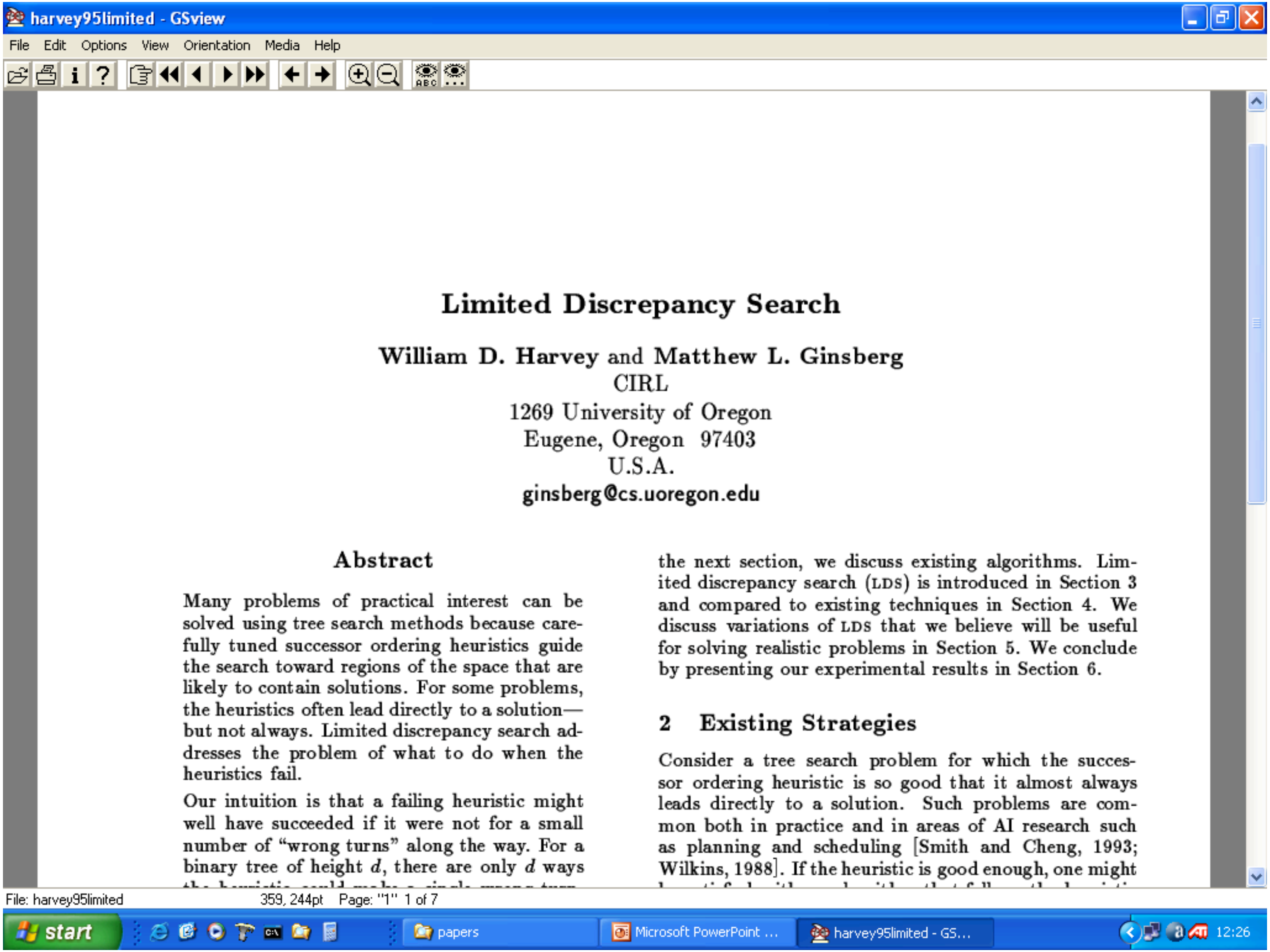
A heuristic (Brelaz)

Conjecture: our heuristic is more reliable as we get deeper ir



What's a heuristic?

Limited Discrepancy Search (LDS)



Limited Discrepancy Search

William D. Harvey and **Matthew L. Ginsberg**
CIRL
1269 University of Oregon
Eugene, Oregon 97403
U.S.A.
ginsberg@cs.uoregon.edu

Abstract

Many problems of practical interest can be solved using tree search methods because carefully tuned successor ordering heuristics guide the search toward regions of the space that are likely to contain solutions. For some problems, the heuristics often lead directly to a solution—but not always. Limited discrepancy search addresses the problem of what to do when the heuristics fail.

Our intuition is that a failing heuristic might well have succeeded if it were not for a small number of "wrong turns" along the way. For a binary tree of height d, there are only d ways

the next section, we discuss existing algorithms. Limited discrepancy search (LDS) is introduced in Section 3 and compared to existing techniques in Section 4. We discuss variations of LDS that we believe will be useful for solving realistic problems in Section 5. We conclude by presenting our experimental results in Section 6.

2 Existing Strategies

Consider a tree search problem for which the successor ordering heuristic is so good that it almost always leads directly to a solution. Such problems are common both in practice and in areas of AI research such as planning and scheduling [Smith and Cheng, 1993; Wilkins, 1988]. If the heuristic is good enough, one might

Motivation for lds

ysis of mistakes provides an explanation [Harvey, 1995]. There is a reasonable chance that, somewhere early in 1-samp's path, it made a mistake by selecting a successor that had no goal nodes in the entire subtree below it. Once this early mistake is made and the successor's subtree is committed to, none of the subsequent decisions makes any difference.

If the subtree below a mistake is large, chronological backtracking will spend all of the allowed run time exploring the empty subtree, without ever returning to the last decision that actually matters. If one is counting on the heuristics to find a goal node in a small fraction of the search space, then chronological backtracking puts a tremendous burden on the heuristics early in the search and a relatively light burden on the heuristics deep in the search. Unfortunately, for many problems the heuristics are *least* reliable early in the search, before making decisions that reduce the problem to a size for which the heuristics become reliable. Because of the uneven reliance on the heuristics, it is unlikely that chronological backtracking is making the best use of the heuristic information.

3 Discrepancies

```
        if result ≠ NIL re
        return LDS-PRC

LDS(node)
    for x ← 0 to maximu
        result ← LDS-PR
        if result ≠ NIL re
    return NIL
```

Figure 1: Limited I

reaches *d*, the maximum dep searches the entire tree exh is guaranteed to find a goa guaranteed to terminate if t

Since each iteration of L ber of discrepancies to x inst to those nodes with exactly reexamines the nodes consid (see Figure 2). As with othe ever, the final iteration is fa sive and the redundancy is

Motivation for LDS

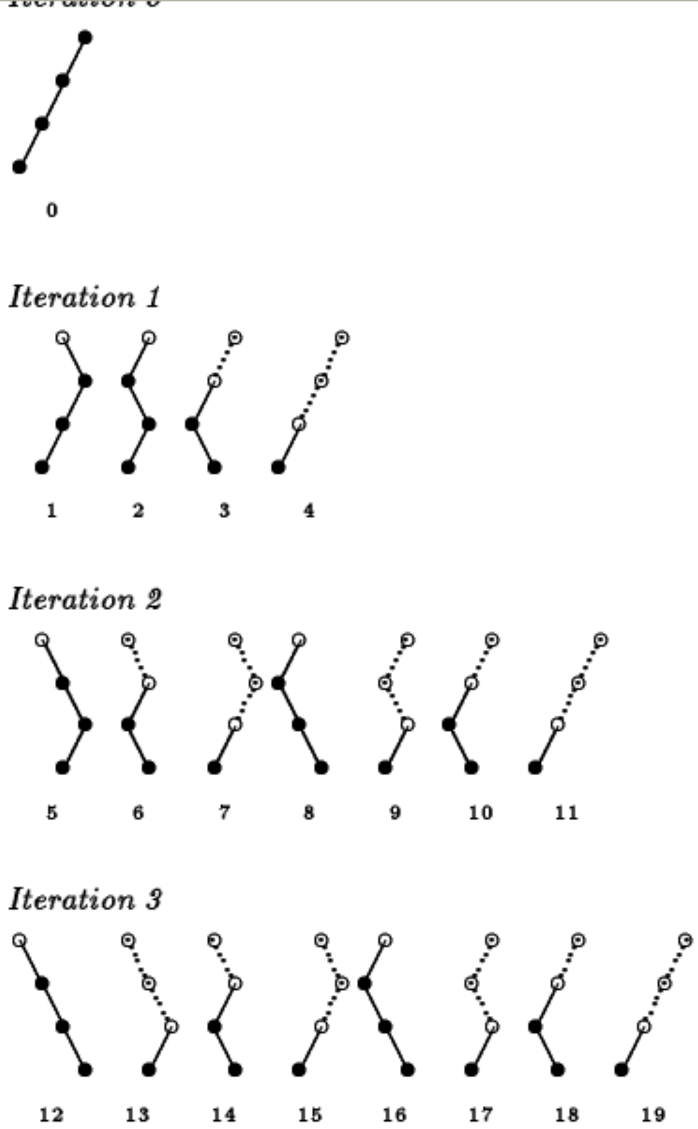


Figure 3: The four possibilities for a node and its children.

In order to simplify our analyses, we will assume that p is constant as well, although the experimental evidence is that p tends to increase somewhat as we search the tree because most heuristics are more accurate at deep nodes than at shallow ones.

The chance of finding a solution on a random path to depth d (i.e., using isamp) is simply $(1-m)^d$. Using heuristics and assuming a constant p, 1-samp has probability p^d of finding a solution on its one and only path.

This observation allows us to estimate p by running 1-samp on a large training set of problems from the domain of interest. Let s be the success rate of 1-samp on the training set. Since the probability of success for 1-samp is p^d, we have $p = s^{1/d}$. If s is small, the training set may have to be impractically large to get a reliable estimate. For some problems, though, s is not small. Heuristics developed for job shop scheduling have been shown to yield a probability s that is nearly one for small research problems [Smith and Cheng, 1993]. We have found in earlier experimental work on the same problems [Harvey, 1994]

Limited Discrepancy Search

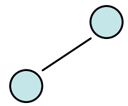
LDS

- show the search process
- assume binary branching
- assume we have 4 variables only
- assume variables have 2 values each

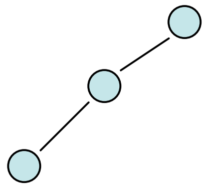
Limited Discrepancy Search (LDS)　　Ginsberg & Harvey

Take no discrepancies (go with the heuristic, go left!)

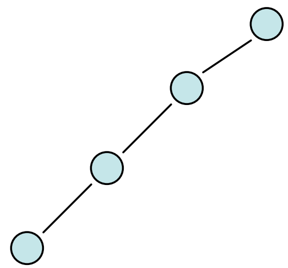
Take no discrepancies



Take no discrepancies

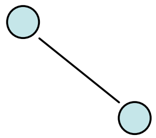


Take no discrepancies

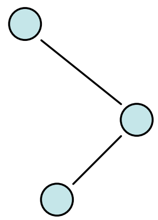


Take 1 discrepancy

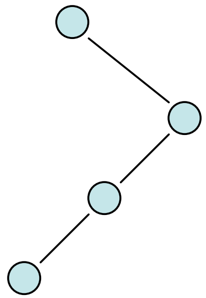
Take 1 discrepancy



Take 1 discrepancy



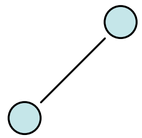
Take 1 discrepancy



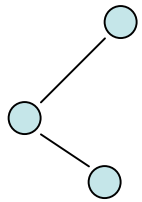
Limited Discrepancy Search (LDS) Ginsberg & Harvey

Take 1 discrepancy

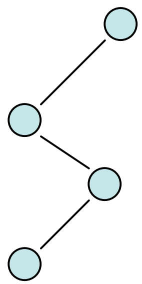
Take 1 discrepancy



Take 1 discrepancy



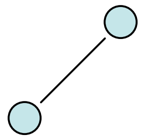
Take 1 discrepancy



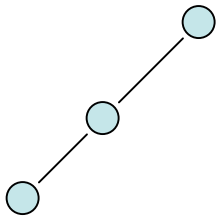
Limited Discrepancy Search (LDS) Ginsberg & Harvey

Take 1 discrepancy

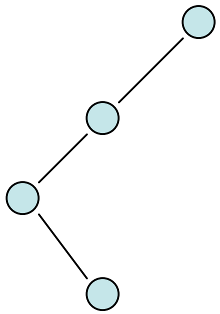
Take 1 discrepancy



Take 1 discrepancy

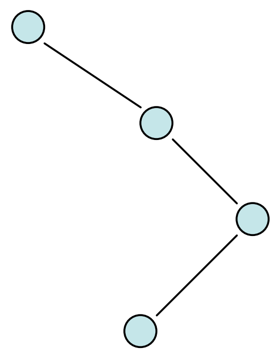


Take 1 discrepancy

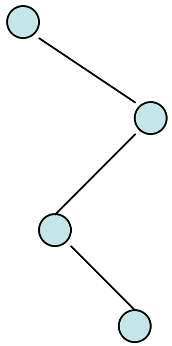


Now take 2 discrepancies

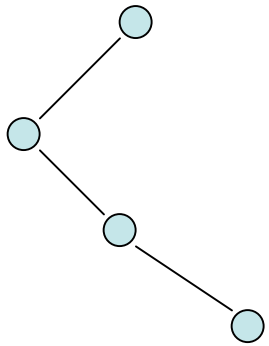
Take 2 discrepancies



Take 2 discrepancies



Take 2 discrepancies



First proposal

harvey95limited - GSview

File Edit Options View Orientation Media Help

be effective.[1]

ıply to backtrack
r experiments in
ıis approach pro-
elf, and the anal-
n [Harvey, 1995].
newhere early in
electing a succes-
subtree below it.
e successor's sub-
sequent decisions

ge, chronological
ved run time ex-
returning to the
ne is counting on
small fraction of

```
LDS-PROBE(node, k)
    if GOAL-P(node) return node
    s ← SUCCESSORS(node)
    if NULL-P(s) return NIL
    if k = 0 return LDS-PROBE(FIRST(s), 0)
    else
        result ← LDS-PROBE(SECOND(s), k − 1)
        if result ≠ NIL return result
        return LDS-PROBE(FIRST(s), k)
```

```
LDS(node)
    for x ← 0 to maximum depth
        result ← LDS-PROBE(node, x)
        if result ≠ NIL return result
    return NIL
```

For discrepancies 0 to n

Figure 1: Limited Discrepancy Search.

File: harvey95limited 568, 573pt Page: "2" 2 of 7

start papers Microsoft PowerPoint ... harvey95limited - GS... 12:27

harvey95limited - GSview

k is remaining discrepancies

First proposal

be effective.[1]

ıply to backtrack
r experiments in
ıis approach pro-
elf, and the anal-
n [Harvey, 1995].
newhere early in
electing a succes-
subtree below it.
e successor's sub-
sequent decisions

ge, chronological
ved run time ex-
returning to the
ne is counting on
small fraction of

```
LDS-PROBE(node, k)
    if GOAL-P(node) return node
    s ← SUCCESSORS(node)
    if NULL-P(s) return NIL
    if k = 0 return LDS-PROBE(FIRST(s), 0)
    else
        result ← LDS-PROBE(SECOND(s), k − 1)
        if result ≠ NIL return result
        return LDS-PROBE(FIRST(s), k)

LDS(node)
    for x ← 0 to maximum depth
        result ← LDS-PROBE(node, x)
        if result ≠ NIL return result
    return NIL
```

For discrepancies 0 to n

Figure 1: Limited Discrepancy Search.

File: harvey95limited 568, 573pt Page: "2" 2 of 7

harvey95limited - GSview

File

k is remaining discrepancies

First proposal

Go with heuristic

be effective.[1]

ıply to backtrack
r experiments in
ıis approach pro-
elf, and the anal-
n [Harvey, 1995].
newhere early in
electing a succes-
subtree below it.
e successor's sub-
sequent decisions

ge, chronological
ved run time ex-
returning to the
ne is counting on
small fraction of

```
LDS-PROBE(node, k)
    if GOAL-P(node) return node
    s ← SUCCESSORS(node)
    if NULL-P(s) return NIL
    if k = 0 return LDS-PROBE(FIRST(s), 0)
    else
        result ← LDS-PROBE(SECOND(s), k − 1)
        if result ≠ NIL return result
        return LDS-PROBE(FIRST(s), k)

LDS(node)
    for x ← 0 to maximum depth
        result ← LDS-PROBE(node, x)
        if result ≠ NIL return result
    return NIL
```

For discrepancies 0 to n

Figure 1: Limited Discrepancy Search.

File: harvey95limited 568, 573pt Page: "2" 2 of 7

start papers Microsoft PowerPoint ... harvey95limited - GS... 12:27

First proposal

harvey95limited - GSview

k is remaining discrepancies

be effective.[1]

…ply to backtrack
…r experiments in
…is approach pro-
…elf, and the anal-
…n [Harvey, 1995].
…newhere early in
…electing a succes-
…subtree below it.
…e successor's sub-
…sequent decisions

…ge, chronological
…ved run time ex-
…returning to the
…ne is counting on
…small fraction of

```
LDS-PROBE(node, k)
    if GOAL-P(node) return node
    s ← SUCCESSORS(node)
    if NULL-P(s) return NIL
    if k = 0 return LDS-PROBE(FIRST(s), 0)
    else
        result ← LDS-PROBE(SECOND(s), k − 1)
        if result ≠ NIL return result
        return LDS-PROBE(FIRST(s), k)

LDS(node)
    for x ← 0 to maximum depth
        result ← LDS-PROBE(node, x)
        if result ≠ NIL return result
    return NIL
```

Go with heuristic

Go against then go with

For discrepancies 0 to n

Figure 1: Limited Discrepancy Search.

File: harvey95limited 568, 573pt Page: "2" 2 of 7

The lds search process: how it goes (a cartoon)

harvey95limited - GSview

File Edit Options View Orientation Media Help

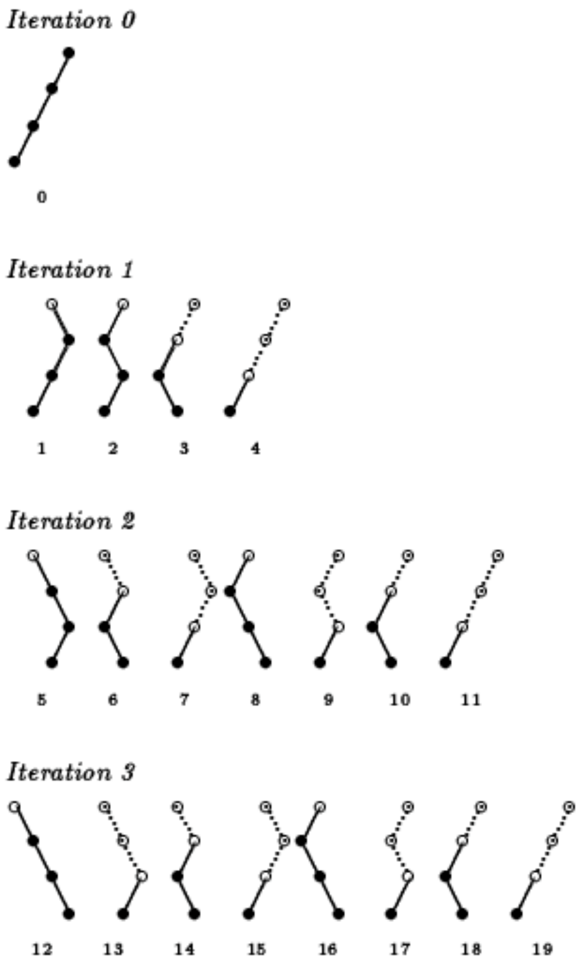


Figure 2: Execution trace of LDS.

assumed to be in the order of heuristic preference. We

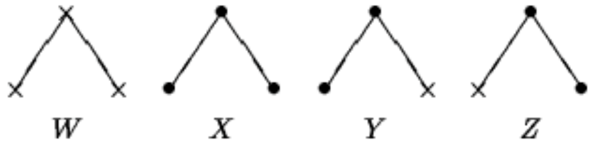


Figure 3: The four possibilities for a node and its children.

In order to simplify our analyses, we will assume that p is constant as well, although the experimental evidence is that p tends to increase somewhat as we search the tree because most heuristics are more accurate at deep nodes than at shallow ones.

The chance of finding a solution on a random path to depth d (i.e., using isamp) is simply $(1-m)^d$. Using heuristics and assuming a constant p, 1-samp has probability p^d of finding a solution on its one and only path.

This observation allows us to estimate p by running 1-samp on a large training set of problems from the domain of interest. Let s be the success rate of 1-samp on the training set. Since the probability of success for 1-samp is p^d, we have $p = s^{1/d}$. If s is small, the training set may have to be impractically large to get a reliable estimate. For some problems, though, s is not small. Heuristics developed for job shop scheduling have been shown to yield a probability s that is nearly one for small research problems [Smith and Cheng, 1993]. We have found in earlier experimental work on the same problems [Harvey, 1994] that even standard CSP heuristics can yield a success rate of about 75%. On larger scheduling problems [Vaessens *et al.*, 1994] the success rate of 1-samp is less, but more sophisticated heuristics from operations research keep 1-

File: harvey95limited 445, 601pt Page: "3" 3 of 7

start papers Microsoft PowerPoint ... harvey95limited - GS... 12:27

The lds search process: how it goes

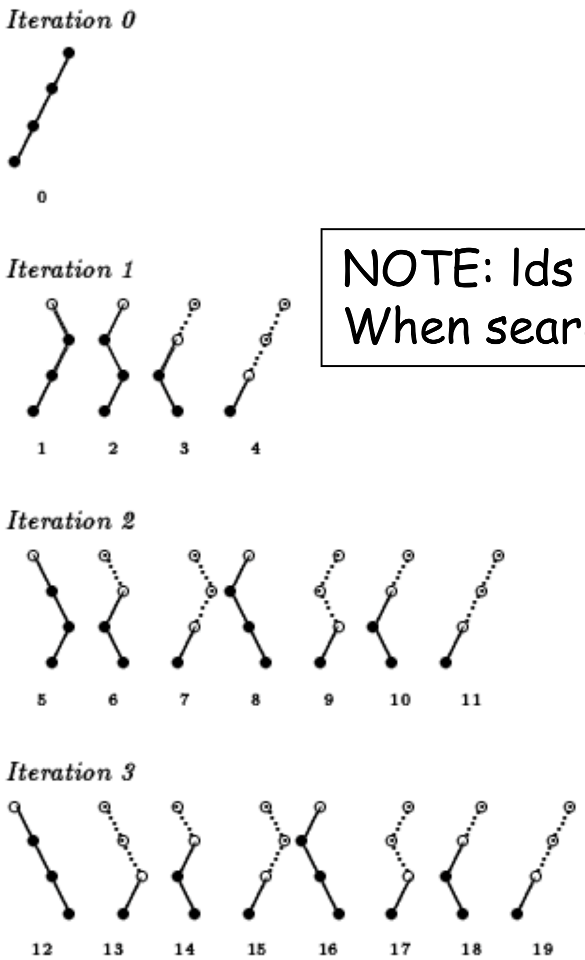
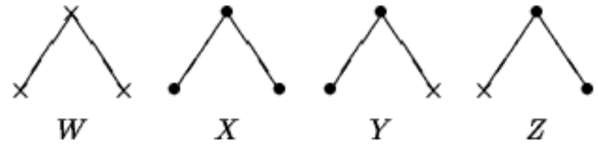


Figure 2: Execution trace of LDS.



NOTE: lds revisits search states with k discrepancies
When searching with > k discrepancies

is constant as well, although the experimental evidence is that p tends to increase somewhat as we search the tree because most heuristics are more accurate at deep nodes than at shallow ones.

The chance of finding a solution on a random path to depth d (i.e., using isamp) is simply $(1-m)^d$. Using heuristics and assuming a constant p, 1-samp has probability p^d of finding a solution on its one and only path.

This observation allows us to estimate p by running 1-samp on a large training set of problems from the domain of interest. Let s be the success rate of 1-samp on the training set. Since the probability of success for 1-samp is p^d, we have $p = s^{1/d}$. If s is small, the training set may have to be impractically large to get a reliable estimate. For some problems, though, s is not small. Heuristics developed for job shop scheduling have been shown to yield a probability s that is nearly one for small research problems [Smith and Cheng, 1993]. We have found in earlier experimental work on the same problems [Harvey, 1994] that even standard CSP heuristics can yield a success rate of about 75%. On larger scheduling problems [Vaessens *et al.*, 1994] the success rate of 1-samp is less, but more sophisticated heuristics from operations research keep 1-

File: harvey95limited 445, 601pt Page: "3" 3 of 7

My pseudo code


```
LDS(node)
for k := 0 to n
do begin
   result := LDS-Probe(node,k)
   if result != nil
   then return result
   end
return nil
```

```
LDS-Probe(node,k)
if isGoal(node) then return node
if failed(node) then return nil
if k = 0
then return LDS-Probe(left(node),0)
else begin
     result := LDS-Probe(right(node),k-1)
     if result = nil
     then result := LDS-PROBE(left(node),k)
     return result
     end
```

Fig. 1. Harvey and Ginsberg's limited discrepancy search (*LDS*)

lds revisits nodes: Korf's improvement (AAAI 96)

Improved Limited Discrepancy Search

Richard E. Korf
Computer Science Department
University of California, Los Angeles
Los Angeles, Ca. 90024
korf@cs.ucla.edu

July 26, 1995

Abstract

We present an improvement to Harvey and Ginsberg's limited discrepancy search algorithm. Our version eliminates much of the redundancy in the original algorithm, generating each search path from the root to the maximum search depth only once. For a uniform-depth binary tree of depth d, this reduces the asymptotic complexity from $O(\frac{d+2}{2}2^d)$ to $O(2^d)$. The savings is much less in a partial tree search,

korf96improved - GSview

File Edit Options View Orientation Media Help

Korf's improvement

Abstract

We present an improvement to Harvey and Ginsberg's limited discrepancy search algorithm. Our version eliminates much of the redundancy in the original algorithm, generating each search path from the root to the maximum search depth only once. For a uniform-depth binary tree of depth d, this reduces the asymptotic complexity from $O(\frac{d+2}{2}2^d)$ to $O(2^d)$. The savings is much less in a partial tree search, or in a heavily pruned tree. We also show that the overhead of the improved algorithm on a uniform-depth b-ary tree is only a factor of $b/(b-1)$ compared to depth-first search. This constant factor is greater on a heavily pruned tree. Finally, we present empirical results showing the utility of limited discrepancy search, as a function of problem difficulty, on the NP-Complete problem of number partitioning.

File: korf96improved 439, 238pt Page: "1" 1 of 14

start papers Microsoft PowerPoint ... korf96improved - GSv... 12:29

korf96improved - GSview

File Edit Options View Orientation Media Help

Korf's 1st mistake!

discrepancies in a binary tree of depth three. Figure 2 gives a pseudo-code description of a single iteration of LDS on a binary tree. Its arguments are a node, and the number of discrepancies k for that iteration. This function is called once for each iteration, with k ranging from zero to the maximum tree depth, terminating if a goal is found.

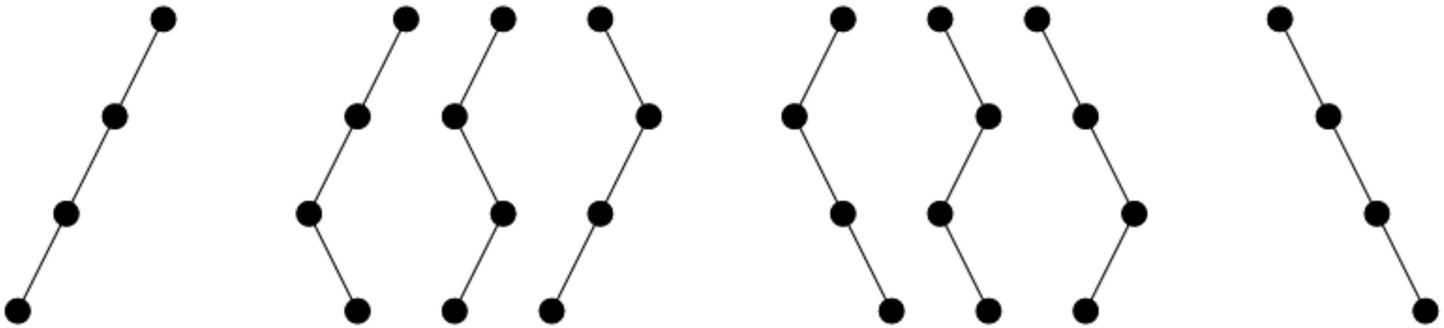


Figure 1: Paths with 0, 1, 2, and 3 Discrepancies in a Depth 3 Binary Tree

LDS can be applied to any tree-search problem where one bra

Woops!

each node is preferred to that of its siblings. The simplest extension to a non-binary tree is to treat any branch except the leftmost as a single discrepancy. In Harve

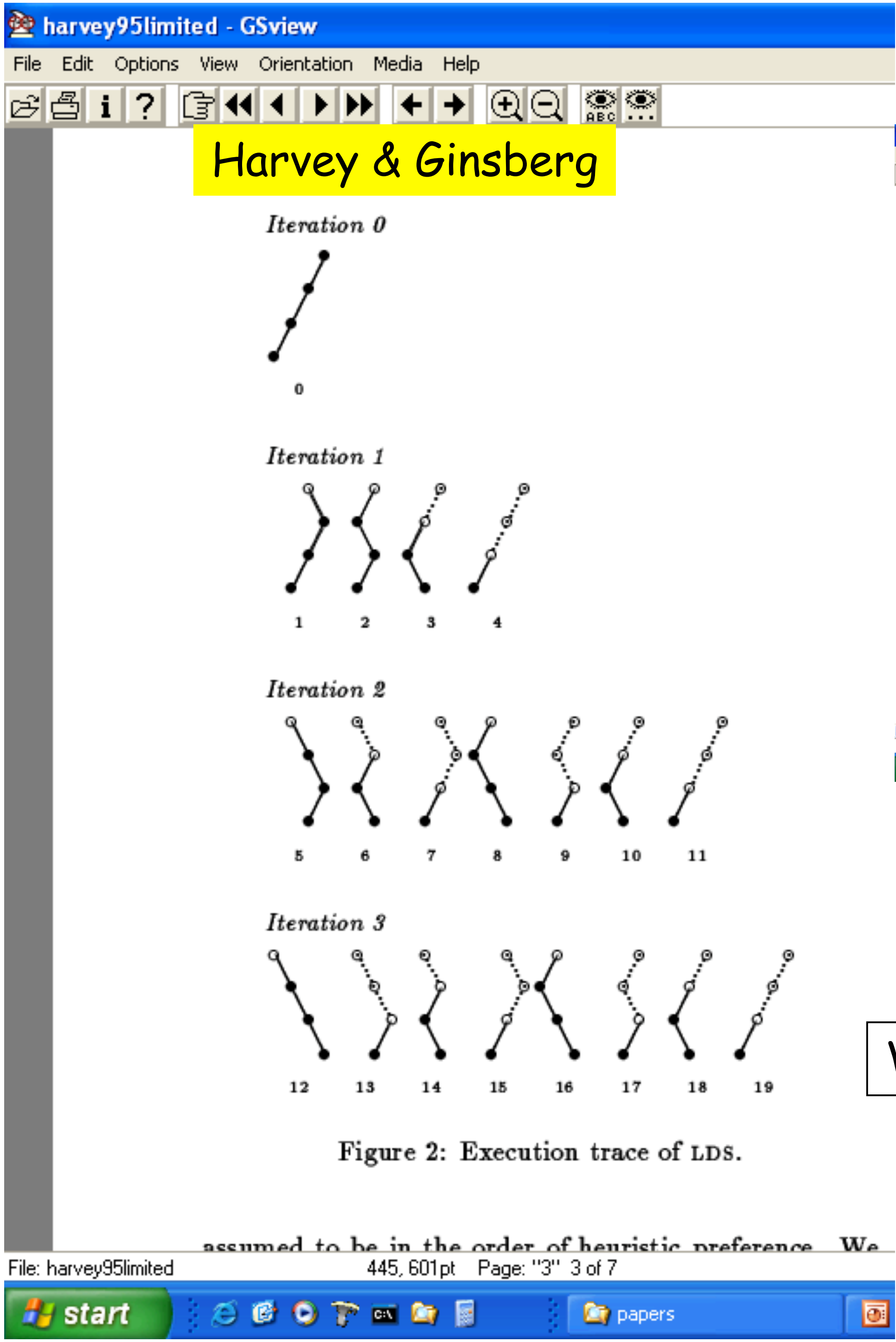
Do you see it? He's taking his discrepancies late/deep!

branch to have a higher probability of containing a solution in its subtree than the right branch, and show that limited discrepancy search has a higher

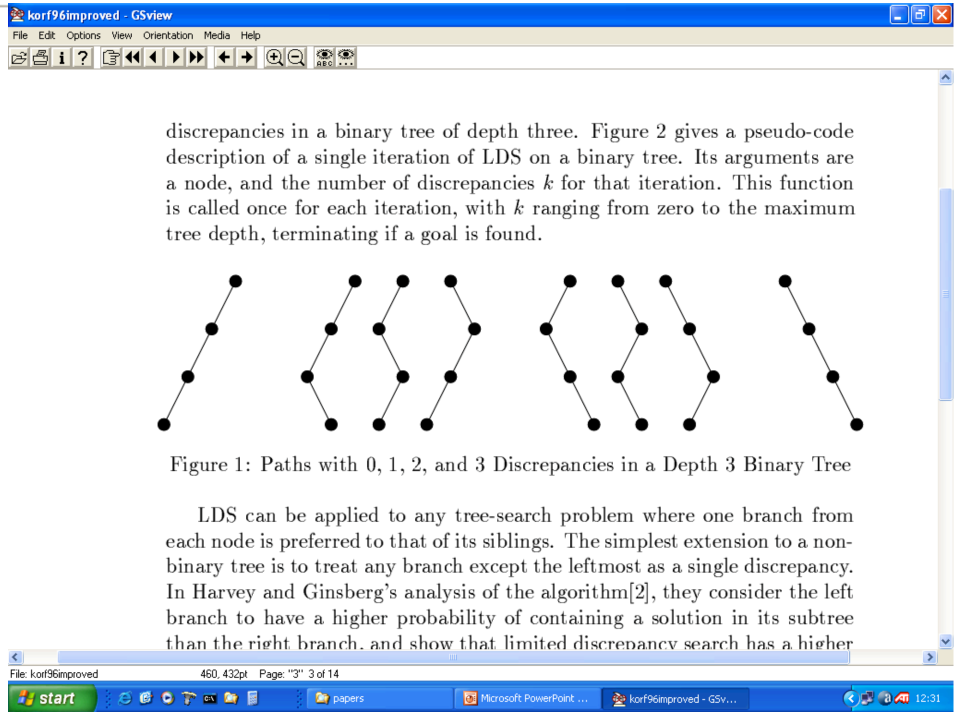
File: korf96improved 460, 432pt Page: "3" 3 of 14

start papers Microsoft PowerPoint ... korf96improved - GSv... 12:31

Harvey & Ginsberg



Korf



Wrong way round.

Is that important?

korf96improved - GSview

Korf's 1st mistake!

than the right branch, and show that limited discrepancy search has a higher probability of finding a solution than depth-first search, for a given number of node generations. They also show experimentally that it outperforms depth-first search in a constraint-satisfaction scheduling task.

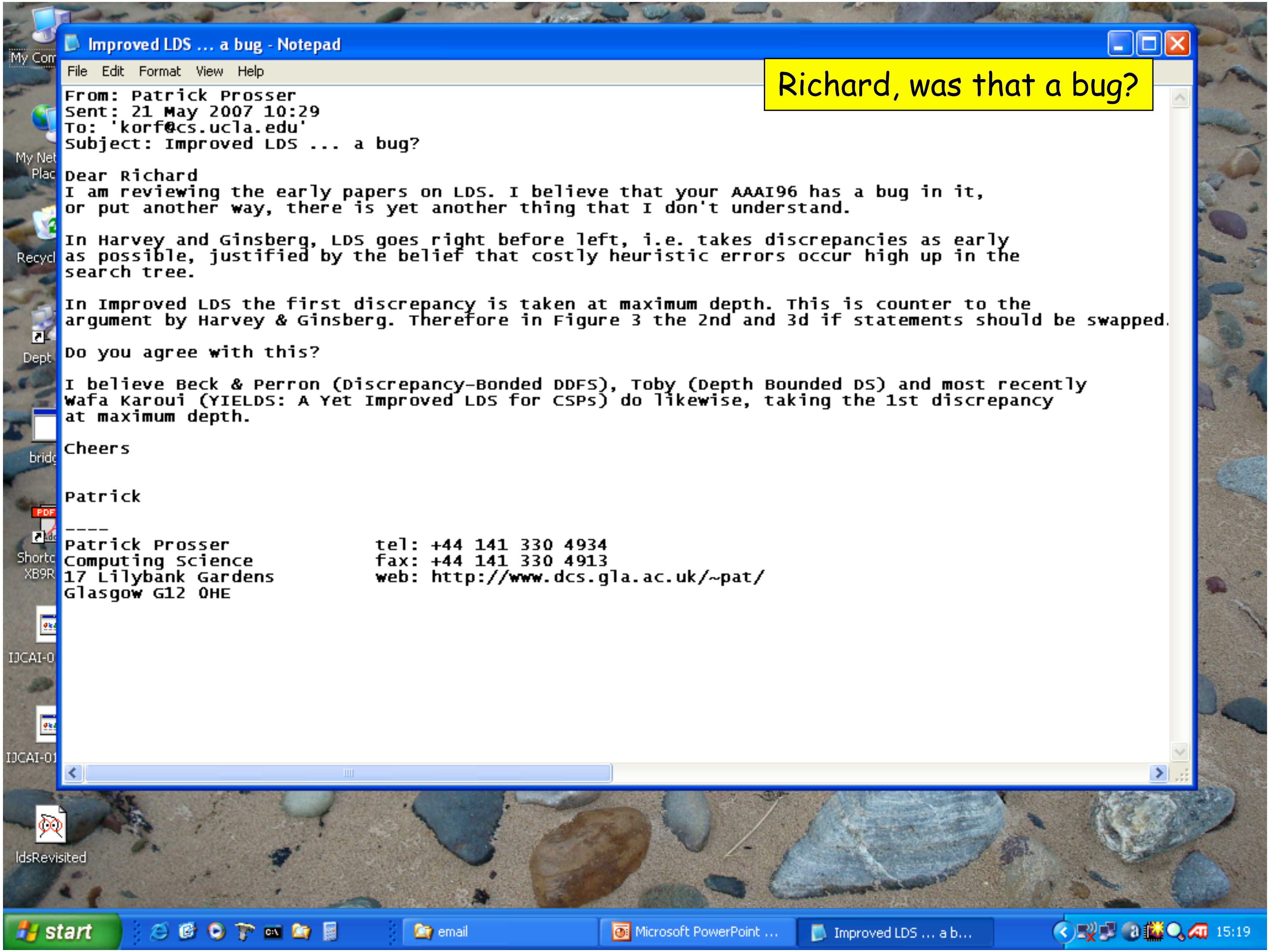
```
LDS (NODE, K)
   If NODE is a leaf, return
   LDS (left-child(NODE), K)
   If (K > 0) LDS (right-child(NODE), K-1)
```

Figure 2: Pseudo-code for a Single Iteration of Original LDS

Wrong way round Richard

File: korf96improved   376, 154pt   Page: "3" 3 of 14

Richard, was that a bug?



From: Patrick Prosser
Sent: 21 May 2007 10:29
To: 'korf@cs.ucla.edu'
Subject: Improved LDS ... a bug?

Dear Richard
I am reviewing the early papers on LDS. I believe that your AAAI96 has a bug in it, or put another way, there is yet another thing that I don't understand.

In Harvey and Ginsberg, LDS goes right before left, i.e. takes discrepancies as early as possible, justified by the belief that costly heuristic errors occur high up in the search tree.

In Improved LDS the first discrepancy is taken at maximum depth. This is counter to the argument by Harvey & Ginsberg. Therefore in Figure 3 the 2nd and 3d if statements should be swapped.

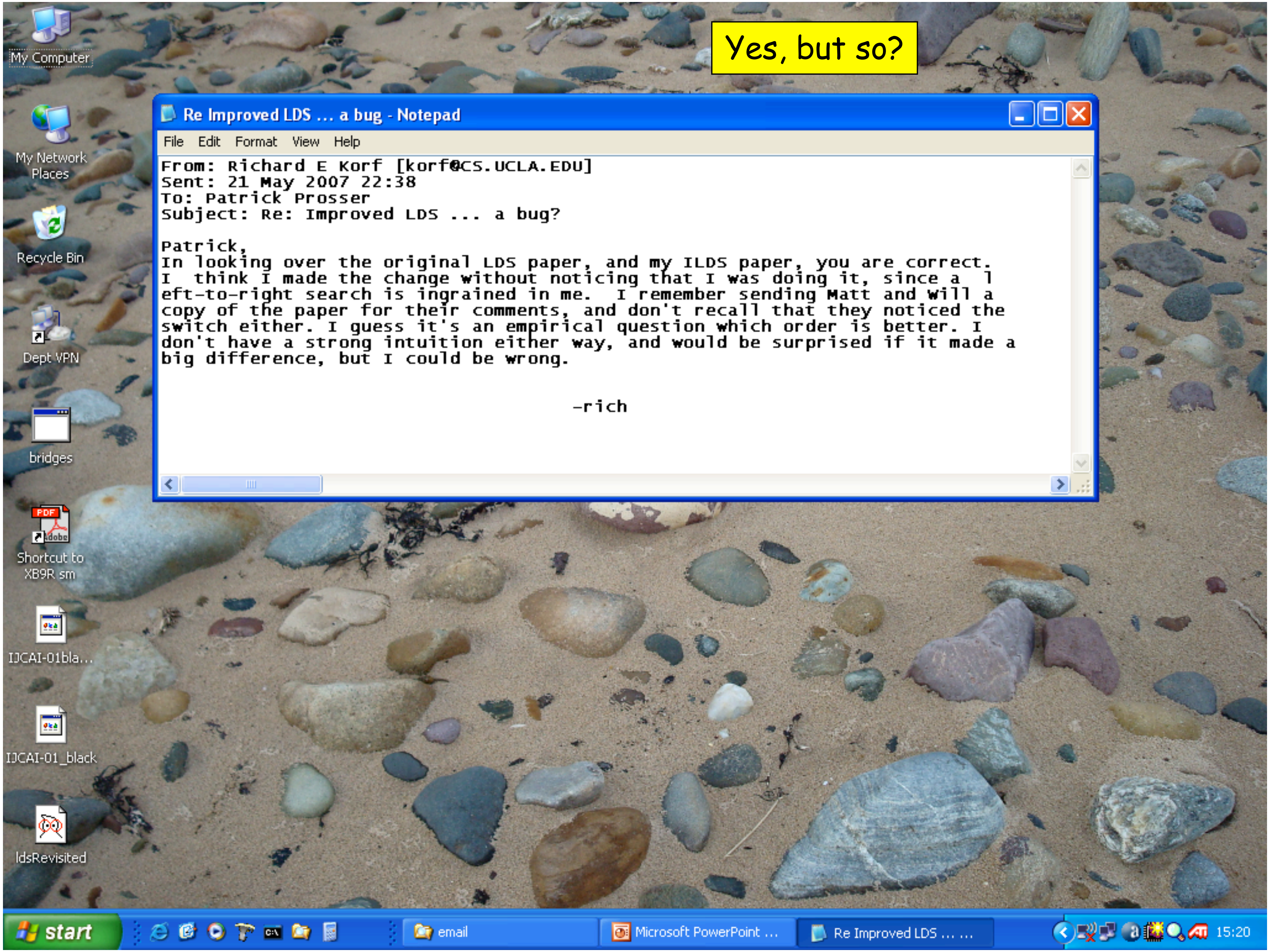
Do you agree with this?

I believe Beck & Perron (Discrepancy-Bonded DDFS), Toby (Depth Bounded DS) and most recently Wafa Karoui (YIELDS: A Yet Improved LDS for CSPs) do likewise, taking the 1st discrepancy at maximum depth.

Cheers

Patrick

Patrick Prosser
Computing Science
17 Lilybank Gardens
Glasgow G12 0HE
tel: +44 141 330 4934
fax: +44 141 330 4913
web: http://www.dcs.gla.ac.uk/~pat/

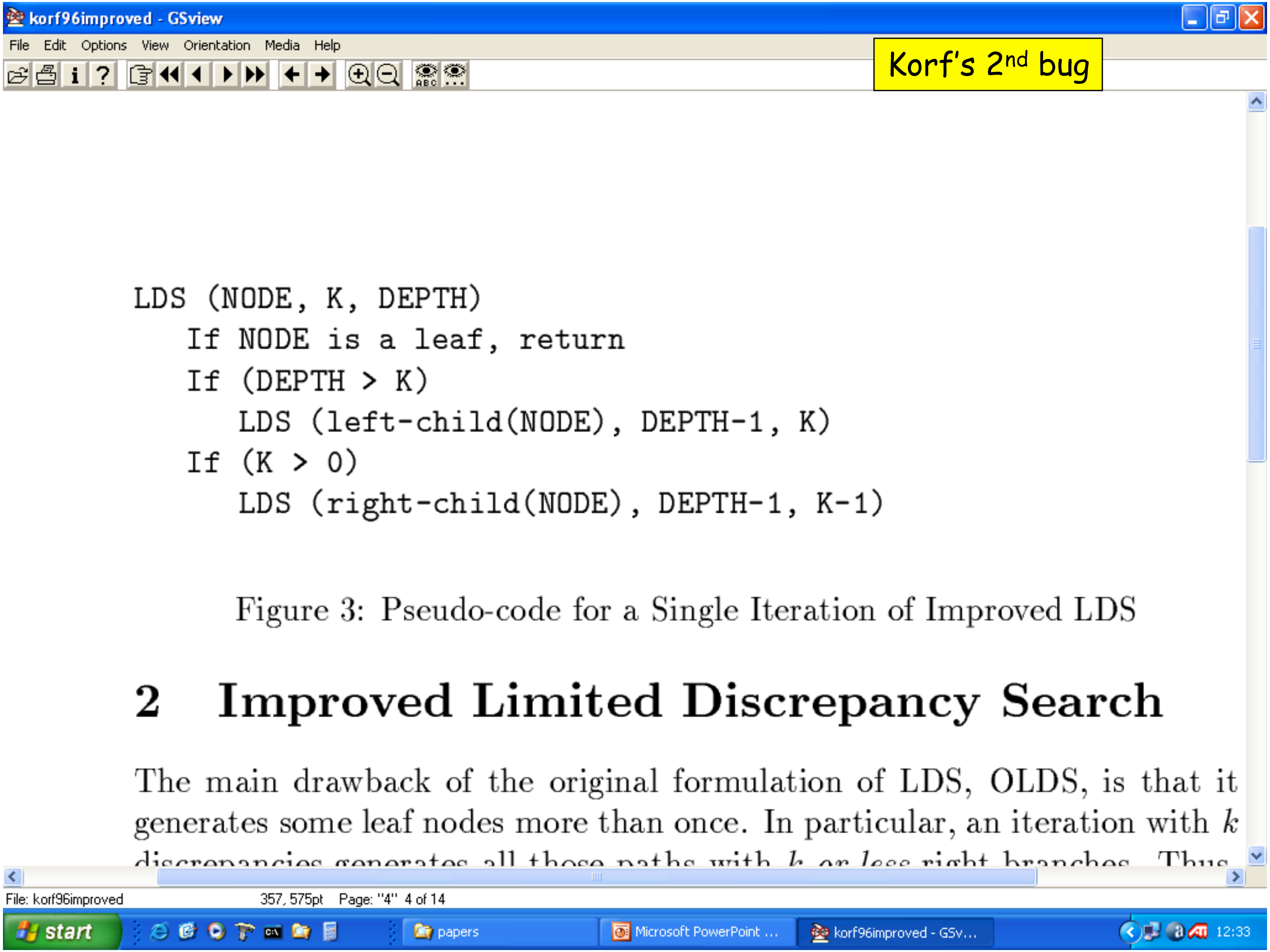


From: Richard E Korf [korf@CS.UCLA.EDU]
Sent: 21 May 2007 22:38
To: Patrick Prosser
Subject: Re: Improved LDS ... a bug?

Patrick,
In looking over the original LDS paper, and my ILDS paper, you are correct.
I think I made the change without noticing that I was doing it, since a l
eft-to-right search is ingrained in me. I remember sending Matt and Will a
copy of the paper for their comments, and don't recall that they noticed the
switch either. I guess it's an empirical question which order is better. I
don't have a strong intuition either way, and would be surprised if it made a
big difference, but I could be wrong.

-rich

Korf's 2nd bug



```
LDS (NODE, K, DEPTH)
   If NODE is a leaf, return
   If (DEPTH > K)
      LDS (left-child(NODE), DEPTH-1, K)
   If (K > 0)
      LDS (right-child(NODE), DEPTH-1, K-1)
```

Figure 3: Pseudo-code for a Single Iteration of Improved LDS

2 Improved Limited Discrepancy Search

The main drawback of the original formulation of LDS, OLDS, is that it generates some leaf nodes more than once. In particular, an iteration with k discrepancies generates all those paths with k or less right branches. Thus

korf96improved - GSview

File Edit Options View Orientation Media Help

Richard, you know there is another bug?

Woops!

```
LDS (NODE, K, DEPTH)
   If NODE is a leaf, return
   If (DEPTH > K)
      LDS (left-child(NODE), DEPTH-1, K)
   If (K > 0)
      LDS (right-child(NODE), DEPTH-1, K-1)
```

Figure 3: Pseudo-code for a Single Iteration of Improved LDS

2 Improved Limited Discrepancy Search

The main drawback of the original formulation of LDS, OLDS, is that it generates some leaf nodes more than once. In particular, an iteration with k discrepancies generates all those paths with k or less right branches. Thus

File: korf96improved 357, 575pt Page: "4" 4 of 14

start papers Microsoft PowerPoint ... korf96improved - GSv... 12:33

This redundancy was addressed by Korf's improved limited discrepancy search (ILDS)[6]. Pseudo-code for improved ILDS is as follows (see Fig 2).

My pseudo code

```
ILDS(node)
for k := 0 to n
do begin
   result := ILDS-Probe(node,k,n)
   if result != nil
   then return result
   end
return nil
```

```
ILDS-Probe(node,k,rDepth)
if isGoal(node) then return node
if failed(node) then return nil
result := nil
if rDepth > k
then result := ILDS-Probe(left(node),k,rDepth-1)
if k > 0 & result = nil
then result := ILDS-Probe(right(node),k-1,rDepth-1)
return result
```

Fig. 2. Korf's improved limited discrepancy search (ILDS)

Has anyone noticed Korf's bug?

Have people been using Korf's LDS?

Have people been using Harvey & Ginsberg's LDS?

Has anyone remembered the motivation for LDS?

limited discrepancy search - Google Scholar - Windows Internet Explorer

http://scholar.google.co.uk/scholar?q=limited+discrepancy+search&hl=en&lr=

Google Scholar BETA

limited discrepancy search [Search]

Advanced Scholar Search
Scholar Preferences
Scholar Help

Scholar **All articles** - **Recent articles**

Results **1** - **10** of about **283,000** for **limited discrepancy search**. (**0.28** seconds)

All Results

- W Harvey
- B Bonet
- M Ginsberg
- I SA
- H Geffner

[PS] **Limited discrepancy search** - all 7 versions »
WD Harvey, ML Ginsberg - Proceedings of the Fourteenth International Joint Conference ..., 1995 - cirl.uoregon.edu
Page 1. **Limited Discrepancy Search** William **... Limited discrepancy search** ad- dresses the problem of what to do when the heuristics fail. Our **...**
Cited by 331 - Related Articles - View as HTML - Web Search

Improved **limited discrepancy search** - all 3 versions »
RE Korf - The 1996 13 th National Conference on Artificial ..., 1996 - csa.com
Improved **limited discrepancy search**. Richard E Korf The 1996 13 th National Conference on Artificial Intelligence, AAAI 96. Part 1(of 2), 286-291, 1996. **...**
Cited by 68 - Related Articles - Web Search - Library Search

[PS] Depth-bounded **discrepancy search** - all 8 versions »
T Walsh - Proceedings of IJCAI, 1997 - dcs.st-and.ac.uk
... Recently, techniques like **limited discrepancy search** have been proposed for improving the chance of nding a goal in a **limited** amount of **search**. **...**
Cited by 109 - Related Articles - View as HTML - Web Search

Using Constraint Programming and Local **Search** Methods to Solve Vehicle Routing Problems - all 11 versions »
I SA - Principles and Practice of Constraint Programming--CP98: 4th ..., 1998 - books.google.com
... Unlike similar methods, we use **Limited Discrepancy Search** during the tree **search** to re-insert visits. **... Limited Discrepancy Search** In many cases. **...**
Cited by 216 - Related Articles - Web Search

[PS] A re-examination of **limited discrepancy search** - all 4 versions »
WK Jackson, M Irgens, WS Havens - Internet: http://fas. sfu. ca/isl/papers/jackson-lds/LDS. ... - cs.sfu.ca
Page 1. A Re-Examination of **Limited Discrepancy Search** ... This paper re- examines

Internet 100%

start Microsoft PowerPoint ... limited discrepancy se... 16:51

limited discrepancy search - Google Scholar - Windows Internet Explorer

http://scholar.google.co.uk/scholar?q=limited+discrepancy+search&hl=en&lr=

Cited by 6 - Related Articles - View as HTML - Web Search

[PS] Interleaved and **discrepancy** based **search** - all 6 versions »
P Meseguer, T Walsh - Proceedings of ECAI-98, 1998 - 4c.ucc.ie
... **Limited discrepancy search** (Lds) explores the leaf nodes in increasing order of the number of **discrepancies** taken to reach them [6]. On the kth iteration ...
Cited by 28 - Related Articles - View as HTML - Web Search

[PDF] **Discrepancy**-bounded depth first **search** - all 9 versions »
JC Beck, L Perron - Proceedings of the Second International Workshop on ..., 2000 - cs.sfu.ca
... In addition, it revisits fewer nodes than other **discrepancy**-based techniques (eg, **Limited Discrepancy Search**), making it a good candidate for proving optimal ...
Cited by 20 - Related Articles - View as HTML - Web Search

MACand Combined Heuristics: Two Reasons to Forsake FC(and CBJ?) on Hard Problems - all 2 versions »
C Bessiere, JC Regin - Principles and Practice of Constraint Programming, 1996 - citeseer.ist.psu.edu
... free **search** (context) - Freuder - 1982 127 Constraint satisfaction algorithms (context) - Nadel - 1989 127 **Limited discrepancy search** - Harvey, Ginsberg - 1995 ...
Cited by 161 - Related Articles - Cached - Web Search

[CITATION] **Limited Discrepancy Search**
ML Ginsberg, WD Harvey - Proceedings of the Fourteenth International Joint Conference ..., 1995
Cited by 6 - Related Articles - Web Search

Programming constraint inference engines - all 12 versions »
C Schulte - Proceedings of the Third International Conference on ..., 1997 - Springer
... inference engines ranging from standard **search** strategies to techniques new to constraint programming, including **limited discrepancy search**, visual **search**, and ...
Cited by 69 - Related Articles - Web Search

Gooooooooooogle ▶

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

limited discrepancy search - Google Scholar - Windows Internet Explorer

http://scholar.google.co.uk/scholar?q=limited+discrepancy+search&hl=en&lr=&start=30&sa=N

Google Scholar BETA

limited discrepancy search [Search]

Advanced Scholar Search
Scholar Preferences
Scholar Help

Scholar **All articles** - **Recent articles** Results **31** - **40** of about **283,000** for **limited discrepancy search**. (**0.24** seconds)

All Results

- W Harvey
- B Bonet
- M Ginsberg
- I SA
- H Geffner

Using regression-match graphs to control **search** in planning - all 7 versions »
D McDermott - Artificial Intelligence, 1999 - Elsevier
... Using regression-match graphs to control **search** in planning Drew McDermott 1 ... Keywords: Planning; **Search**; Means-ends analysis 1. Definition of the problem ...
Cited by 70 - Related Articles - Web Search

YIELDS: A Yet Improved **Limited Discrepancy Search** for CSPs* - all 2 versions »
W Karoui, MJ Huguet, P Lopez, W Naanaa - Proceedings CP-AI-OR-07, Brussels, Belgium, 2007 - Springer
... As indicated in its name, YIELDS is an improved version of **Limited Discrepancy Search** (LDS). ... YIELDS: A Yet Improved **Limited Discrepancy Search** for CSPs 101 ...
Cited by 1 - Related Articles - Web Search - BL Direct

[CITATION] **Limited Discrepancy Search**. In proc. IJCAI'95, 14th International Joint Conference on Artificial ...
W Harvey, M Ginsberg - 1995 - Montreal, Canada
Cited by 1 - Related Articles - Web Search

[CITATION] MLG, **Limited Discrepancy Search**
WD Harvey - Proceedings of the Fourteenth International Joint Conference ..., 1995
Cited by 1 - Related Articles - Web Search

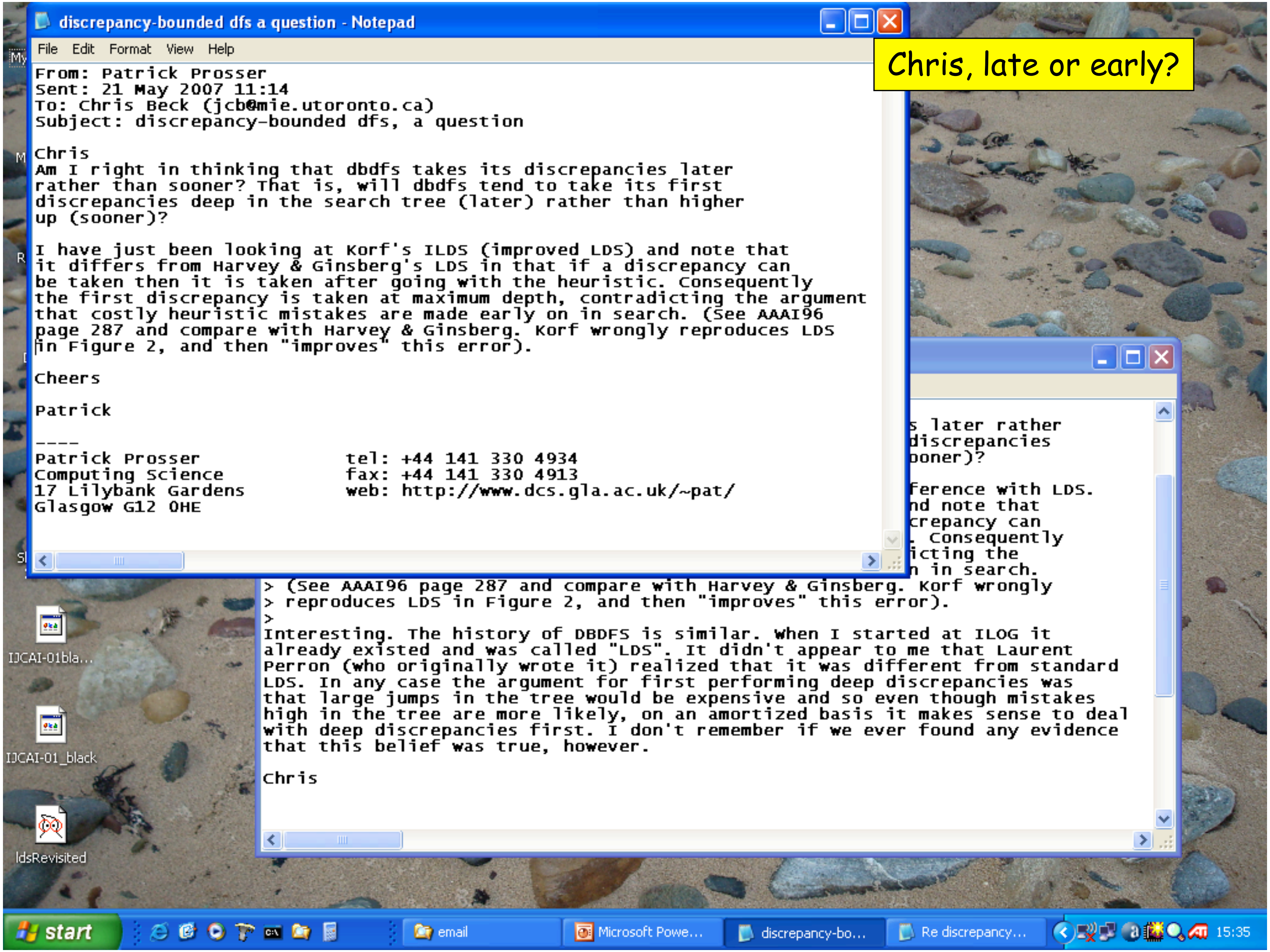
[CITATION] **Limited discrepancy search**. T. Dean, ed
W Harwey, M Ginsberg - Proc. 14th IJCAI, 1995
Cited by 1 - Related Articles - Web Search

[CITATION] **Limited discrepancy search**. Fourteenth International Joint Conference on Artificial Intelligence IJCAI ...
WD Harvey, ML Ginsberg - Morgan Kaufmann. citØ page (s)

Done Internet 100%

start Microsoft PowerPoint ... limited discrepancy se... 16:56

Chris, late or early?



discrepancy-bounded dfs a question - Notepad

From: Patrick Prosser
Sent: 21 May 2007 11:14
To: Chris Beck (jcb@mie.utoronto.ca)
Subject: discrepancy-bounded dfs, a question

Chris
Am I right in thinking that dbdfs takes its discrepancies later rather than sooner? That is, will dbdfs tend to take its first discrepancies deep in the search tree (later) rather than higher up (sooner)?

I have just been looking at Korf's ILDS (improved LDS) and note that it differs from Harvey & Ginsberg's LDS in that if a discrepancy can be taken then it is taken after going with the heuristic. Consequently the first discrepancy is taken at maximum depth, contradicting the argument that costly heuristic mistakes are made early on in search. (See AAAI96 page 287 and compare with Harvey & Ginsberg. Korf wrongly reproduces LDS in Figure 2, and then "improves" this error).

Cheers

Patrick

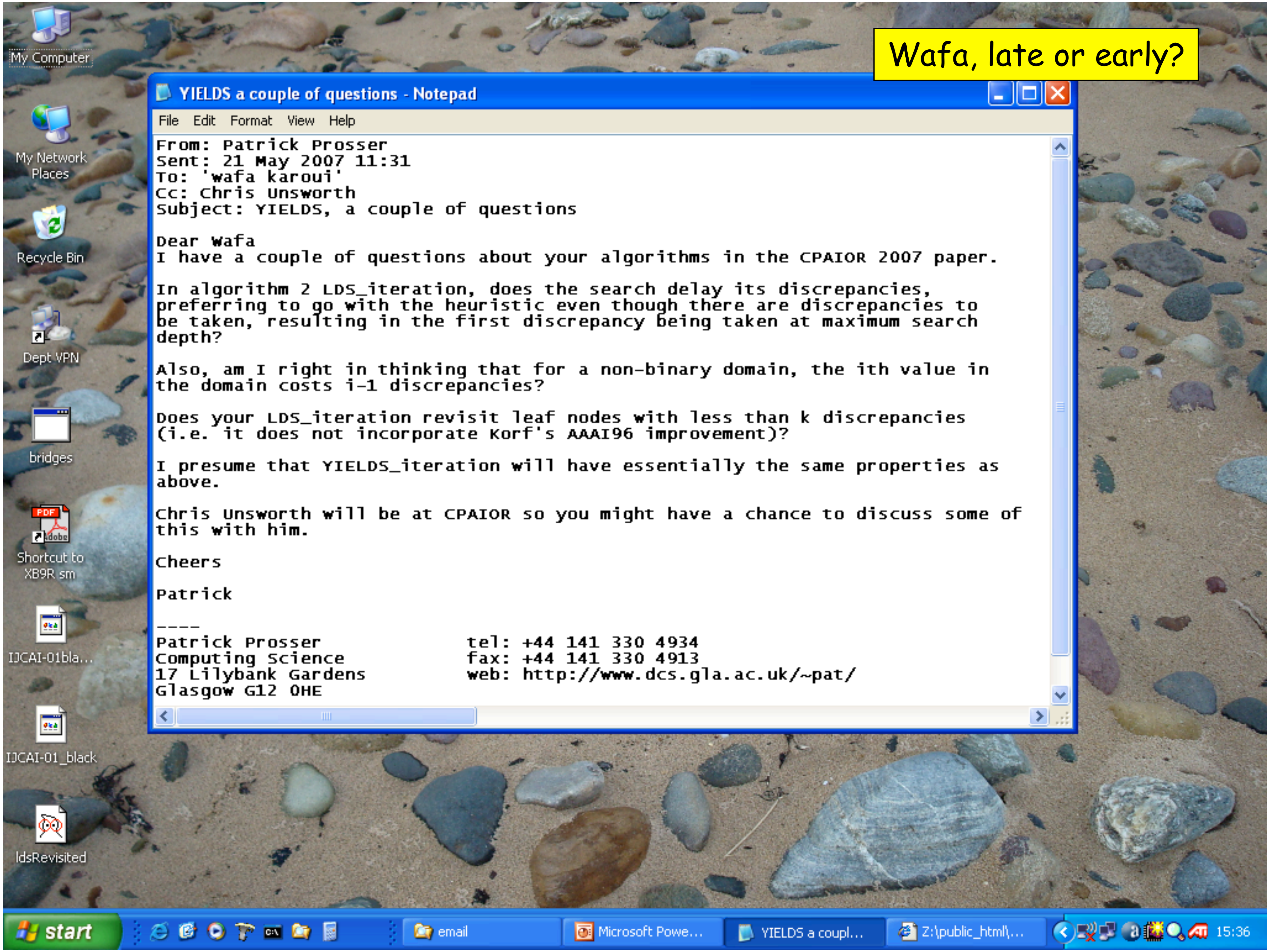
Patrick Prosser　　　　tel: +44 141 330 4934
Computing Science　　　fax: +44 141 330 4913
17 Lilybank Gardens　　web: http://www.dcs.gla.ac.uk/~pat/
Glasgow G12 0HE

s later rather
discrepancies
ooner)?

ference with LDS.
nd note that
crepancy can
. Consequently
icting the
n in search.
> (See AAAI96 page 287 and compare with Harvey & Ginsberg. Korf wrongly
> reproduces LDS in Figure 2, and then "improves" this error).
>
Interesting. The history of DBDFS is similar. When I started at ILOG it already existed and was called "LDS". It didn't appear to me that Laurent Perron (who originally wrote it) realized that it was different from standard LDS. In any case the argument for first performing deep discrepancies was that large jumps in the tree would be expensive and so even though mistakes high in the tree are more likely, on an amortized basis it makes sense to deal with deep discrepancies first. I don't remember if we ever found any evidence that this belief was true, however.

Chris

Wafa, late or early?



YIELDS a couple of questions - Notepad

From: Patrick Prosser
Sent: 21 May 2007 11:31
To: 'wafa karoui'
Cc: Chris Unsworth
Subject: YIELDS, a couple of questions

Dear Wafa
I have a couple of questions about your algorithms in the CPAIOR 2007 paper.

In algorithm 2 LDS_iteration, does the search delay its discrepancies, preferring to go with the heuristic even though there are discrepancies to be taken, resulting in the first discrepancy being taken at maximum search depth?

Also, am I right in thinking that for a non-binary domain, the ith value in the domain costs i-1 discrepancies?

Does your LDS_iteration revisit leaf nodes with less than k discrepancies (i.e. it does not incorporate Korf's AAAI96 improvement)?

I presume that YIELDS_iteration will have essentially the same properties as above.

Chris Unsworth will be at CPAIOR so you might have a chance to discuss some of this with him.

Cheers

Patrick

Patrick Prosser tel: +44 141 330 4934
Computing Science fax: +44 141 330 4913
17 Lilybank Gardens web: http://www.dcs.gla.ac.uk/~pat/
Glasgow G12 0HE

Z:\public_html\papers\ldsRevisited\email\Re YIELDS a couple of questions.htm - Microsoft Internet Explorer

Wafa's response

You are right, even though the original principle of LDS is to give priority to discrepancies in the top of the tree. In fact, I now have to study how to combine this way for exploring the tree search with my method.

>Also, am I right in thinking that for a non-binary domain, the ith value
>in the domain costs i-1 discrepancies?

Yes. It is a non-binary counting way.

>Does your LDS_iteration revisit leaf nodes with less than k
>discrepancies (i.e. it does not incorporate Korf's AAAI96 improvement)?

No. YIELDS really incorporates Korf's improvement (ILDS). However, since the variable order is supposed to change (because of weights), sometimes we are obliged to restart discrepancies.

>I presume that YIELDS_iteration will have essentially the same
>properties as above.

I think so, too. YIELDS is an improvement of LDS family which integrates learning in addition of all known improvements.

Best regards,

Wafa Karoui

IdsRevisited - GSview

File Edit Options View Orientation Media Help

I think this has not been reported

My pseudo code

```
ILDSN(node)
n = fd(node)
for k := 0 to n
do begin
   result := ILDSN-Probe(node,k,n)
   if result != nil
   then return result
   end
return nil
```

```
ILDSN-Probe(node,k,rDepth)
if isGoal(node) then return node
if failed(node) then return nil
result := nil
s := successors(node);
for i := 1 to min(card(s),k) while result = nil
do result := ILDSN-Probe(succ(i,s),k-i,rDepth-1)
if result = nil & fd(node) > k
then result := ILDSN-Probe(succ(0,s),k,rDepth-1)
return result
```

Fig. 3. Improved limited discrepancy search for non-binary domains (ILDSN)

File: IdsRevisited 462, 549pt Page: "5" 5 of 7

start Presentation1 IdsRevisited - GSview 12:09

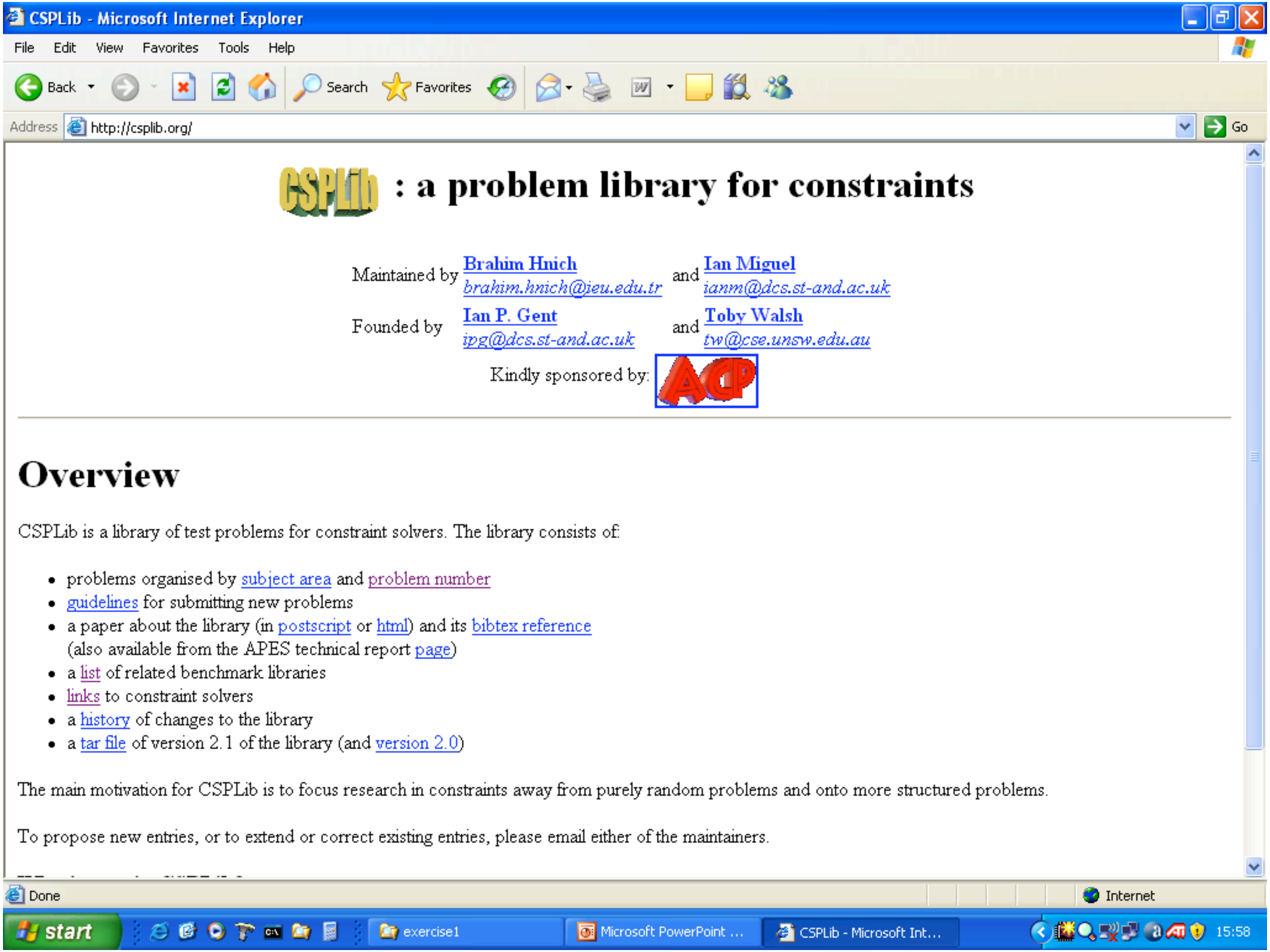
Does it make a difference if we take discrepancies late or early?

An empirical study

Tests Harvey & Ginsberg's motivation for LDS

Car Sequencing Problem

Assessed exercise 2


CSPLib - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://csplib.org/ Go
CSPLib : a problem library for constraints
Maintained by Brahim Hnich brahim.hnich@ieu.edu.tr and Ian Miguel ianm@dcs.st-and.ac.uk
Founded by Ian P. Gent ipg@dcs.st-and.ac.uk and Toby Walsh tw@cse.unsw.edu.au
Kindly sponsored by: ACP
Overview
CSPLib is a library of test problems for constraint solvers. The library consists of:
problems organised by subject area and problem number
guidelines for submitting new problems
a paper about the library (in postscript or html) and its bibtex reference
(also available from the APES technical report page)
a list of related benchmark libraries
links to constraint solvers
a history of changes to the library
a tar file of version 2.1 of the library (and version 2.0)
The main motivation for CSPLib is to focus research in constraints away from purely random problems and onto more structured problems.
To propose new entries, or to extend or correct existing entries, please email either of the maintainers.
Done
Internet
start
exercise1
Microsoft PowerPoint ...
CSPLib - Microsoft Int...
15:58

CSPLib - Microsoft Internet Explorer

http://csplib.org/

CSPLib : a problem library for constraints

Maintained by [Brahim Hnich](#) *brahim.hnich@ieu.edu.tr* and [Ian Miguel](#) *ianm@dcs.st-and.ac.uk*

Founded by [Ian P. Gent](#) *ipg@dcs.st-and.ac.uk* and [Toby Walsh](#) *tw@cse.unsw.edu.au*

Kindly sponsored by: ACP

Problems in Numerical Order:

prob001: Car sequencing.

prob002: Template design.

prob003: Quasigroup existence.

prob004: Mystery shopper.

prob005: Low autocorellation binary sequences.

prob006: Golomb rulers.

prob007: All-interval series.

prob008: Vessel loading.

CSPLib - Microsoft Internet Explorer

http://csplib.org/

prob001: car sequencing

proposed by Barbara Smith
bms@scs.leeds.ac.uk

Specification

A number of cars are to be produced; they are not identical, because different options are available as variants on the basic model. The assembly line has different stations which install the various options (air-conditioning, sun-roof, etc.). These stations have been designed to handle at most a certain percentage of the cars passing along the assembly line. Furthermore, the cars requiring a certain option must not be bunched together, otherwise the station will not be able to cope. Consequently, the cars must be arranged in a sequence so that the capacity of each station is never exceeded. For instance, if a particular station can only cope with at most half of the cars passing along the line, the sequence must be built so that at most 1 car in any 2 requires that option. The problem has been shown to be NP-complete (Gent 1999).

The format of the data files is as follows:

- First line: number of cars; number of options; number of classes.
- Second line: for each option, the maximum number of cars with that option in a block.
- Third line: for each option, the block size to which the maximum number refers.
- Then for each class: index no.; no. of cars in this class; for each option, whether or not this class requires it (1 or 0).

This is the example given in (Dincbas et al., ECAI88):

```
10 5 6
1 2 1 2 1
```

Done   Internet

start   exercise1   Microsoft PowerPoint ...   CSPLib - Microsoft Int...   16:00

CSPLib - Microsoft Internet Explorer

(Gent 1999).

The format of the data files is as follows:

- First line: number of cars; number of options; number of classes.
- Second line: for each option, the maximum number of cars with that option in a block.
- Third line: for each option, the block size to which the maximum number refers.
- Then for each class: index no.; no. of cars in this class; for each option, whether or not this class requires it (1 or 0).

This is the example given in (Dincbas et al., ECAI88):

```
10 5 6
1 2 1 2 1
2 3 3 5 5
0 1 1 0 1 1 0
1 1 0 0 0 1 0
2 2 0 1 0 0 1
3 2 0 1 0 1 0
4 2 1 0 1 0 0
5 2 1 1 0 0 0
```

A valid sequence for this set of cars is:

Class	Options req.
0	1 0 1 1 0
1	0 0 0 1 0
5	1 1 0 0 0
2	0 1 0 0 1
4	1 0 1 0 0
3	0 1 0 1 0
3	0 1 0 1 0
4	1 0 1 0 0

- Then for each class: index no., no. of cars in this class, for each option, whether or not this class requires it (1 or 0).

This is the example given in (Dincbas et al., ECAI88):

```
10 5 6
1 2 1 2 1
2 3 3 5 5
0 1 1 0 1 1 0
1 1 0 0 0 1 0
2 2 0 1 0 0 1
3 2 0 1 0 1 0
4 2 1 0 1 0 0
5 2 1 1 0 0 0
```

A valid sequence for this set of cars is:

Class	Options req.
0	1 0 1 1 0
1	0 0 0 1 0
5	1 1 0 0 0
2	0 1 0 0 1
4	1 0 1 0 0
3	0 1 0 1 0
3	0 1 0 1 0
4	1 0 1 0 0
2	0 1 0 0 1
5	1 1 0 0 0

Back to CSPLib home page.

Algorithms and Data Structures Group

university
- Vienna University of Technology
- Institute of Computer Graphics and Algorithms

ads
- ADS-Home
- Staff
- News
- Lehre
- Praktika und Diplomarbeiten
- Research
- Publications
- Contact
- Intranet

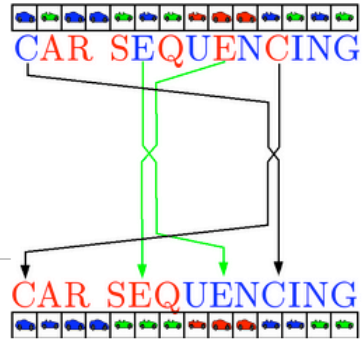
search

Search ADS

Car Sequencing

< Research

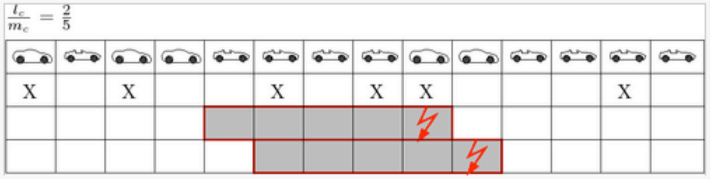
(G. Raidl, M. Prandtstetter, B. Hu)



Contents [hide]

Goal of Car Sequencing

Paintshop constraints with s=4



Assembly line constraints (X indicating that a car requires component c).

The goal of Car Sequencing is to find an optimal arrangement of commissioned vehicles along a production line. Although the individual cars are similar to each other, each automobile requires particular components which are assembled by different working bays along the production line. Further, each car has to be painted with exactly one color. The workload for each of these stations has to be smoothed to ensure a proper output. Workers with too much load get tired and make mistakes whereas underemployed workers only raise costs. The constraints defined by the various working bays have to be considered in the Car Sequencing Problem (CarSP). The demanded sequence of vehicles is a permutation of all cars to be produced at the day of question. In addition, the number of color changes has to be minimized.

The production line itself consists of three stages: the body shop, the paint shop, and the assembly shop. In the body shop the chassis of the cars are manufactured. The paint shop workers paint the cars, and in the assembly shop different options like air condition, sun roofs, or sound systems get installed. The constraints defined by the body shop and the assembly shop are similar to each other, whereas the paint shop constraints differ significantly. For the former, we consider restrictions which can be expressed as "No more than l_c cars are allowed to require component c in a sequence of m_c consecutive cars." For the latter, we consider those of the form: "At most s cars with the same color are allowed to be arranged consecutively." Changing the color after at most s cars is motivated by a more psychological reason: In automobile industry, the paint of a car is applied using an injector. During this spraying the paint slowly agglutinates. To obtain good results the injector has to be cleaned in regular intervals. If the same color would be applied after cleaning the injector again, the staff concerned with the cleaning process would get imprecise, because "It's the same color again." This would lead to improper painting results, which would in the following lead to reclamations.

Our work

In our work, we try to combine exact methods (e.g. based on integer linear programming) and meta-heuristics (e.g. Variable Neighborhood Search) for achieving good and compareable results in relatively short computation time (see also Hybridization of Exact and Heuristic Optimization Techniques).

Other Sources related to Car Sequencing


ROADEF 2005 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.ocea.li.univ-tours.fr/roadef05/
Go
> Accueil < | Généralités | Dates Importantes | Programme | Inscriptions | Liste des Inscrits
Soumission | Sponsors | Activités | Informations Pratiques | Contacts | Photos | Actes
ROADEF'2005
6ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision
14,15,16 Février 2005 à Tours
Laboratoire d'Informatique - Département Informatique
Polytech'Tours, 64 avenue Jean Portalis, 37200 Tours
tél : 02 47 36 14 14 - Fax : 02 47 36 14 22
http://www.polytech.univ-tours.fr
Site maintenu par Laure-Emmanuelle Drezet et Fabrice Tercinet
Internet
start
slides
Microsoft PowerPoint ...
ROADEF 2005 - Micro...
10:42

http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005/challenge2005_en.html - Microsoft Internet Explorer

ROADEF Challenge 2005

supported by

RENAULT CRÉATEUR D'AUTOMOBILES · RoaDeF · EURODECISION OPERATIONAL RESEARCH

(Updated on March 16th, 2005)

Version française de cette page

After the full success of the first three challenges ROADEF'99 , ROADEF'2001 and ROADEF2003, the French Society of Operations Research and Decision Analysis (ROADEF) decided to organize consecutively a fourth challenge. This time the results **will be presented at the annual conference of the French Operations Research society ROADEF which will be held on february 14-16, 2005 at Tours (closed to the famous Loire castles)**. The dates and the place of the conference will be announced later.

The subject of this fourth challenge is proposed and sponsored by the automobile manufacturer RENAULT.

This challenge is ***open*** to everybody, in particular to young researchers, except those working for the industrial partner having submitted the problem, or any of its subsidiaries.

1. Challenge subject (almost everthing is here !) (Updated on **March 16th, 2005: data instance set X is here !**)
2. How to participate and planning (Updated on February 17th, 2004)
3. Challenge prizes
4. List of the participants of the qualification stage (Updated on February 18th, 2004)

http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005/challenge2005_en.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Address http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005/challenge2005_en.html Go

Challenge subject sponsored by RENAULT

The problem retained for the ROADEF'2005 challenge is the **Car Sequencing problem**.

The detailed definition of the problem can be found below :

- **Files of the description of the problem in three formats** (Postscript, RTF, MSWord) **(ATTENTION: subject updated on October 12th, 2003)**
- Files of one example data instance for one scenario per class (i.e. a total of three scenari) provided in the zip format (instance_039_ch1_s26_mar.zip, instance_048_ch2_s25_mar.zip, instance_064_ch2_s24_mar.zip)
- **Files of the data instance set A in two formats** (Instances_set_A.tar.gz, Instances_set_A.zip) (created on October 10th, 2003)
- **Files of the data instance set B in two formats** (Instances_set_B.tar.gz, Instances_set_B.zip) (updated on July 15th, 2004)
- **Files of the data instance set X in two formats** (Instances_set_X.tar.gz, Instances_set_X.zip) (**updated on March 16th, 2005**)

Important remarks:

- It is **compulsory to respect the formats of the data and result files**.
- Solution checking tools (checkers.tar.gz or checkers.zip) are now available to help the candidates.
  - **New checkers with the new coefficients 1000000, 1000 and 1** (Windows: exeCarSeq.exe, Linux: exeCarSeq) **(created on July 15th, 2004)**
- This subject is provided by RENAULT for the **use in the context of this challenge and for research purpose**,
  RENAULT reserves the right to bring minor modifications to the subject if it is necessary.

Challenge prizes

French OR Society ROADEF associated to RENAULT and EURODECISION will present prizes to the finalists.

PRIZES provided by RENAULT :

Internet

start slides Microsoft PowerPoint ... http://www.prism.uv... 10:45

http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005/challenge2005_en.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Address http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005/challenge2005_en.html Go

Challenge prizes

French OR Society ROADEF associated to RENAULT and EURODECISION will present prizes to the finalists.

PRIZES provided by RENAULT :

- 3000 Euros for the Junior category (team compounded with **at least one student** and **at most one Senior researcher**).
- 3000 Euros for the Senior category.

French OR society **ROADEF** would support part of the travel expenses to the conference ROADEF'2005 for the students who reach the final stage.

PRIZES from other partner :

- **EURODECISION** company sponsors the second prize of the Junior category : 300 Euros.

Challenge organizing committee

- **Leader of this commitee (please to feel free to send me any comment on this challenge):**
  Van-Dat CUNG Van-Dat.Cung@gilco.inpg.fr
  Since September 1st, 2003
  GILCO lab., ENSGI-INPG
  46, avenue Félix-Viallet
  38031 Grenoble Cedex
  FRANCE

  Previously at PRiSM lab.- CNRS UMR 81 44
  Université de Versailles-St. Quentin en Yvelines
  45, avenue des Etats-Unis
  78035 Versailles Cedex

Internet

start slides Microsoft PowerPoint ... http://www.prism.uv... 10:45

Espace membres
Surnom/pseudo
Mot de passe
Retenir
Identification

Certaines rubriques et forums ne sont accessibles qu'aux adhérents. L'adhésion en ligne est possible:
Adhérer

Menu Principal
- Accueil
- L'association
- Parrainage

Modules
- FAQ
- Articles & Tutoriaux
- Téléchargements
- Liens internet
- Forums
- JFPC
- Séminaires
- Lettres de l'AFPC
- Nouvelles
- Compte rendus
- Sujets
- Recherche
- Statistiques
- Recommandation
- Top 10

Calendrier
<< February 2007 >>

M	T	W	T	F	S	S
29	30	31	01	02	03	04
05	06	07	08	09	10	11

CFP EJOR for a special issues on the Challenge ROADEF'2005 "Car Sequencing"

Transmis par: leberre actif 27 Sept 2005 @ 14:21

==

A Call for Papers EJOR for a special issues on the Challenge ROADEF'2005 "Car Sequencing"
is announced here below, and you are welcome to submit an article.

The related EJOR WEB site is
http://www.elsevier.com/wps/find/S03.cws_home/eor11

Please to note that the deadline for submission initially announced on October 30th, 2005
is asked to extend to Novemeber 27th, 2005; and to forward this message to colleagues who
would be interested.

CALL FOR PAPERS

Special Issue of The European Journal of Operational Research (EJOR)
on ROADEF Challenge 2005 on Car Sequencing

Brief description of the topic:

The European Journal of Operational Research (EJOR) will publish a
feature issue on the car sequencing problem proposed jointly by the RENAULT
company and the French Operations Research society (ROADEF) in the
context of the ROADEF Challenge 2005.

The ROADEF Challenge 2005 has involved 37 junior teams and 18 senior
teams and
was started on July 28, 2003. The final results were proclaimed
during the annual ROADEF conference which was held on february 14-16,
2005 at Tours.

The particular car sequencing problem considered in the challenge aims
at computing a
sequence of vehicles under hard and soft constraints issued from both
paint shop
and assembly line requirements of a RENAULT factory. The problem was
defined as a multiobjective optimization problem due to the presence of
soft constraints. For more information on the ROADEF challenge 2005,
including
participants, winners, problem description and instance sets visit
http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005/challenge2005_en.html

The special issue is open to all the participants to the ROADEF
Challenge 2005
and also to every original theoretical and/or experimental contribution
on this
problem. In particular, any submission of interest out of the challenge
scope
(lower bounds, polyedral analysis, exact methods or heuristics involving
commercial
solvers, Pareto set approximation, goal programming) is encouraged. The
results can
also be presented with no restriction on computational times, software
or hardware requirements.

Liens connexes
- Plus à propos de problemes de satisfaction de contraintes
- Info de leberre

La nouvelle la plus lue à propos de problemes de satisfaction de contraintes:
CFP MOPGP'06

My empirical study on car sequencing problems
Using various search algorithms, heuristics.

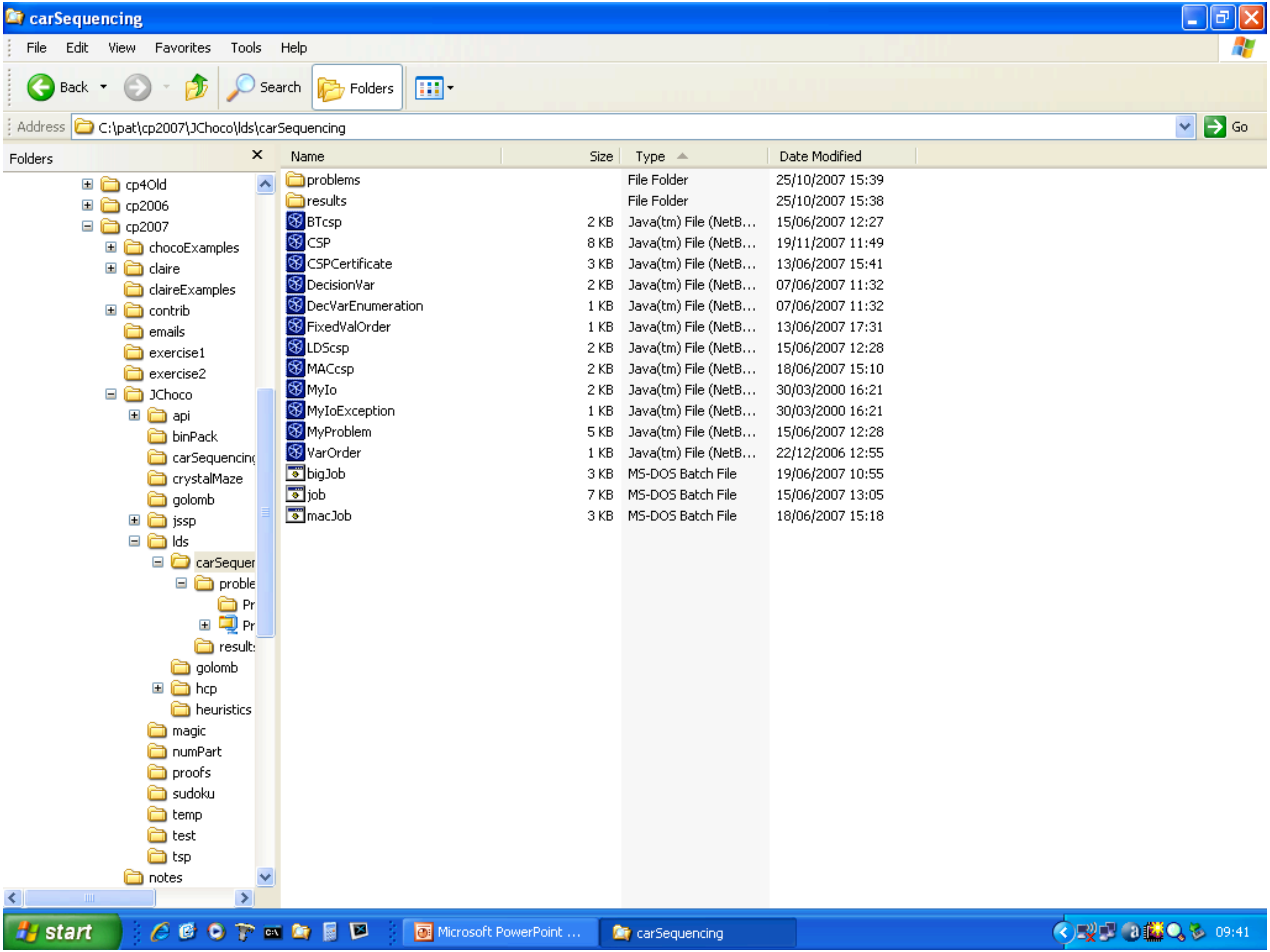
Question:

- does the order (late/early) that we take discrepancies in lds matter?
- is the order sensitive to the heuristics used?

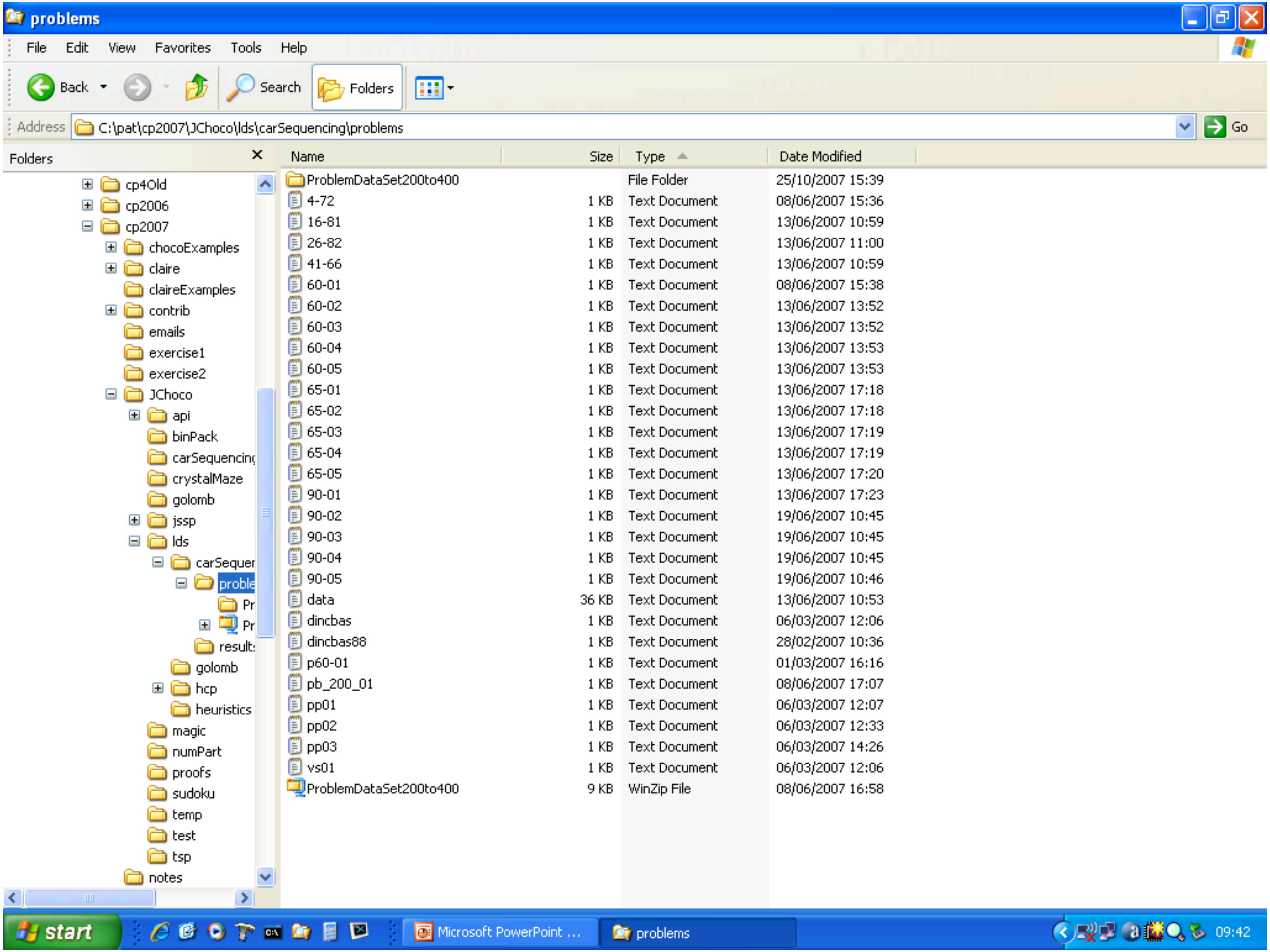
Performing the experiments (what's involved)

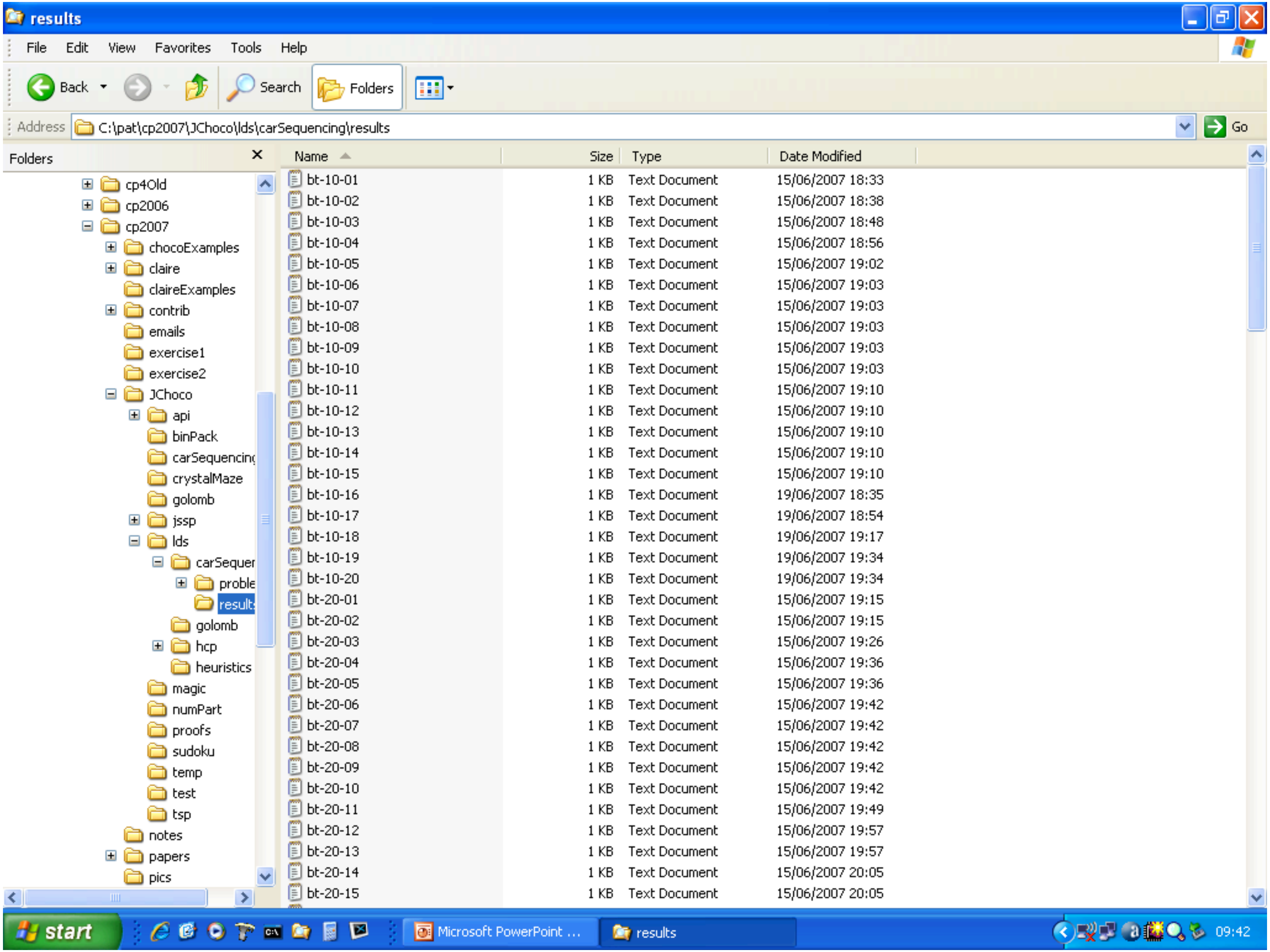
- code up lds in JChoco
  - for non-binary domains
  - paramaterised late/early discrepancies
  - using Korf's improvement
- code up model of car sequencing problem
  - using Pascal Van Hentenryk's model
- code up my BT (as a gold standard)
- code up a certificate checker
  - is a solution a solution?
- code up 4 different heuristics
  - 2 published heuristics for car sequencing
  - the 2 anti-heuristics
- Perform experiments on benchmark problems
  - limits on CPU time (minutes sometimes hours per instance)
  - test that all solutions are solutions (paranoia?)
  - problems typically have 200 cars (non-trivial)

- NOTE TO SELF
  - also did Golomb rulers
  - started on HC
  - did this to show results were general and not car seqn specific



carSequencing
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\pat\cp2007\JChoco\lds\carSequencing
Go
Folders
cp4Old
cp2006
cp2007
chocoExamples
claire
claireExamples
contrib
emails
exercise1
exercise2
JChoco
api
binPack
carSequencing
crystalMaze
golomb
jssp
lds
carSequen
proble
Pr
Pr
result:
golomb
hcp
heuristics
magic
numPart
proofs
sudoku
temp
test
tsp
notes
Name Size Type Date Modified
problems File Folder 25/10/2007 15:39
results File Folder 25/10/2007 15:38
BTcsp 2 KB Java(tm) File (NetB... 15/06/2007 12:27
CSP 8 KB Java(tm) File (NetB... 19/11/2007 11:49
CSPCertificate 3 KB Java(tm) File (NetB... 13/06/2007 15:41
DecisionVar 2 KB Java(tm) File (NetB... 07/06/2007 11:32
DecVarEnumeration 1 KB Java(tm) File (NetB... 07/06/2007 11:32
FixedValOrder 1 KB Java(tm) File (NetB... 13/06/2007 17:31
LDScsp 2 KB Java(tm) File (NetB... 15/06/2007 12:28
MACcsp 2 KB Java(tm) File (NetB... 18/06/2007 15:10
MyIo 2 KB Java(tm) File (NetB... 30/03/2000 16:21
MyIoException 1 KB Java(tm) File (NetB... 30/03/2000 16:21
MyProblem 5 KB Java(tm) File (NetB... 15/06/2007 12:28
VarOrder 1 KB Java(tm) File (NetB... 22/12/2006 12:55
bigJob 3 KB MS-DOS Batch File 19/06/2007 10:55
job 7 KB MS-DOS Batch File 15/06/2007 13:05
macJob 3 KB MS-DOS Batch File 18/06/2007 15:18
start
Microsoft PowerPoint ...
carSequencing
09:41


problems
File Edit View Favorites Tools Help
Back
Search
Folders
Address C:\pat\cp2007\JChoco\lds\carSequencing\problems
Go
Folders
cp4Old
cp2006
cp2007
chocoExamples
claire
claireExamples
contrib
emails
exercise1
exercise2
JChoco
api
binPack
carSequencing
crystalMaze
golomb
jssp
lds
carSequer
proble
Pr
Pr
result
golomb
hcp
heuristics
magic
numPart
proofs
sudoku
temp
test
tsp
notes
Name Size Type Date Modified
ProblemDataSet200to400 File Folder 25/10/2007 15:39
4-72 1 KB Text Document 08/06/2007 15:36
16-81 1 KB Text Document 13/06/2007 10:59
26-82 1 KB Text Document 13/06/2007 11:00
41-66 1 KB Text Document 13/06/2007 10:59
60-01 1 KB Text Document 08/06/2007 15:38
60-02 1 KB Text Document 13/06/2007 13:52
60-03 1 KB Text Document 13/06/2007 13:52
60-04 1 KB Text Document 13/06/2007 13:53
60-05 1 KB Text Document 13/06/2007 13:53
65-01 1 KB Text Document 13/06/2007 17:18
65-02 1 KB Text Document 13/06/2007 17:18
65-03 1 KB Text Document 13/06/2007 17:19
65-04 1 KB Text Document 13/06/2007 17:19
65-05 1 KB Text Document 13/06/2007 17:20
90-01 1 KB Text Document 13/06/2007 17:23
90-02 1 KB Text Document 19/06/2007 10:45
90-03 1 KB Text Document 19/06/2007 10:45
90-04 1 KB Text Document 19/06/2007 10:45
90-05 1 KB Text Document 19/06/2007 10:46
data 36 KB Text Document 13/06/2007 10:53
dincbas 1 KB Text Document 06/03/2007 12:06
dincbas88 1 KB Text Document 28/02/2007 10:36
p60-01 1 KB Text Document 01/03/2007 16:16
pb_200_01 1 KB Text Document 08/06/2007 17:07
pp01 1 KB Text Document 06/03/2007 12:07
pp02 1 KB Text Document 06/03/2007 12:33
pp03 1 KB Text Document 06/03/2007 14:26
vs01 1 KB Text Document 06/03/2007 12:06
ProblemDataSet200to400 9 KB WinZip File 08/06/2007 16:58
start
Microsoft PowerPoint ...
problems
09:42



results
File Edit View Favorites Tools Help
Back
Search
Folders
Address C:\pat\cp2007\JChoco\lds\carSequencing\results
Go
Folders
cp4Old
cp2006
cp2007
chocoExamples
claire
claireExamples
contrib
emails
exercise1
exercise2
JChoco
api
binPack
carSequencing
crystalMaze
golomb
jssp
lds
carSequer
proble
results
golomb
hcp
heuristics
magic
numPart
proofs
sudoku
temp
test
tsp
notes
papers
pics
Name Size Type Date Modified
bt-10-01 1 KB Text Document 15/06/2007 18:33
bt-10-02 1 KB Text Document 15/06/2007 18:38
bt-10-03 1 KB Text Document 15/06/2007 18:48
bt-10-04 1 KB Text Document 15/06/2007 18:56
bt-10-05 1 KB Text Document 15/06/2007 19:02
bt-10-06 1 KB Text Document 15/06/2007 19:03
bt-10-07 1 KB Text Document 15/06/2007 19:03
bt-10-08 1 KB Text Document 15/06/2007 19:03
bt-10-09 1 KB Text Document 15/06/2007 19:03
bt-10-10 1 KB Text Document 15/06/2007 19:03
bt-10-11 1 KB Text Document 15/06/2007 19:10
bt-10-12 1 KB Text Document 15/06/2007 19:10
bt-10-13 1 KB Text Document 15/06/2007 19:10
bt-10-14 1 KB Text Document 15/06/2007 19:10
bt-10-15 1 KB Text Document 15/06/2007 19:10
bt-10-16 1 KB Text Document 19/06/2007 18:35
bt-10-17 1 KB Text Document 19/06/2007 18:54
bt-10-18 1 KB Text Document 19/06/2007 19:17
bt-10-19 1 KB Text Document 19/06/2007 19:34
bt-10-20 1 KB Text Document 19/06/2007 19:34
bt-20-01 1 KB Text Document 15/06/2007 19:15
bt-20-02 1 KB Text Document 15/06/2007 19:15
bt-20-03 1 KB Text Document 15/06/2007 19:26
bt-20-04 1 KB Text Document 15/06/2007 19:36
bt-20-05 1 KB Text Document 15/06/2007 19:36
bt-20-06 1 KB Text Document 15/06/2007 19:42
bt-20-07 1 KB Text Document 15/06/2007 19:42
bt-20-08 1 KB Text Document 15/06/2007 19:42
bt-20-09 1 KB Text Document 15/06/2007 19:42
bt-20-10 1 KB Text Document 15/06/2007 19:42
bt-20-11 1 KB Text Document 15/06/2007 19:49
bt-20-12 1 KB Text Document 15/06/2007 19:57
bt-20-13 1 KB Text Document 15/06/2007 19:57
bt-20-14 1 KB Text Document 15/06/2007 20:05
bt-20-15 1 KB Text Document 15/06/2007 20:05
start
Microsoft PowerPoint ...
results
09:42

and now the results ...

	ldse1	ldsl1	ldse2	ldsl2	bt1	bt2	MAC1	MAC2
Dincbas 88	17/92	14/62	15/77	13 77	11/61	11/77	5/61	5/77
4-72	2659/4974	2417/2864	13660 28078	13188 19098	—	26522/29463	—	8737 30079
16-81	193459 368545	52519 95567	21666 44049	13415 25412	—	—	—	—
26-82	30354/54322	50986 66243	29845 62084	48974 69846	—	—	—	—
41-66	19222/26630	38821 51364	8530 13600	17081 28401	—	109/370	—	58 339
60-01	694/3943	26270 61549	291552 396144	—	39138/42693	296391/343563	~~39138~~/~~43879~~ 13249 ~~42693~~	102706 347244
60-02	200/570	200 570	200 570	200 570	200/555	200/570	166/524	188 539
60-03	492 2264	394 1602	2096 10335	1393 5946	218/878	203/862	184 863	175 832
60-04	36755 38766	5544 9041	30403 28016	15206 13414	313/770	215/724	202 647	164 663
60-05	270 1278	209 863	238 1048	239 1063	201/785	201/785	163 770	162 801
65-01	—	—	—	—	—	—	—	—
65-02	200/770	200/786	700 2787	380 1340	200/770	—	160/770	—
65-03	362 2280	390 1663	977 5991	1161 4836	346/1140	203/940	223 1078	173 878
65-04	2480 6161	14811 17204	—	—	766/1340	—	343 1478	—
65-05	210 924	209 877	1137 3727	9673 10535	201/770	205/801	159 771	162 786
90-01	—	—	39727 121704	13497 37780	—	—	—	—

Well, did you see a pattern?

If there is no pattern what does this say about H&G's hypothesis?

And, if no pattern, why is lds any good?

See anything?

	lds e1	lds l1	lds e2	lds l2	bt 1	bt 2	MAC1	MAC2
Dinbas 88	17/92	14/62	15/77	13 77	11/61	11/77	5/61	5/77
4-72	2659/4974	2417/2864	13660 28078	13188 19098	–	26522/29463	–	8737 30079
16-81	193459 368545	52519 95567	21666 44049	13415 25412	–	–	–	–
26-82	30354/54322	50986 66243	29845 62084	48974 69846	–	–	–	–
41-66	19222/26630	38821 51364	8530 13600	17081 28401	–	109/370	–	58 339
60-01	694/3943	26270 61549	291552 396144	–	39138/42693	296391/343563	~~39138~~/~~43879~~ 13249/~~42693~~	102706 347244
60-02	200/570	200 570	200 570	200 570	200/555	200/570	166/524	188 539
60-03	492 2264	394 1602	2096 10335	1393 5946	218/878	203/862	184 863	175 832
60-04	36755 38766	5544 9041	30403 28016	15206 13414	313/770	215/724	202 647	164 663
60-05	270 1278	209 863	238 1048	239 1063	201/785	201/785	163 770	162 801
65-01	–	–	–	–	–	–	–	–
65-02	200/770	200/786	700 2787	380 1340	200/770	–	160/770	–
65-03	362 2280	390 1663	977 5991	1161 4836	346/1140	203/940	223 1078	173 878
65-04	2480 6161	14811 17204	–	–	766/1340	–	343 1478	–
65-05	210 924	209 877	1137 3727	9673 10535	201/770	205/801	159 771	162 786
90-01	–	–	39727 121704	13497 37780	–	–	–	–

Got my act together for ECAI08 reject

o deep in search, consequently taking discrepan-
an advantage. In the second set of experiments,
it (HC), we use dynamic heuristics, recomputed
nsequently heuristic information should be more
arch where the remaining subproblem is small and
has taken place via constraint propagation. In this
pect that taking discrepancies early should be the
HC experiments are then repeated, this time us-
and value ordering heuristics, and we should then
, i.e. late discrepancies should be an advantage.

encing

ing problem we are to manufacture a number of
asses. Each class of car requires a number of op-
bly line there are stations that install options, and
capacity, usually referred to as a p/q constraint.
nce of length q_i at most p_i cars can have option i.
n to sequence the cars such that the demand is met
ty constraints are respected.
as coded in JChoco using the model due to Van
he decision variables correspond to the positions
nd values correspond to the possible classes of
manufactured in a position. Two static value or-
were used. Heuristic H1 orders values in non-
f the number of options required for the corre-
is is a heuristic used by Barbara Smith in one of
able and value ordering [15]. The second heuris-
es in non-increasing order of the option load re-

solution is found in that bound we have a table entry of -. The figures in brackets are for search using heuristic H1.

problem	ildsn-e		ildsn-l		bt	
db88	15	(17)	13	(14)	11	(11)
4-72	13660	(2659)	13188	(2417)	26522	(-)
16-81	21666	(193459)	**3415**	**(52519)**	-	(-)
26-82	**29845**	**(30354)**	48974	(50986)	-	(-)
41-66	**8530**	**(19222)**	17081	(38821)	109	(-)
60-01	**291552**	**(694)**	-	(26270)	296391	(39138)
60-02	200	(200)	200	(200)	200	(200)
60-03	2096	(492)	**1393**	**(394)**	203	(218)
60-04	30403	(36755)	**15206**	**(5544)**	215	(313)
60-05	238	(270)	239	(209)	201	(201)
65-01	-	(-)	-	(-)	-	(-)
65-02	700	(200)	**380**	(200)	-	(200)
65-03	**977**	**(362)**	1161	(390)	203	(346)
65-04	-	**(2480)**	-	**(14811)**	-	(766)
65-05	**1137**	(210)	9673	(209)	205	(201)
90-01	39727	(-)	**13497**	(-)	-	(-)
90-02	-	(-)	-	(-)	-	(-)
90-03	6676	(-)	**2455**	(-)	203	(-)
90-04	-	(-)	-	(-)	-	(-)
90-05	30297	(-)	**14897**	(-)	205	(231)

Table 1. The number of nodes explored to solve an instance of the car sequencing problem. First column is ILDSN taking discrepacies early, 2nd column ILDSN with late dicrepancies, and 3d column for chronological backtracking. An entry of - corresponds to 500000 nodes explored and no solution found. Variables are statically ordered, by index. Static value ordering is H2 by default (and H1 in brackets). Best results for ILDSN with H1 and H2 are given in bold.

ecai08 rejects

How about another problem domain?

Hamiltonian Circuit

WIKIPEDIA
The Free Encyclopedia

navigation
- Main Page
- Contents
- Featured content
- Current events
- Random article

interaction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

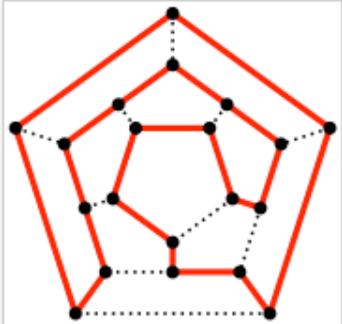
search
Go Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this page

Hamiltonian path

From Wikipedia, the free encyclopedia

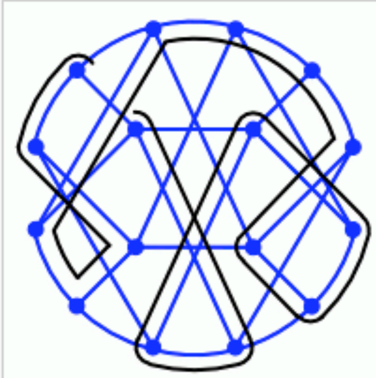
In the mathematical field of graph theory, a **Hamiltonian path** is a path in an undirected graph which visits each vertex exactly once. A **Hamiltonian cycle** (or **Hamiltonian circuit**) is a cycle in an undirected graph which visits each vertex exactly once and also returns to the starting vertex. Determining whether such paths and cycles exist in graphs is the Hamiltonian path problem which is NP-complete.



A Hamiltonian cycle in a dodecahedron. Like all platonic solids, the dodecahedron is Hamiltonian.

Hamiltonian paths and cycles are named after William Rowan Hamilton who invented the Icosian Game, now also known as *Hamilton's puzzle*, which involves finding a Hamiltonian cycle in the edge graph of the dodecahedron. Hamilton solved this problem using the Icosian Calculus, an algebraic structure based on roots of unity with many similarities to the quaternions (also invented by Hamilton). Unfortunately, this solution does not generalize to arbitrary graphs.

Contents [hide]



A Hamiltonian path (black) over a graph (blue).

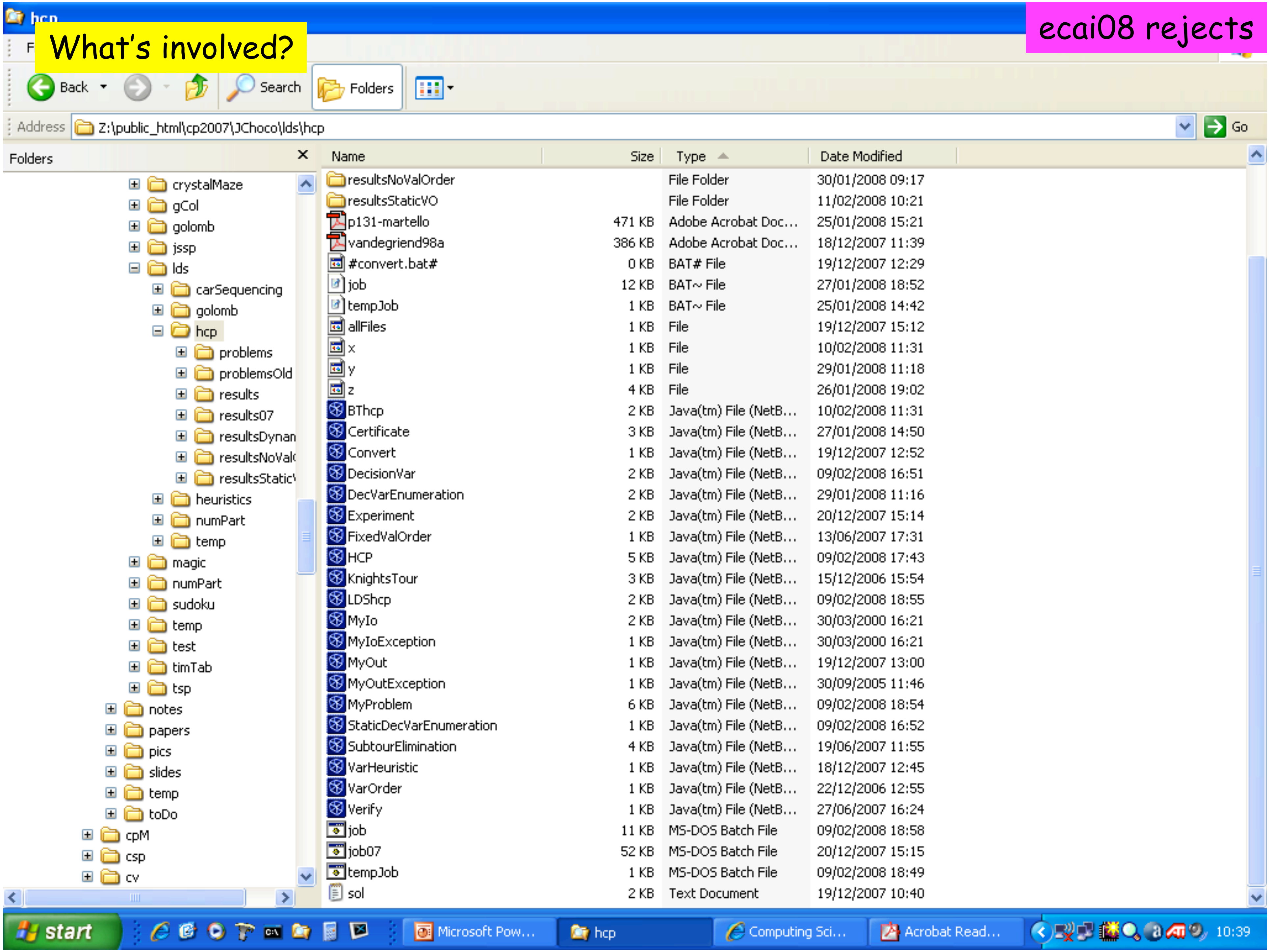
Definitions [edit]

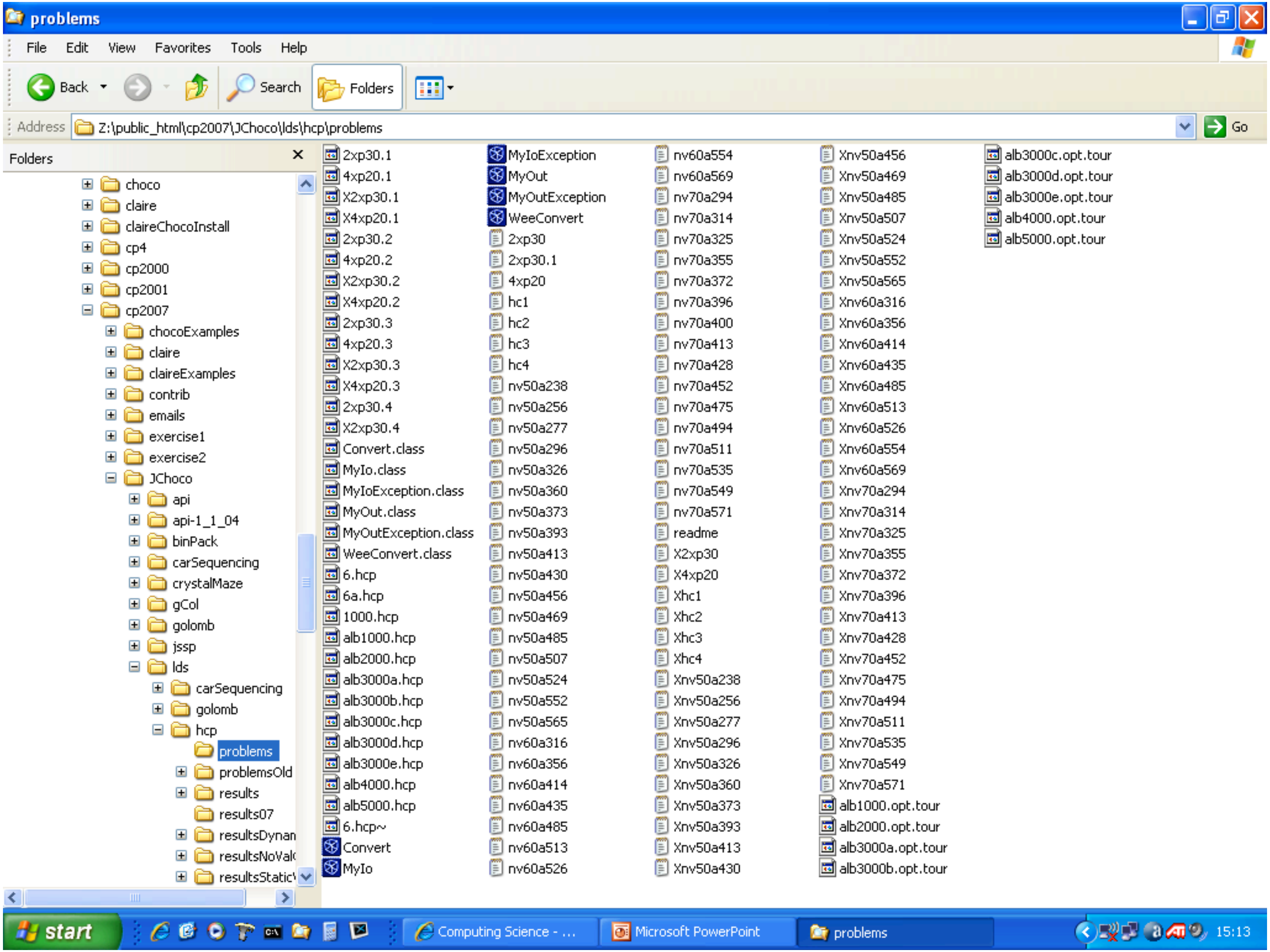
A *Hamiltonian path* or *traceable path* is a path that visits each vertex exactly once. A graph that contains a Hamiltonian path is called a *traceable graph*. A graph is *Hamilton-connected* if for every pair of vertices there is a Hamiltonian path between the two vertices.

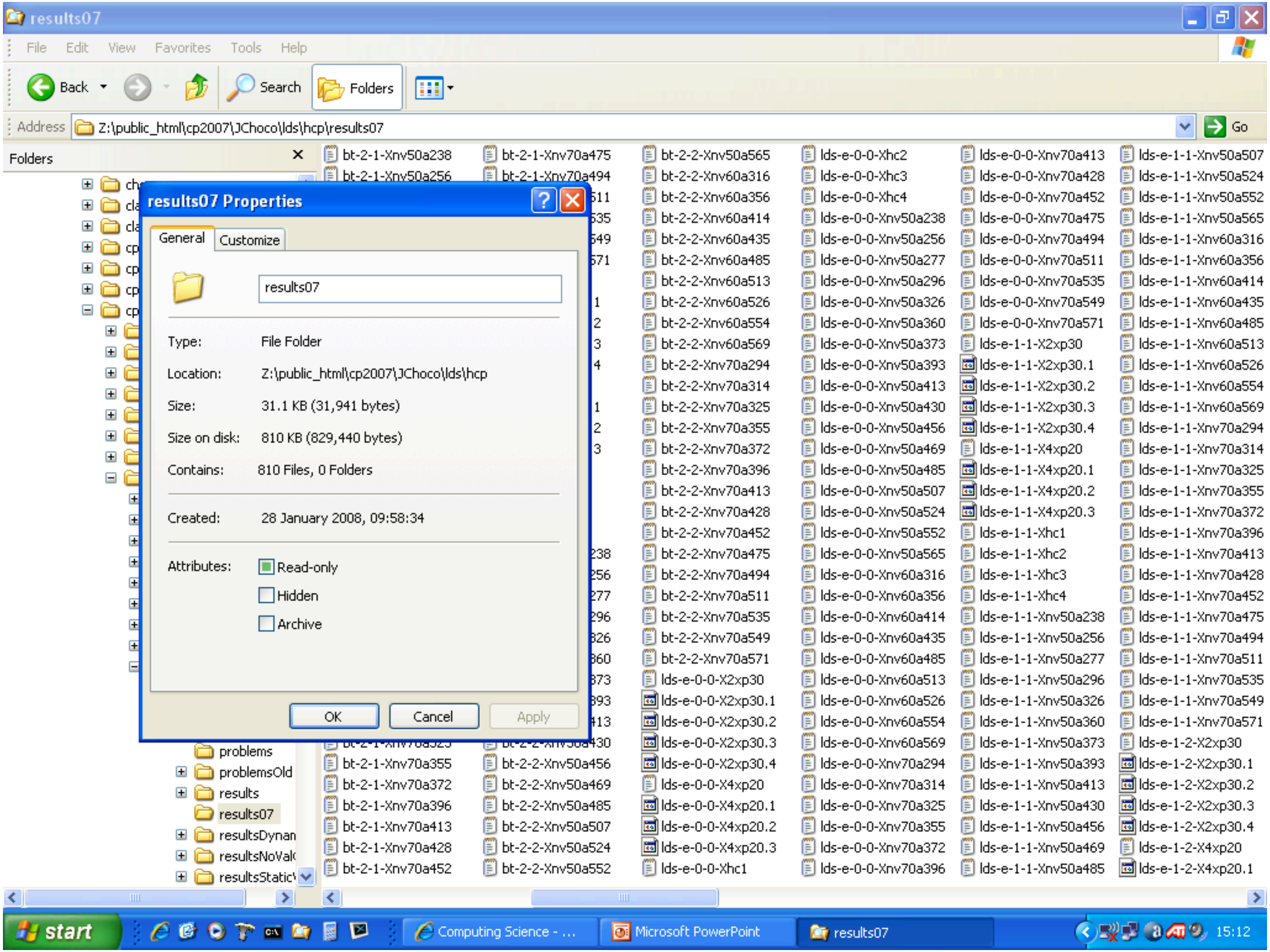
A *Hamiltonian cycle*, *Hamiltonian circuit*, *vertex tour* or *graph cycle* is a cycle that visits each vertex exactly once (except the vertex which is both the start and end, and so is visited twice). A graph that contains a Hamiltonian cycle is called a *Hamiltonian graph*.

What's involved?

ecai08 rejects




problems
File Edit View Favorites Tools Help
Back
Search
Folders
Address Z:\public_html\cp2007\JChoco\lds\hcp\problems
Go
Folders
choco
claire
claireChocoInstall
cp4
cp2000
cp2001
cp2007
chocoExamples
claire
claireExamples
contrib
emails
exercise1
exercise2
JChoco
api
api-1_1_04
binPack
carSequencing
crystalMaze
gCol
golomb
jssp
lds
carSequencing
golomb
hcp
problems
problemsOld
results
results07
resultsDynan
resultsNoVal
resultsStatic
2xp30.1
4xp20.1
X2xp30.1
X4xp20.1
2xp30.2
4xp20.2
X2xp30.2
X4xp20.2
2xp30.3
4xp20.3
X2xp30.3
X4xp20.3
2xp30.4
X2xp30.4
Convert.class
MyIo.class
MyIoException.class
MyOut.class
MyOutException.class
WeeConvert.class
6.hcp
6a.hcp
1000.hcp
alb1000.hcp
alb2000.hcp
alb3000a.hcp
alb3000b.hcp
alb3000c.hcp
alb3000d.hcp
alb3000e.hcp
alb4000.hcp
alb5000.hcp
6.hcp~
Convert
MyIo
MyIoException
MyOut
MyOutException
WeeConvert
2xp30
2xp30.1
4xp20
hc1
hc2
hc3
hc4
nv50a238
nv50a256
nv50a277
nv50a296
nv50a326
nv50a360
nv50a373
nv50a393
nv50a413
nv50a430
nv50a456
nv50a469
nv50a485
nv50a507
nv50a524
nv50a552
nv50a565
nv60a316
nv60a356
nv60a414
nv60a435
nv60a485
nv60a513
nv60a526
nv60a554
nv60a569
nv70a294
nv70a314
nv70a325
nv70a355
nv70a372
nv70a396
nv70a400
nv70a413
nv70a428
nv70a452
nv70a475
nv70a494
nv70a511
nv70a535
nv70a549
nv70a571
readme
X2xp30
X4xp20
Xhc1
Xhc2
Xhc3
Xhc4
Xnv50a238
Xnv50a256
Xnv50a277
Xnv50a296
Xnv50a326
Xnv50a360
Xnv50a373
Xnv50a393
Xnv50a413
Xnv50a430
Xnv50a456
Xnv50a469
Xnv50a485
Xnv50a507
Xnv50a524
Xnv50a552
Xnv50a565
Xnv60a316
Xnv60a356
Xnv60a414
Xnv60a435
Xnv60a485
Xnv60a513
Xnv60a526
Xnv60a554
Xnv60a569
Xnv70a294
Xnv70a314
Xnv70a325
Xnv70a355
Xnv70a372
Xnv70a396
Xnv70a413
Xnv70a428
Xnv70a452
Xnv70a475
Xnv70a494
Xnv70a511
Xnv70a535
Xnv70a549
Xnv70a571
alb1000.opt.tour
alb2000.opt.tour
alb3000a.opt.tour
alb3000b.opt.tour
alb3000c.opt.tour
alb3000d.opt.tour
alb3000e.opt.tour
alb4000.opt.tour
alb5000.opt.tour
start
Computing Science - ...
Microsoft PowerPoint
problems
15:13



results07
File Edit View Favorites Tools Help
Back
Search
Folders
Address Z:\public_html\cp2007\JChoco\lds\hcp\results07
Go
Folders
results07 Properties
General Customize
results07
Type: File Folder
Location: Z:\public_html\cp2007\JChoco\lds\hcp
Size: 31.1 KB (31,941 bytes)
Size on disk: 810 KB (829,440 bytes)
Contains: 810 Files, 0 Folders
Created: 28 January 2008, 09:58:34
Attributes: Read-only
Hidden
Archive
OK Cancel Apply
problems
problemsOld
results
results07
start
Computing Science - ...
Microsoft PowerPoint
results07
15:12

HCP: What's involved?

ecai08 rejects

- constraint model, single successor
- subtour elimination constraint
- heuristic
    - constraint model maintains information
- resultant model is then Algorithm 595 (a surprise)
- locate benchmark problems
- perform experiments
    - late v early discrepancies
    - heuristics static v dynamic

Acrobat Reader - [LDS testing the hypothesis.pdf]

ecai08 rejects

clude from this? There is no clear winner, suggesting that these static value heuristics are as reliable at the top of search as they are deep in search. Now looking at BT, heuristic H2 is as good as or better than H1 on 12 instances and worse than H1 on 2 instances. This suggests that H2 is a better heuristic than H1. However, ILDSN solves more instances that either of the backtrackers. Therefore in this class of problem it appears that ILDSN is a better option than BT.

3.2 Hamiltonian Circuit

A Hamiltonian Circuit (HC) of a graph is a circuit that visits each of the vertices once and once only. The problem was coded in JChoco using a single successor model of a directed graph. For the directed graph $G = (V, E)$ we have constrained integer variables v_0 to v_{n-1} with domains $domain(v_i)$ such that $j \in domain(v_i)$ if and only if $(i, j) \in E$. An allDifferent constraint [13] is posted across the v variables. However, this is not sufficient to guarantee that circuits are produced. Caseau and Laburthe's sub-tour elimination constraint [4] was encoded as a specialised constraint in JChoco and posted over the v variables. A second set of auxiliary variables was used to supply heuristic information. For each variable v_i we have a corresponding variable w_i and the channelling constraints $v_i = j \leftrightarrow w_j = i$, consequently the cardinality of $domain(v_i)$ is equal to the out-degree $deg^-(i)$ and the cardinality of $domain(w_i)$ is the in-degree $deg^+(i)$.

Dynamic variable and value ordering heuristics were used. The current variable selected is the variable of greatest in-degree, tie breaking on the vertex with smallest out-degree. That is, we choose the vertex that is easiest to get to (i.e. v_i such that $|domain(w_i)|$ is a maximum) tie breaking on the easiest vertex to get to but hardest to leave (i.e. tie breaking on minimum $|domain(v_i)|$. Having selected variable v_i the value $j \in domain(v_i)$ is selected such that it has the minimum value of $min(|domain(v_j)|, |domain(w_j)|)$, tie breaking on the minimum of $|domain(v_j)| + |domain(w_j)|$. That is, we choose the edge that goes to the vertex that is hardest to get to or hardest to leave, tie breaking on the vertex with smallest connectivity. This model corresponds to a constraint encoding of Martello's Algorithm 595 [11].

Experiments were performed over the HC benchmark problems of Dovier, Formisano and Pontelli [5]. Only soluble instances are presented in Table 2. Again, an entry of - corresponds to the search limit of 500000 nodes being exceeded with no HC found and figures in brackets correspond to search using the above value ordering heuristic statically and variables ordered by index. Instance 2xp30.3 is omitted as it is identical to instance 2xp30.2. Instances hc1 to hc4 have 200 vertices and 1250 directed edges. Instances rw50, rw60, and rw70 have 50, 60, and 70 vertices respectively, and instances 2xp30 and 4xp20 have 60 and 80 vertices.

When using the dynamic ordering heuristics we expect that taking early discrepancies should dominate late discrepancies, again assuming that the heuristic is less accurate at the top of search and more reliable when re-computed deep in search on a smaller problem where more inferencing has taken place. In instances 2xp30.2 and 4xp20.2 early discrepancies are indeed an advantage when using our dynamic value ordering heuristic, and late discrepancy an advantage when using our static value ordering heuristic. This is what we should expect. However, elsewhere early discrepancies are either a significant disadvantage or make no significant difference regardless of whether heuristics are dynamic or static. Finally, when using the static heuristics chronological backtracking matches or out performs both versions of ILDSN over all instances, and we should note that

problem	ildsn-e		ildsn-l		bt	
2xp30.1	-	(-)	-	(-)	-	(-)
2xp30.2	**7404**	(32205)	117907	(**15423**)	-	(60)
4xp20.2	**14932**	(12058)	69194	(**7485**)	-	(102)
hc1	200	(200)	200	(200)	200	(200)
hc2	396	(326)	400	(327)	211	(201)
hc3	563	(1065)	**387**	(**470**)	201	(211)
hc4	945	(721)	**387**	(**485**)	201	(205)
rw50a238	50	(96)	50	(74)	50	(51)
rw50a256	174	(**61**)	**126**	(71)	51	(54)
rw50a277	70	(50)	71	(50)	52	(50)
rw50a296	50	(50)	50	(50)	50	(50)
rw50a326	239	(50)	**73**	(50)	52	(50)
rw50a360	50	(50)	50	(50)	50	(50)
rw50a373	81	(50)	82	(50)	52	(50)
rw50a393	50	(85)	50	(**55**)	50	(52)
rw50a413	78	(50)	79	(50)	56	(50)
rw50a430	121	(50)	**82**	(50)	51	(50)
rw50a456	**94**	(88)	95	(89)	51	(51)
rw50a469	120	(50)	**91**	(50)	51	(50)
rw50a485	50	(89)	50	(90)	50	(51)
rw50a507	50	(122)	50	(**87**)	50	(51)
rw50a524	50	(129)	50	(**96**)	50	(52)
rw50a552	95	(50)	103	(50)	51	(50)
rw50a565	50	(50)	50	(50)	50	(50)
rw60a316	60	(60)	60	(60)	60	(60)
rw60a356	60	(60)	60	(60)	60	(60)
rw60a414	60	(60)	60	(60)	60	(60)
rw60a435	107	(60)	113	(60)	163	(60)
rw60a485	60	(60)	60	(60)	60	(60)
rw60a513	113	(60)	114	(60)	61	(60)
rw60a526	60	(60)	60	(60)	60	(60)
rw60a554	60	(60)	60	(60)	60	(60)
rw60a569	211	(105)	**118**	(110)	61	(62)
rw70a294	2456	(70)	**115**	(70)	71	(70)
rw70a314	146	(70)	**122**	(70)	74	(70)
rw70a325	70	(146)	70	(**115**)	70	(73)
rw70a355	104	(139)	105	(**97**)	75	(71)
rw70a372	70	(160)	70	(**115**)	70	(94)
rw70a396	70	(105)	70	(111)	70	(73)
rw70a413	208	(70)	**126**	(70)	71	(70)
rw70a428	70	(70)	70	(70)	70	(70)
rw70a452	70	(70)	70	(70)	70	(70)
rw70a475	104	(244)	105	(**115**)	141	(71)
rw70a494	70	(70)	70	(70)	70	(70)
rw70a511	70	(70)	70	(70)	70	(70)
rw70a535	70	(110)	70	(120)	70	(72)
rw70a549	70	(113)	70	(114)	70	(71)
rw70a571	70	(70)	70	(70)	70	(70)

Table 2. The number of nodes explored to determine if a specified directed graph has a Hamiltonian Circuit. First column is ILDSN taking discrepancies early, 2nd column ILDSN with late discrepancies, and 3d column for chronological backtracking, all using dynamic variable and value ordering heuristics. The figures in brackets are for search using a static value ordering heuristic and variables in index order. An entry of - corresponds to 500000 nodes explored and no HC found. Best results for ILDSN are given in bold. Benchmark problems are due to [5].

Still to do

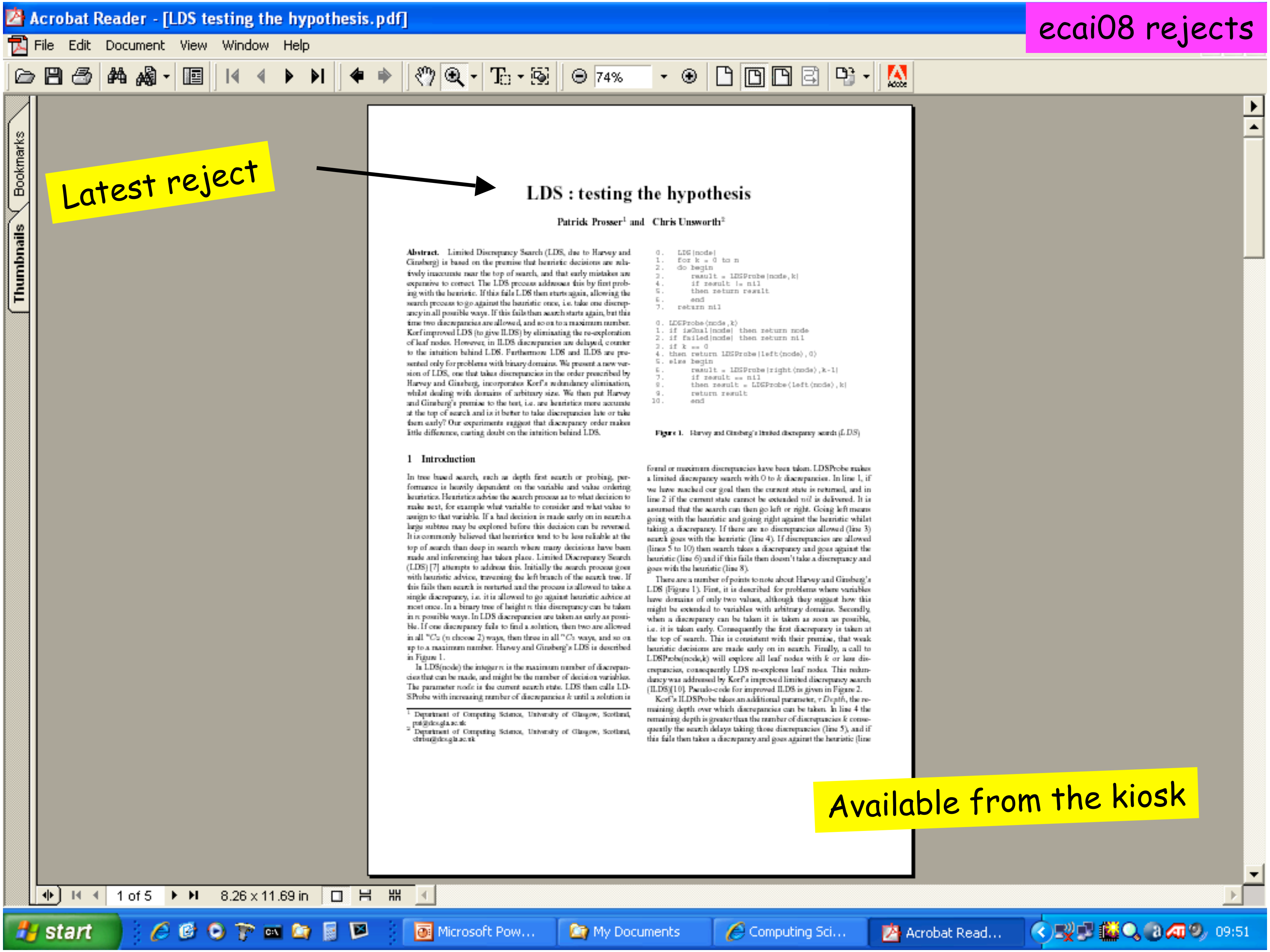
- lds - extremely early and extremely late
- jobshop scheduling using Cheng & Smith's heuristic
  - repeat H & G's experiments going late/early
- number partitioning with CKK
  - repeat Korf's experiments going late/early
- fair bit of implementation and analysis
  - 2 months work at least
  - should get 2 more tables ☺
  - might get new ****09 rejection ☹

What lessons can we learn?

- we can just follow on without question (read the most recent paper)
  - forget the basic/initial hypothesis
  - forget to really look at our results
  - take too much for granted
- it is not uninteresting to repeat someone elses's experiments.
- do not be frightened or disinterested in –ve results or different results (above)
- don't do just one set of experiments when you can do many (different domains)
- published papers may have multiple errors (beware)
- be paranoid (Am I right? Is he right? How do I know I'm right?)
- know that we are human

ecai08 rejects

Latest reject



LDS : testing the hypothesis

Patrick Prosser[1] and Chris Unsworth[2]

Abstract. Limited Discrepancy Search (LDS, due to Harvey and Ginsberg) is based on the premise that heuristic decisions are relatively inaccurate near the top of search, and that early mistakes are expensive to correct. The LDS process addresses this by first probing with the heuristic. If this fails LDS then starts again, allowing the search process to go against the heuristic once, i.e. take one discrepancy in all possible ways. If this fails then search starts again, but this time two discrepancies are allowed, and so on to a maximum number. Korf improved LDS (to give ILDS) by eliminating the re-exploration of leaf nodes. However, in ILDS discrepancies are delayed, counter to the intuition behind LDS. Furthermore LDS and ILDS are presented only for problems with binary domains. We present a new version of LDS, one that takes discrepancies in the order prescribed by Harvey and Ginsberg, incorporates Korf's redundancy elimination, whilst dealing with domains of arbitrary size. We then put Harvey and Ginsberg's premise to the test, i.e. are heuristics more accurate at the top of search and is it better to take discrepancies late or take them early? Our experiments suggest that discrepancy order makes little difference, casting doubt on the intuition behind LDS.

```
0.   LDS(node)
1.   for k = 0 to n
2.   do begin
3.      result = LDSProbe(node,k)
4.      if result != nil
5.      then return result
6.      end
7.   return nil

0. LDSProbe(node,k)
1. if isGoal(node) then return node
2. if failed(node) then return nil
3. if k == 0
4. then return LDSProbe(left(node),0)
5. else begin
6.      result = LDSProbe(right(node),k-1)
7.      if result == nil
8.      then result = LDSProbe(left(node),k)
9.      return result
10.     end
```

Figure 1. Harvey and Ginsberg's limited discrepancy search (*LDS*)

1 Introduction

In tree based search, such as depth first search or probing, performance is heavily dependent on the variable and value ordering heuristics. Heuristics advise the search process as to what decision to make next, for example what variable to consider and what value to assign to that variable. If a bad decision is made early on in search a large subtree may be explored before this decision can be reversed. It is commonly believed that heuristics tend to be less reliable at the top of search than deep in search where many decisions have been made and inferencing has taken place. Limited Discrepancy Search (LDS) [7] attempts to address this. Initially the search process goes with heuristic advice, traversing the left branch of the search tree. If this fails then search is restarted and the process is allowed to take a single discrepancy, i.e. it is allowed to go against heuristic advice at most once. In a binary tree of height n this discrepancy can be taken in n possible ways. In LDS discrepancies are taken as early as possible. If one discrepancy fails to find a solution, then two are allowed in all ${}^nC_2$ (n choose 2) ways, then three in all ${}^nC_3$ ways, and so on up to a maximum number. Harvey and Ginsberg's LDS is described in Figure 1.

In LDS(node) the integer n is the maximum number of discrepancies that can be made, and might be the number of decision variables. The parameter *node* is the current search state. LDS then calls LDSProbe with increasing number of discrepancies k until a solution is found or maximum discrepancies have been taken. LDSProbe makes a limited discrepancy search with 0 to k discrepancies. In line 1, if we have reached our goal then the current state is returned, and in line 2 if the current state cannot be extended *nil* is delivered. It is assumed that the search can then go left or right. Going left means going with the heuristic and going right against the heuristic whilst taking a discrepancy. If there are no discrepancies allowed (line 3) search goes with the heuristic (line 4). If discrepancies are allowed (lines 5 to 10) then search takes a discrepancy and goes against the heuristic (line 6) and if this fails then doesn't take a discrepancy and goes with the heuristic (line 8).

There are a number of points to note about Harvey and Ginsberg's LDS (Figure 1). First, it is described for problems where variables have domains of only two values, although they suggest how this might be extended to variables with arbitrary domains. Secondly, when a discrepancy can be taken it is taken as soon as possible, i.e. it is taken early. Consequently the first discrepancy is taken at the top of search. This is consistent with their premise, that weak heuristic decisions are made early on in search. Finally, a call to LDSProbe(node,k) will explore all leaf nodes with k or less discrepancies, consequently LDS re-explores leaf nodes. This redundancy was addressed by Korf's improved limited discrepancy search (ILDS)[10]. Pseudo-code for improved ILDS is given in Figure 2.

Korf's ILDSProbe takes an additional parameter, *rDepth*, the remaining depth over which discrepancies can be taken. In line 4 the remaining depth is greater than the number of discrepancies k consequently the search delays taking those discrepancies (line 5), and if this fails then takes a discrepancy and goes against the heuristic (line

[1] Department of Computing Science, University of Glasgow, Scotland, pat@dcs.gla.ac.uk
[2] Department of Computing Science, University of Glasgow, Scotland, chrisu@dcs.gla.ac.uk

Available from the kiosk

Questions?

Start without me