The University of Auckland

Open Source In Optimisation

Dr. Stuart Mitchell
Department of Engineering Science
University of Auckland
New Zealand
s.mitchell@auckland.ac.nz

Contents of presentation

What is Open Source

The open source future for OR

PuLP

Genrating a Yield Frontier with PuLP

Traveling tournement problem with PuLP

Further PuLP examples

What is open source

- A term used to describe software distributed in source under licenses guaranteeing anybody rights to freely use, modify, and redistribute the code.
- Examples of Open Source software include:
  - The Linux Kernel
  - The Apache Web Server
  - The R Project for Statistical Computing
  - Open Cmiss

Statistical software nets technology award

- Otago Daily Times 11 Nov 2008

Ross Ihaka has won New Zealand's top award for technology achievements, the 2008 Pickering Medal.
Dr Ihaka, of Auckland University, gained his honour for a package of computer programmes.
The medal was one of series awarded by the national science academy, the Royal Society, before 380 of the nation's senior scientists at Te Papa in Wellington,
Dr Ihaka's software for statisticians, called R, can be downloaded free and has had huge uptake by universities, industry and government.
Readily customised for different applications, it has proved invaluable for major "data crunching" tasks such as processing genomic information.

Statistical software nets technology award

- The package and the paper introducing it have been cited over 1700 times in research papers: the highest 'hit-rate' for publications in the mathematical sciences over the past 10 years, worldwide, and it is now disseminated from over 75 internet sites in 30 countries.

The open source future for OR

- Presently the Computational OR tools used, taught within this department, are closed source.
  - Excel /Storm
  - AMPL, GAMS
  - CPLEX, EXPRESS, ZIP
- Students can not afford commercial licences of this software
- Students cannot see how this software works.

The open source future for OR

- These tools follow the 'Cathedral' style of development.
  - ... software .. built like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta to be released before its time

    Eric S. Raymond
- They are also typically expensive for non-academic use

The open source future for OR

- Open source tools, for instance the Linux Kernel, can follow a 'Bazaar' development model
  - a great babbling bazaar of differing agendas and approaches out of which a coherent and stable system could seemingly emerge only by a succession of miracles

    Eric S Raymond

The open source future for OR

- What are the attributes of the Bazaar?
  - Publicly available code that is released and updated often
  - The ability for anyone to make improvements
  - A community that pushes for constant improvement and code quality

The open source future for OR

- What are the attributes of the Bazaar?
  - Publication:
    - Publicly available code that is released and updated often
  - Collaboration
    - The ability for anyone to make improvements
  - Peer Review:
    - A community that pushes for constant improvement and code quality

The open source future for OR

- Open source therefore naturally complements the research and publication process.
- From the Coin-OR website

  Why for OR? Consider the following scenario. You read about an optimization algorithm in the literature and you get an idea on how to improve it. Today, testing your new idea typically requires re-implementing (and re-debugging and re-testing) the original algorithm. Often, clever implementation details aren't published. It can be difficult to replicate reported performance. Now imagine the scenario if the original algorithm was publicly available in a community repository. Weeks of re-implementing would no longer be required. You would simply check out a copy of it for yourself and modify it.

  Imagine the productivity gains from software reuse!

The open source future for OR

- Outcomes for students
  - Ability to access free (no cost) software to implement their own solutions once they graduate
  - Ability to access free (open) source code to see how the algorithms are implemented.
    - Imagine the difference to 391??
  - The ability to improve the software they use.

PuLP

- PuLP is a python module that allows the easy expression of Mathematical Programs
- PuLP is built to interface with separate solvers
- PuLP is similar in style to:
  - AMPL
  - GAMS
  - OPL
  - LINGO
  - FLOPC++ etc.

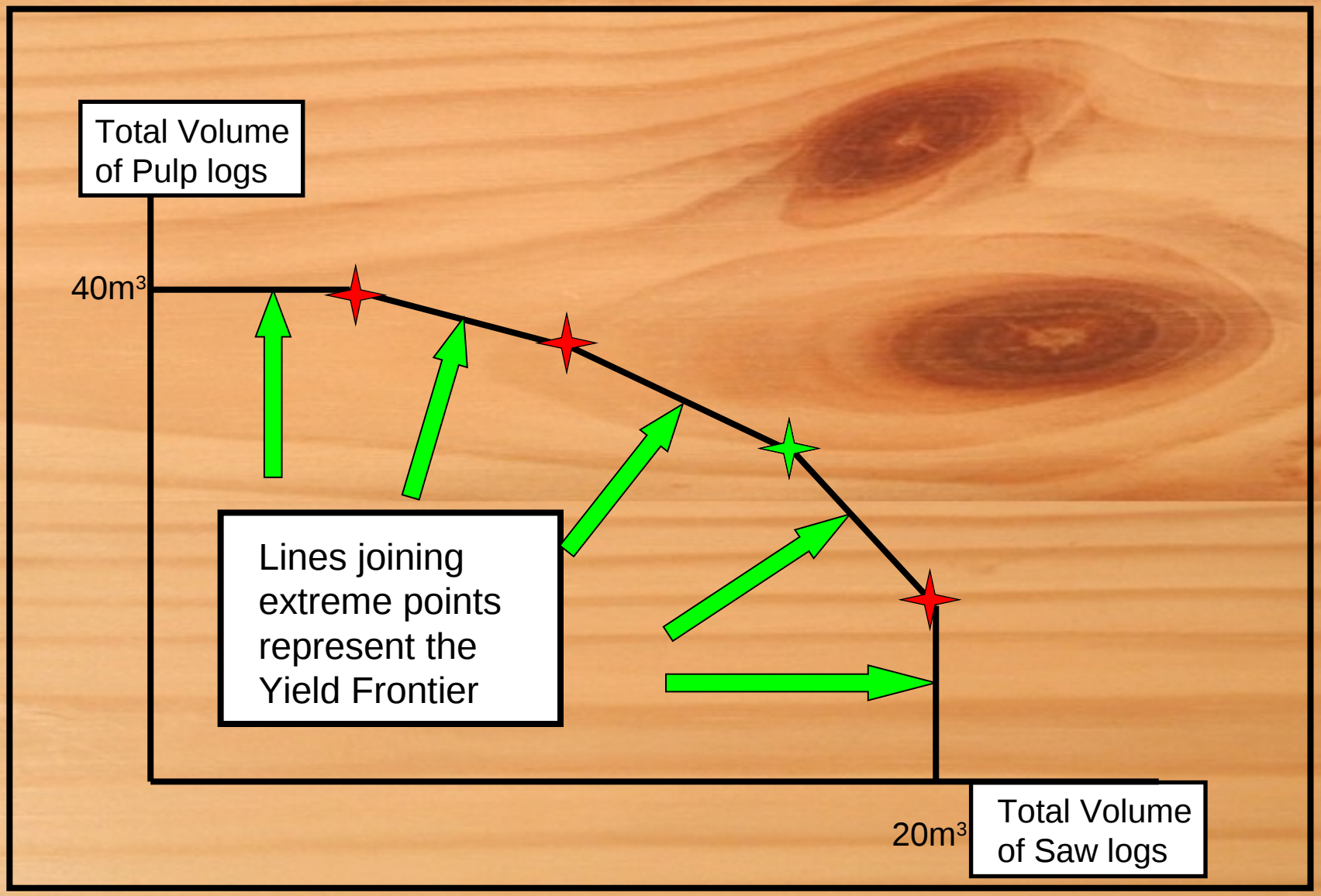
PuLP

- Why Python?
  - Core Python syntax leads to the concise statement of MP's
  - Python is a scripting language so no compilation is needed and the code is platform independent
  - Python interfaces easily with external solvers that do the heavy lifting
  - Python comes with 'batteries included'
    - The Python standard library is huge

PuLP

- Written initially by J. S. Roy
- Now maintained by S. A. Mitchell
- It is available at http://pulp-or.google-code.com
- Now available for Windows and Linux

Generating a Yield Frontier



Generating a Yield Frontier

- Using pulp we formulate the bucking problem (with a single objective) as a set packing problem by log section.

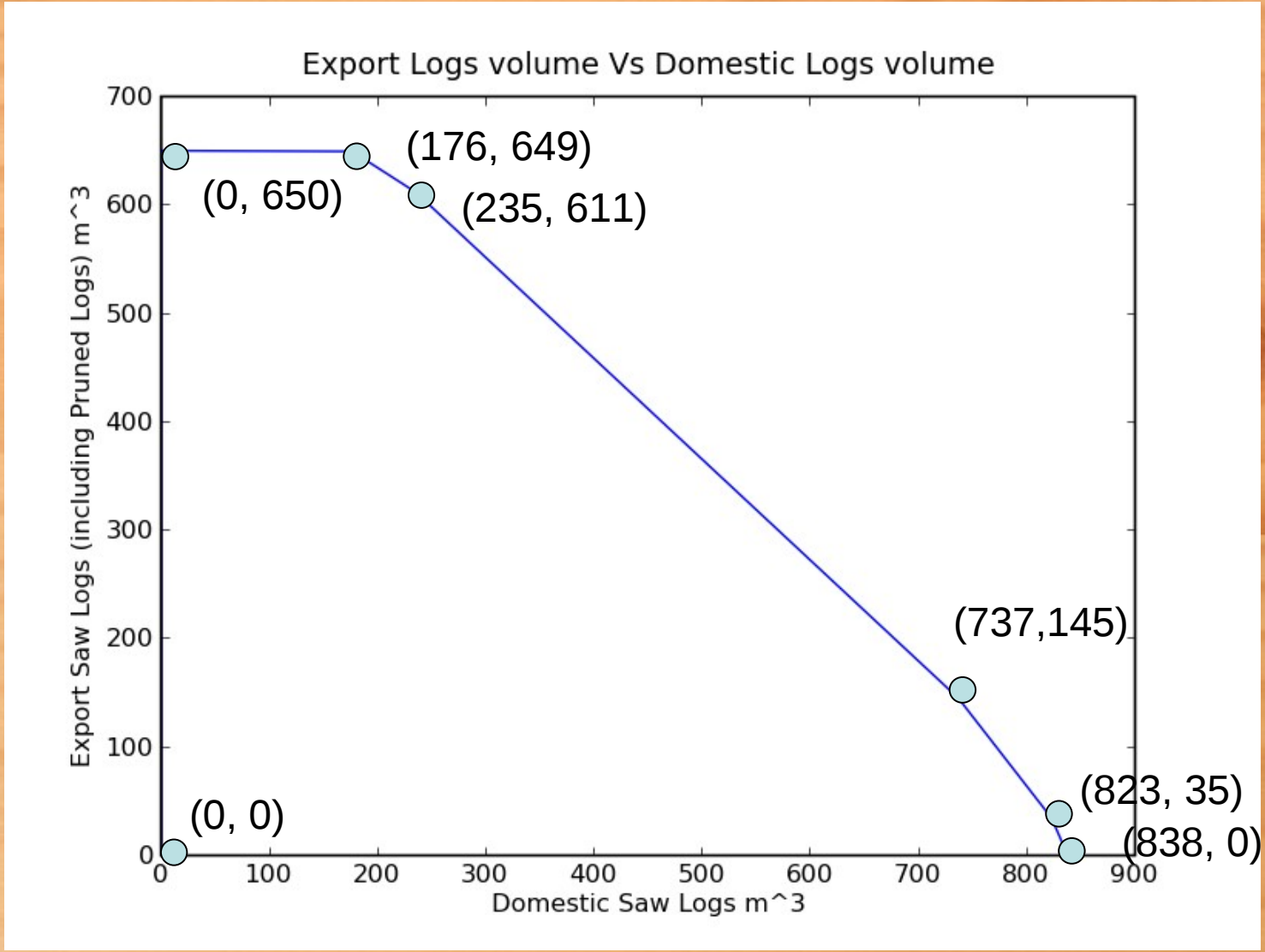
```
lp = LpProblem("Bucking Model", LpMaximize)
#set up the logvolume variables
logvol=LpVariable.dicts("LogVolume(\"%s\")",logtypes,0)
#objective
lp+=lpSum([l.price * logvol[l] for l in logtypes]), "Revenue“
#setup the arc variables
x=LpVariable.dict("x(%s)",f_logs,0,1,LpInteger)
#set up a section set partitioning problem
count = 0
for s in stems:
        slogs = fs_logs[s]
        for i,sec in enumerate(s.sections):
            lp +=( lpSum((x[log] for log in slogs
                        if log.startl <= sec.start
                        if log.endl > sec.start)) <= 1
                        , "Stem_Section(\"%s\",%i)" % (str(s),i))
            count += 1
#add the constraints that link the log volumes
for lt in logtypes:
        lp +=( lpSum((log.volume*x[log]
                    for log in fl_logs[lt])) - logvol[lt] == 0
                    , "Logtype_volume(\"%s\")" % str(lt))
```

Generating a Yield Frontier Using Pulp

- We then iteratively solve the problem to find all extreme supported solutions on the Yield Frontier
- Equivalent to projecting the problem into the log volume space
- I added a module to PuLP that implements projection using Iterative Hull Methods (Lassez, Lassez 1992)

```
>>> pprob, ppoints = polytope.project(lp, totalvars)
```

Find Yield Frontier for the Dataset



Find Yield Frontier for a Single Stem

```
\* Total projected *\
Minimize
OBJ: __dummy
Subject To
_C1: DomSaw + 1.11154598826 ex <= 2669.27592955
_C2: DomSaw + 1.34653465347 ex <= 3118.57425743
_C3: 1.00863930886 DomSaw + ex <= 2522.60691145
Bounds
__dummy = 0
End
```

Travelling tournament problem with PuLP

- This problem models the allocation of teams to Home and Away games in a tournament
- A full problem description and datasets are found at Michael Trick's page
- http://mat.tepper.cmu.edu/TOURN/

Travelling tournament problem with PuLP

- At IFORS 2008 M. Trick presented an approach to finding lower bounds to this problem using combinatorial benders cuts
- That evening I implemented his algorithm using PuLP
- Along the way I also added Powerset, Combination and Permutation operators to PuLP

```
lp = LpProblem("Travelling tournement Master", LpMinimize)
#create variables
triplist = [Trip(t1,p) for t1 in teams
            for p in
            allpermutations([t for t in
            teams if t !=t1] ,k)
            if p[0] <= p[-1]]
tripvars = LpVariable.dict("mastervar ",triplist,0,1,LpInteger)
#objective
lp += lpSum([t.cost()*tripvars[t]
                for t in triplist])
#construct constraints to ensure that all teams visit each other
    for t1 in teams:
        for t2 in teams:
            if t1 != t2:
                lp += lpSum([tripvars[t] for t in triplist
                                if t.team == t1
                                if t2 in a.awayteams]) == 1, \
                               "Team_%s_Visits_%s"%(t1,t2)
```

Further examples

- The 392 course has been converted from AMPL to PuLP
  http://130.216.209.237/engsci392/pulp/OptimisationWithPuLP
- There you can see a number of different ways to construct problems
- Note that new language features can be added very easily only needing approval from the BDFL