Quantum Annealing Applied to Optimization Problems in Radiation Medicine

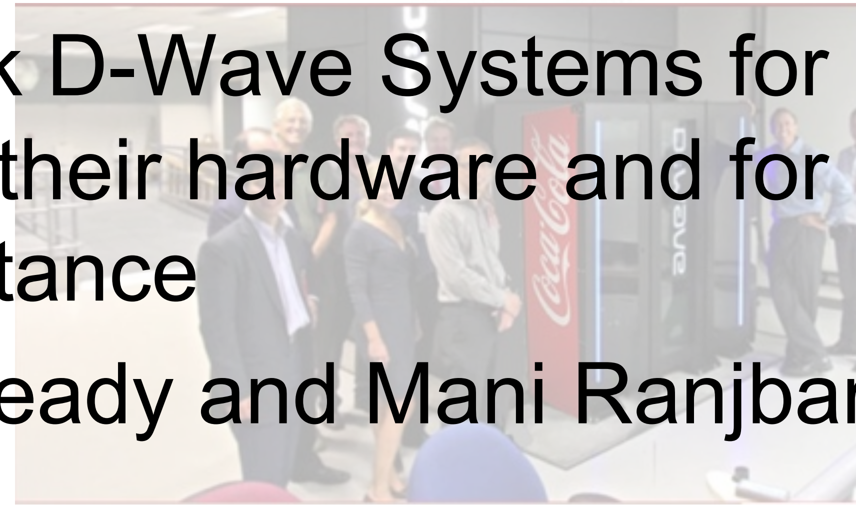
ROSWELL PARK
CANCER INSTITUTE

Daryl P. Nazareth, PhD, DABR
Medical Physicist, Assistant Professor
Department of Radiation Medicine

ROSWELL PARK CANCER INSTITUTE

Acknowledgments

- We would like to thank D-Wave Systems for providing access to their hardware and for computational assistance
- In particular, Bill Macready and Mani Ranjbar
- Jason Spaans
- Tyler Paplham



Introduction

- Cancer is one of the leading causes of morbidity and mortality worldwide
- In 2012 there were 14 million new cases of cancer
- There were 8.2 million cancer-related deaths
- Number of new cases expected to rise by 70% over next 2 decades (WHO)

Introduction

Cancer in the US:

- In 2016 there will be an estimated 1.69 million new cases of cancer
- There will be 596,000 cancer deaths
- 39.6% of people will be diagnosed with cancer in their lifetimes
- Most common types: breast, lung, prostate, colon, bladder

Introduction

Cancer is treated using three methods:

Surgery

Introduction

Cancer is treated using three methods:

Chemotherapy

Introduction

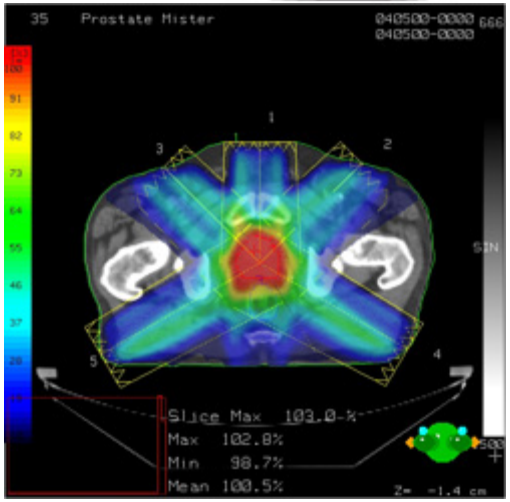
Cancer is treated using three methods:

Radiation Therapy

This is the subject of our work

Dose Calculation

Radiation dose distribution

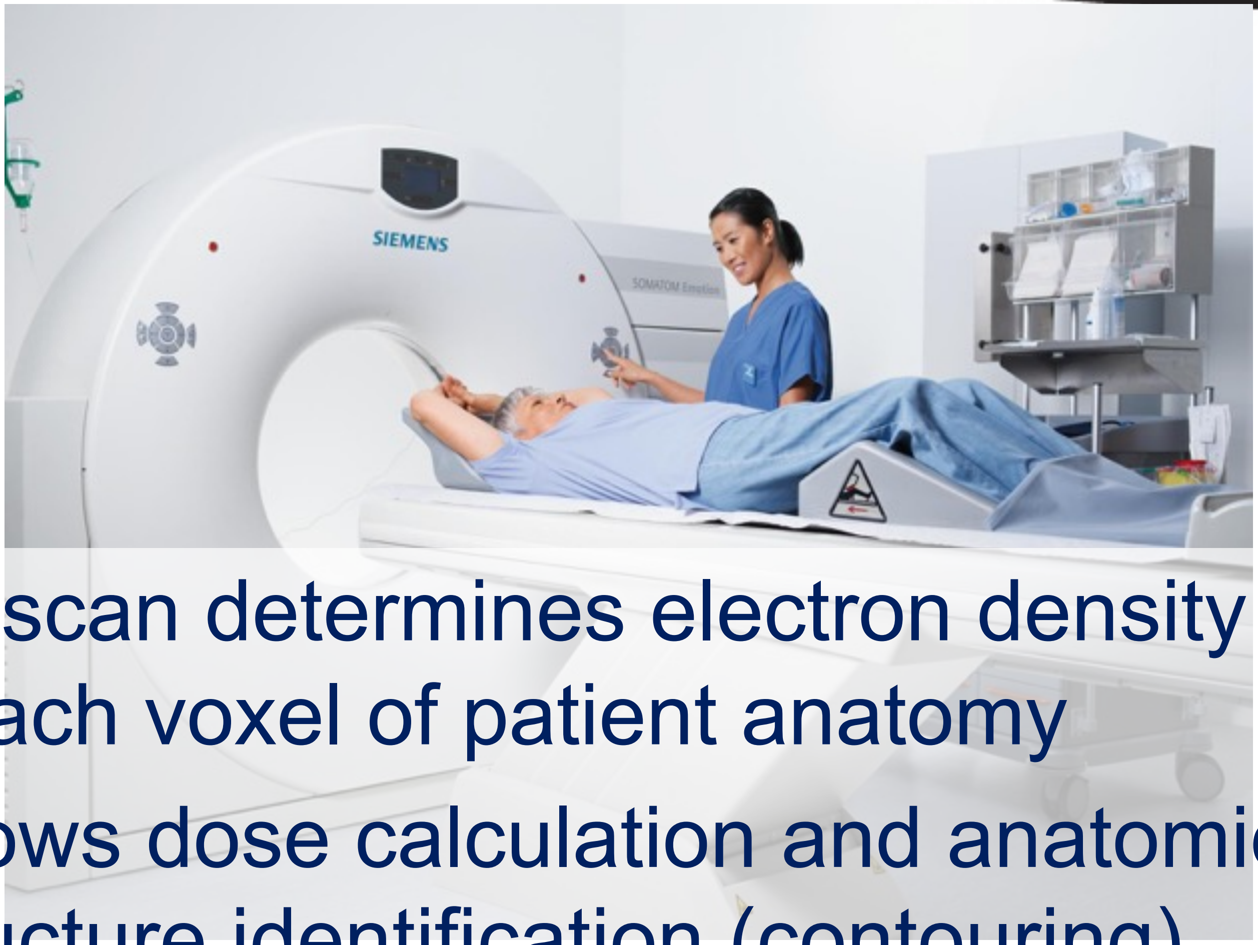


Absorbed dose measured in Gy (J/kg)

Calculated from well-known physics principles

Clinical calculations use FDA-approved software

CT Simulation



- CT scan determines electron density of each voxel of patient anatomy
- Allows dose calculation and anatomic structure identification (contouring)

Medical Linear Accelerator

[Linear Accelerator Part 1]

Radiation Production

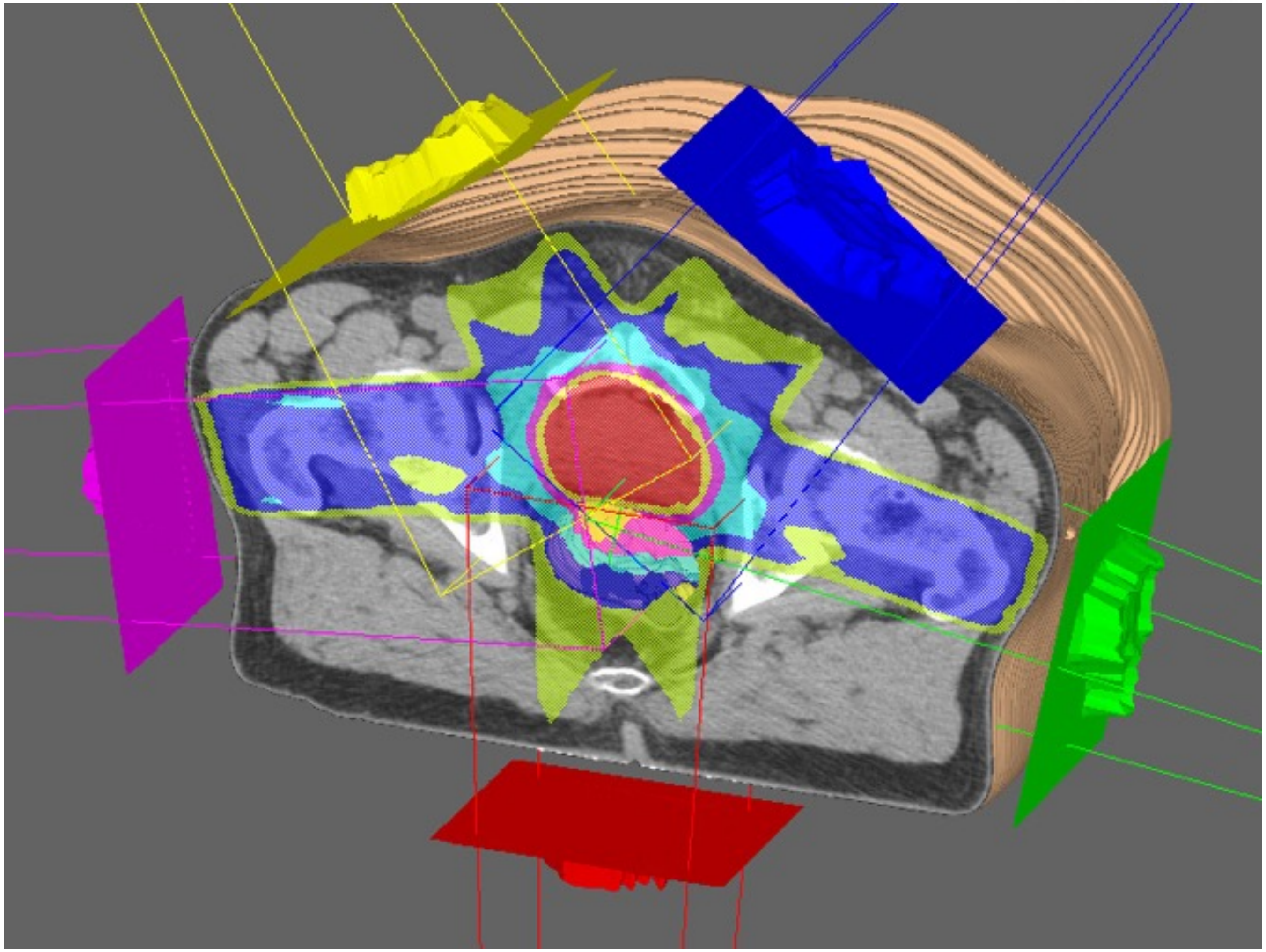
[Linear Accelerator Part 2]

Energy ~ 6 MeV

Beam Shaping

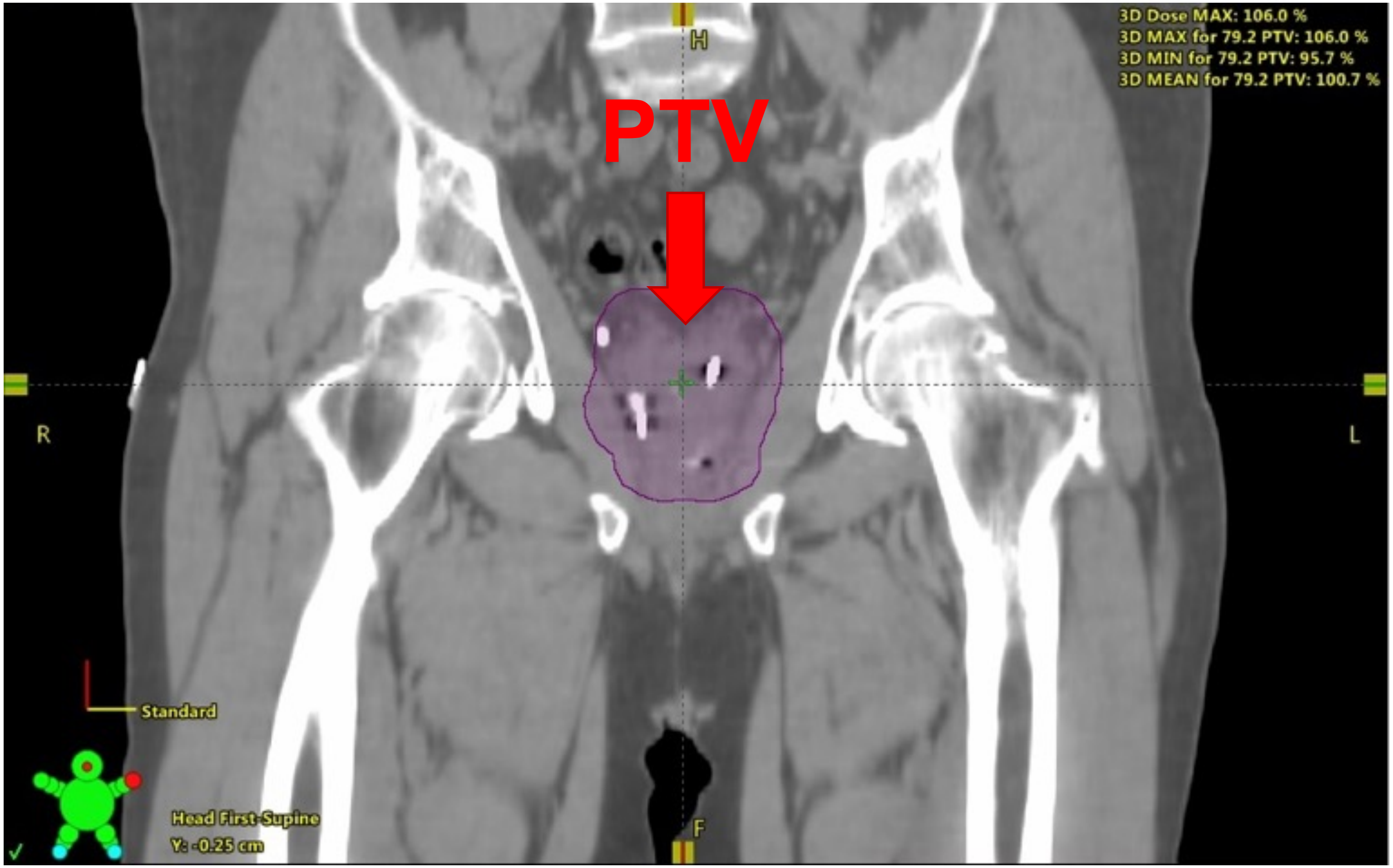
[Linear Accelerator Part 3]

IMRT



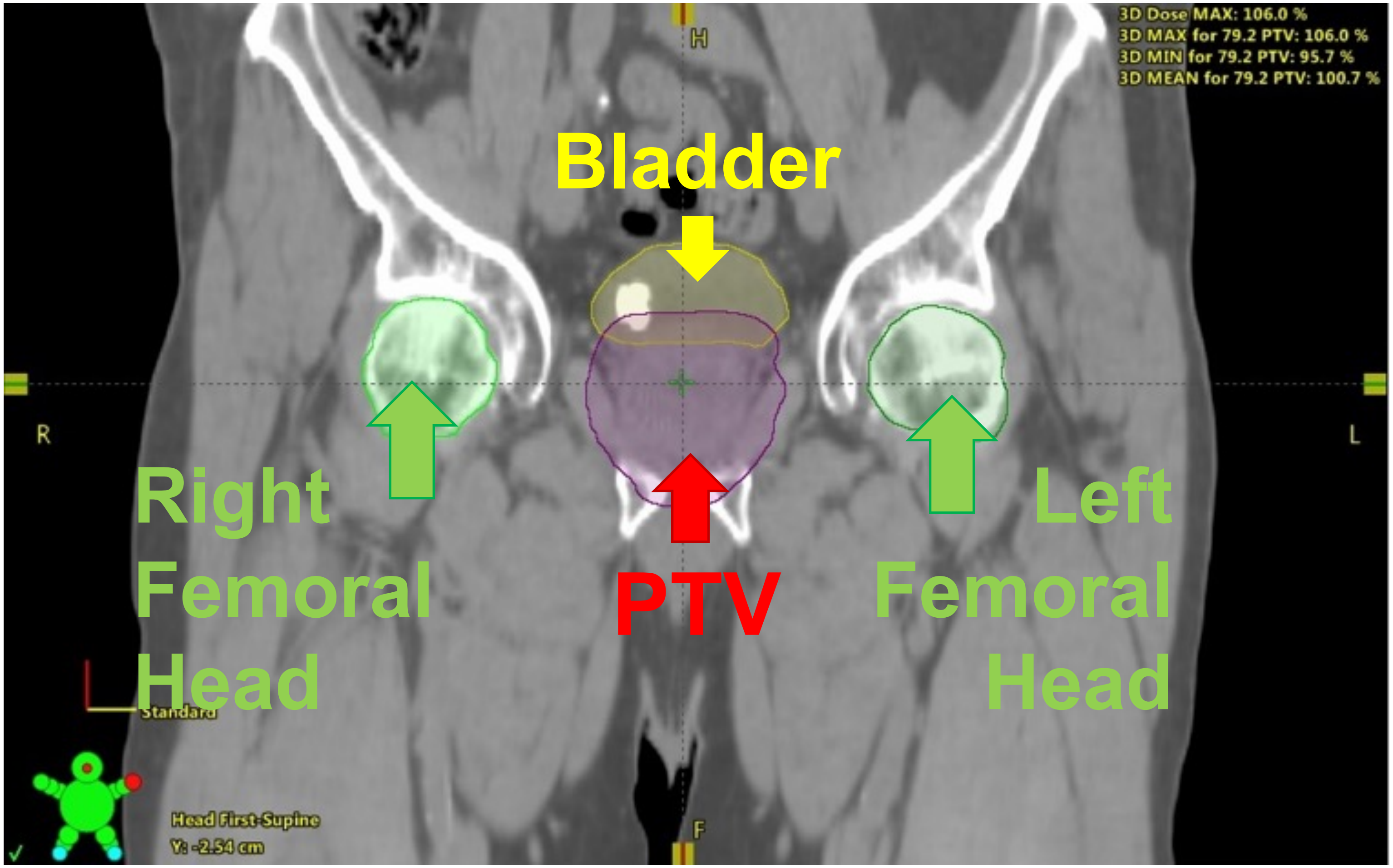
IMRT Treatment

The PTV



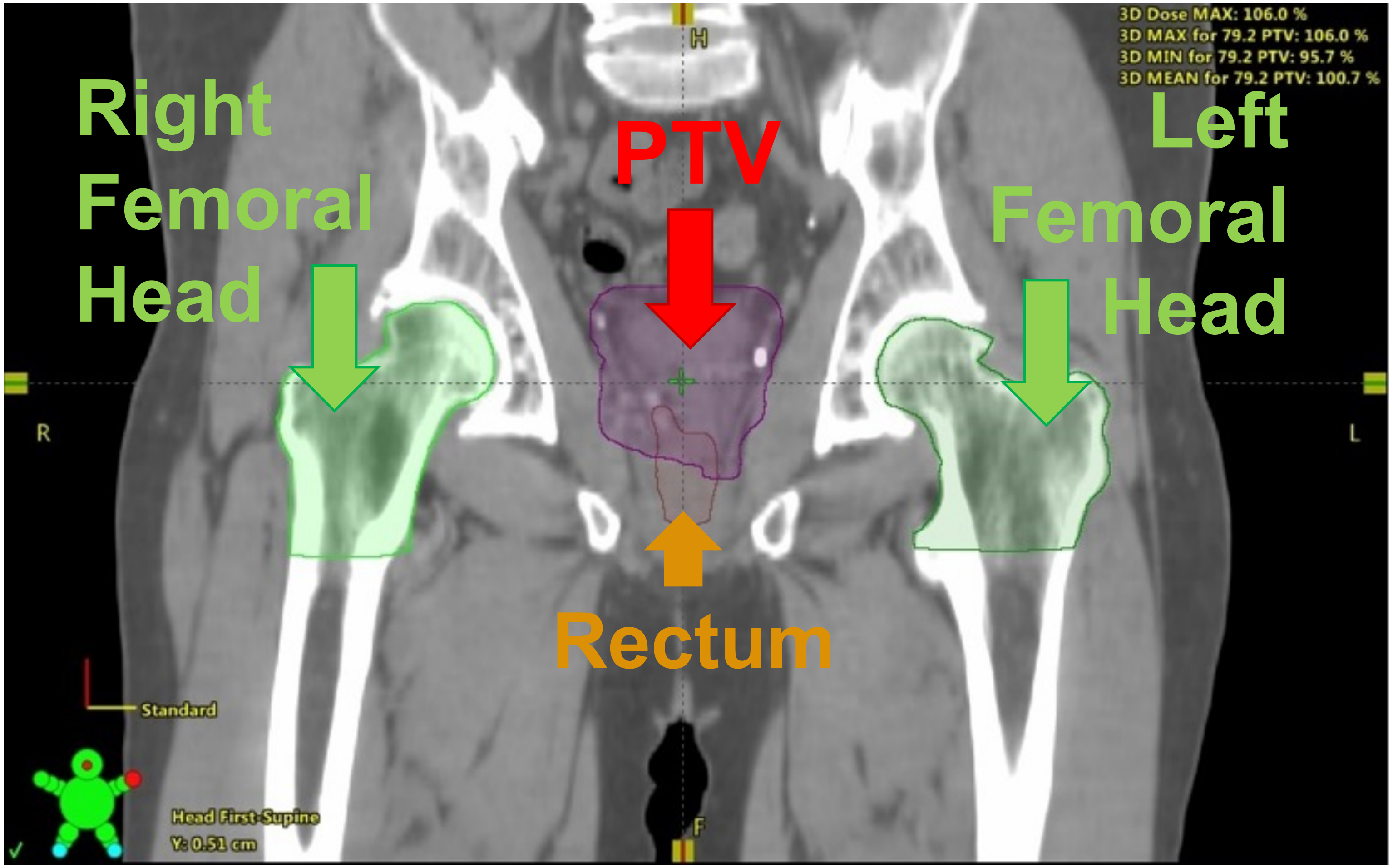
3D Dose MAX: 106.0 %
3D MAX for 79.2 PTV: 106.0 %
3D MIN for 79.2 PTV: 95.7 %
3D MEAN for 79.2 PTV: 100.7 %
PTV
H
R
L
Standard
Head First-Supine
Y: -0.25 cm
F

OARs Slice 1

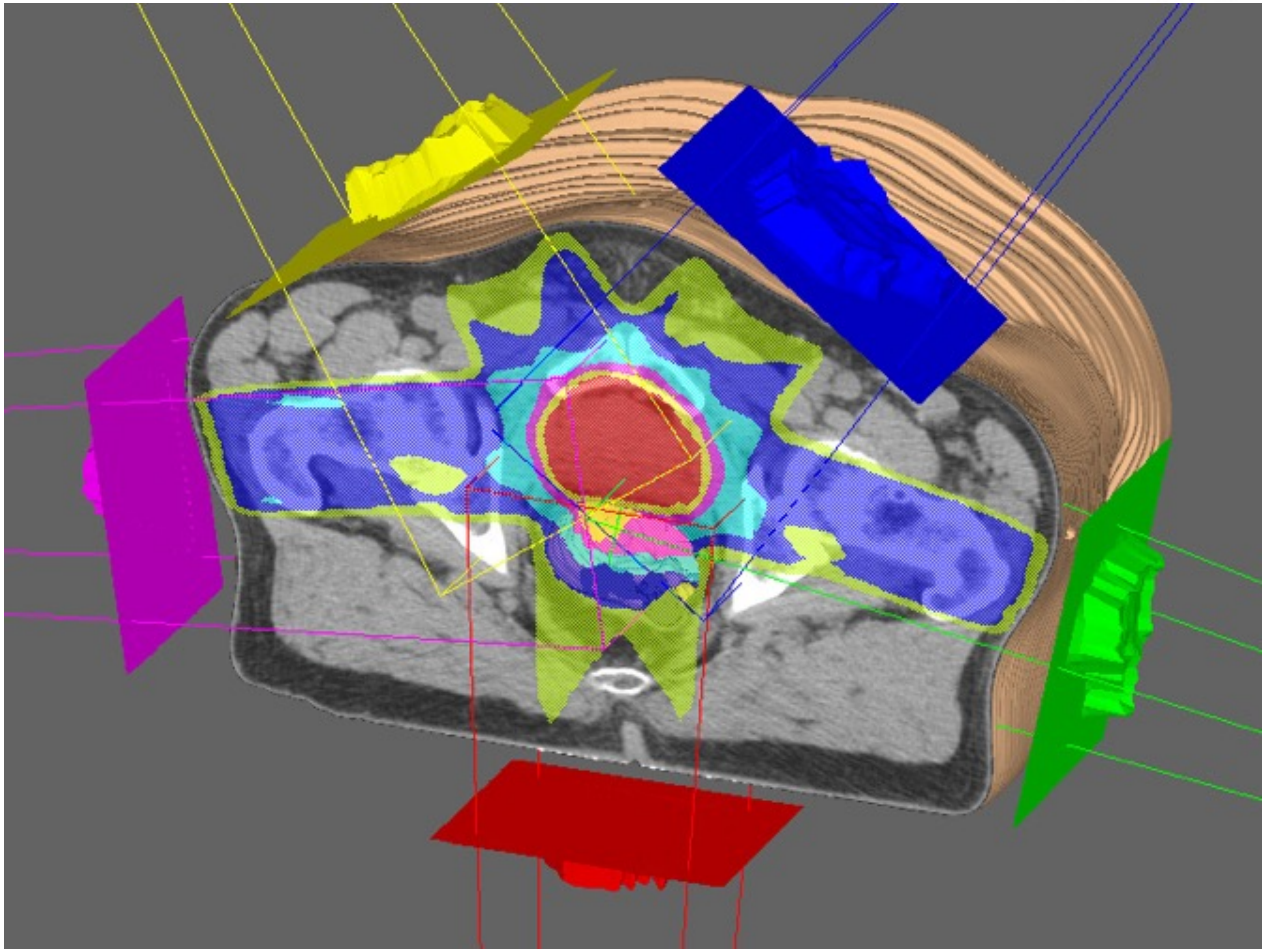


3D Dose MAX: 106.0 %
3D MAX for 79.2 PTV: 106.0 %
3D MIN for 79.2 PTV: 95.7 %
3D MEAN for 79.2 PTV: 100.7 %
H
Bladder
R
L
Right Femoral Head
PTV
Left Femoral Head
Standard
Head First-Supine
Y: -2.54 cm
F

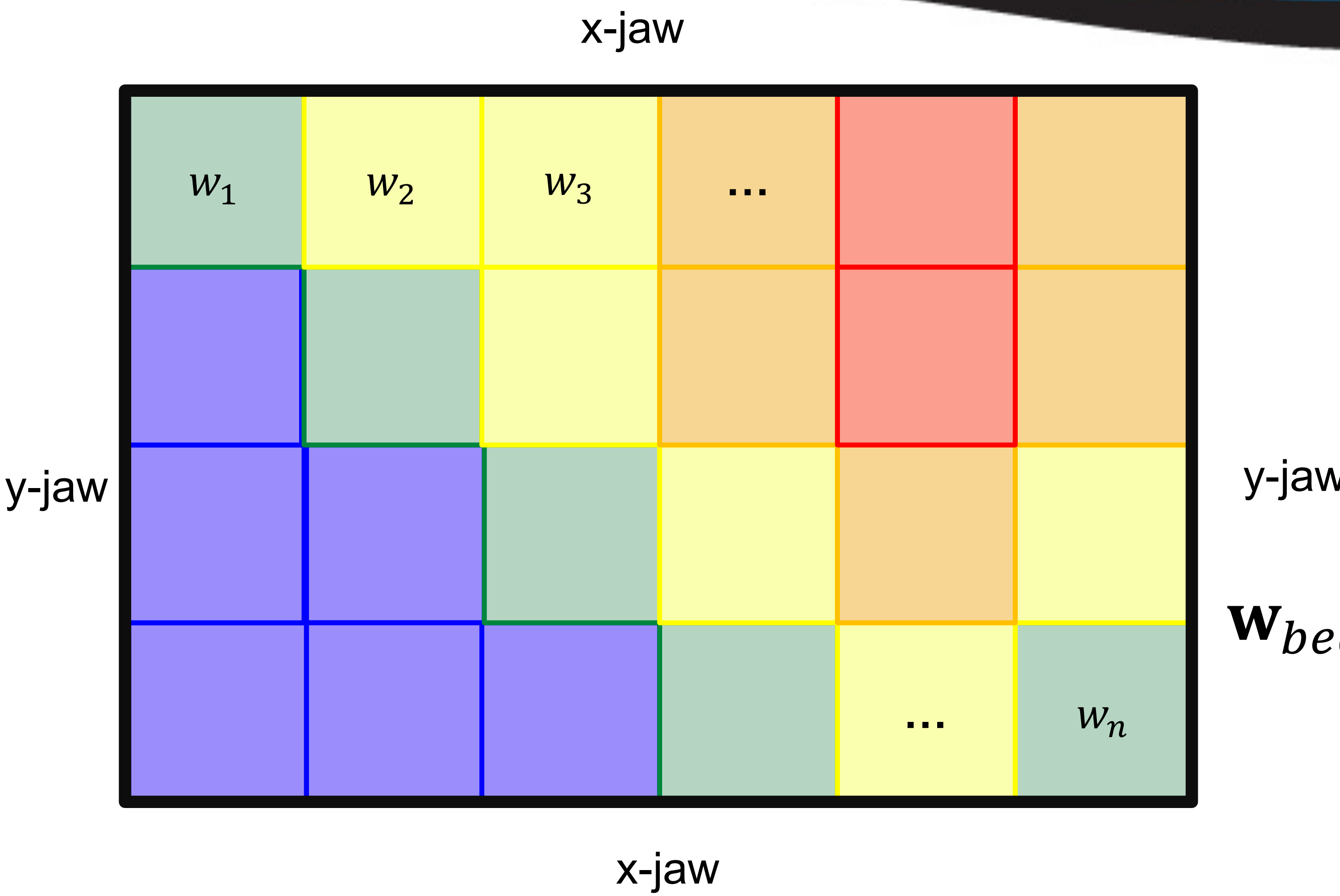
OARs Slice 2


3D Dose MAX: 106.0 %
3D MAX for 79.2 PTV: 106.0 %
3D MIN for 79.2 PTV: 95.7 %
3D MEAN for 79.2 PTV: 100.7 %
Right Femoral Head
PTV
Left Femoral Head
H
R
L
Rectum
Standard
Head First-Supine
Y: 0.51 cm
F

IMRT

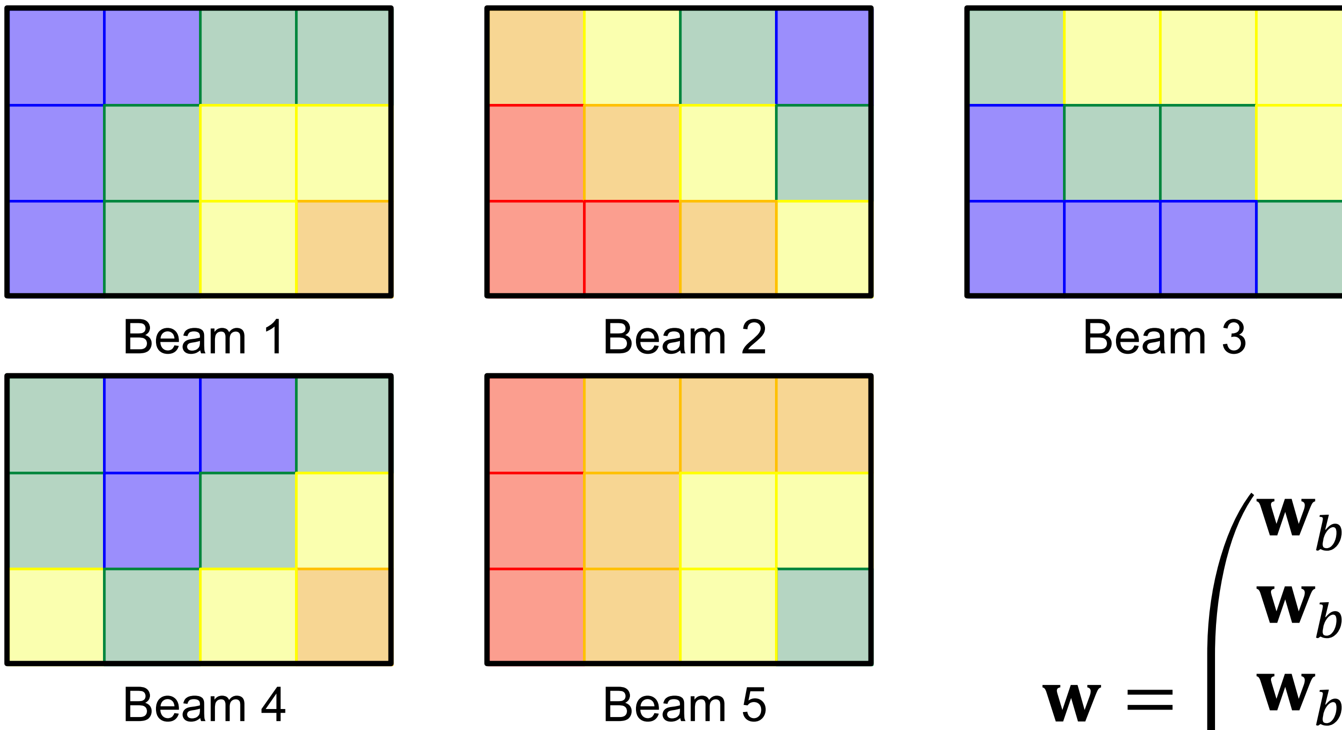


Beamlets



$$\mathbf{w}_{beam} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{pmatrix}$$

Beams

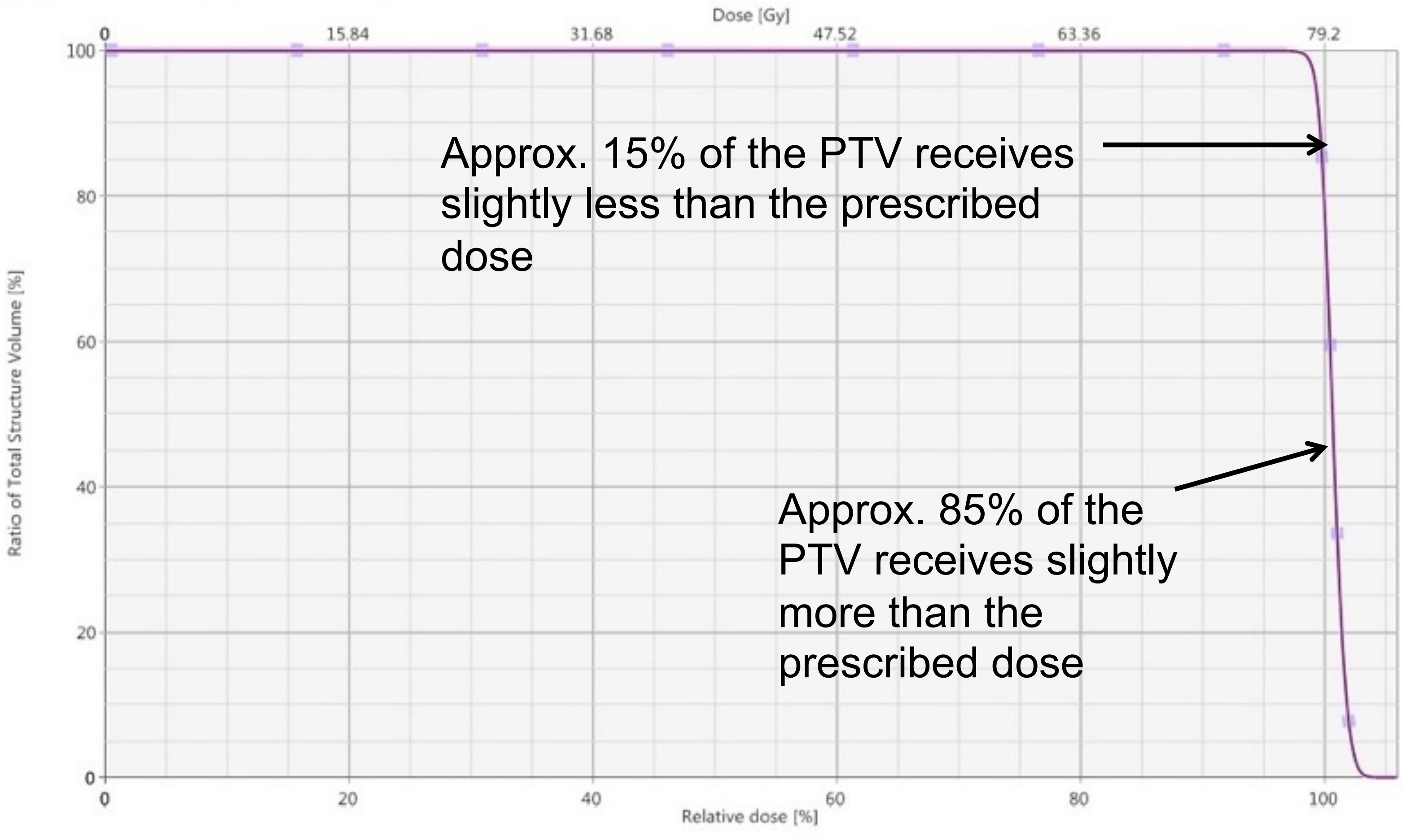


$$\mathbf{w} = \begin{pmatrix} \mathbf{w}_{beam\ 1} \\ \mathbf{w}_{beam\ 2} \\ \mathbf{w}_{beam\ 3} \\ \mathbf{w}_{beam\ 4} \\ \mathbf{w}_{beam\ 5} \end{pmatrix}$$

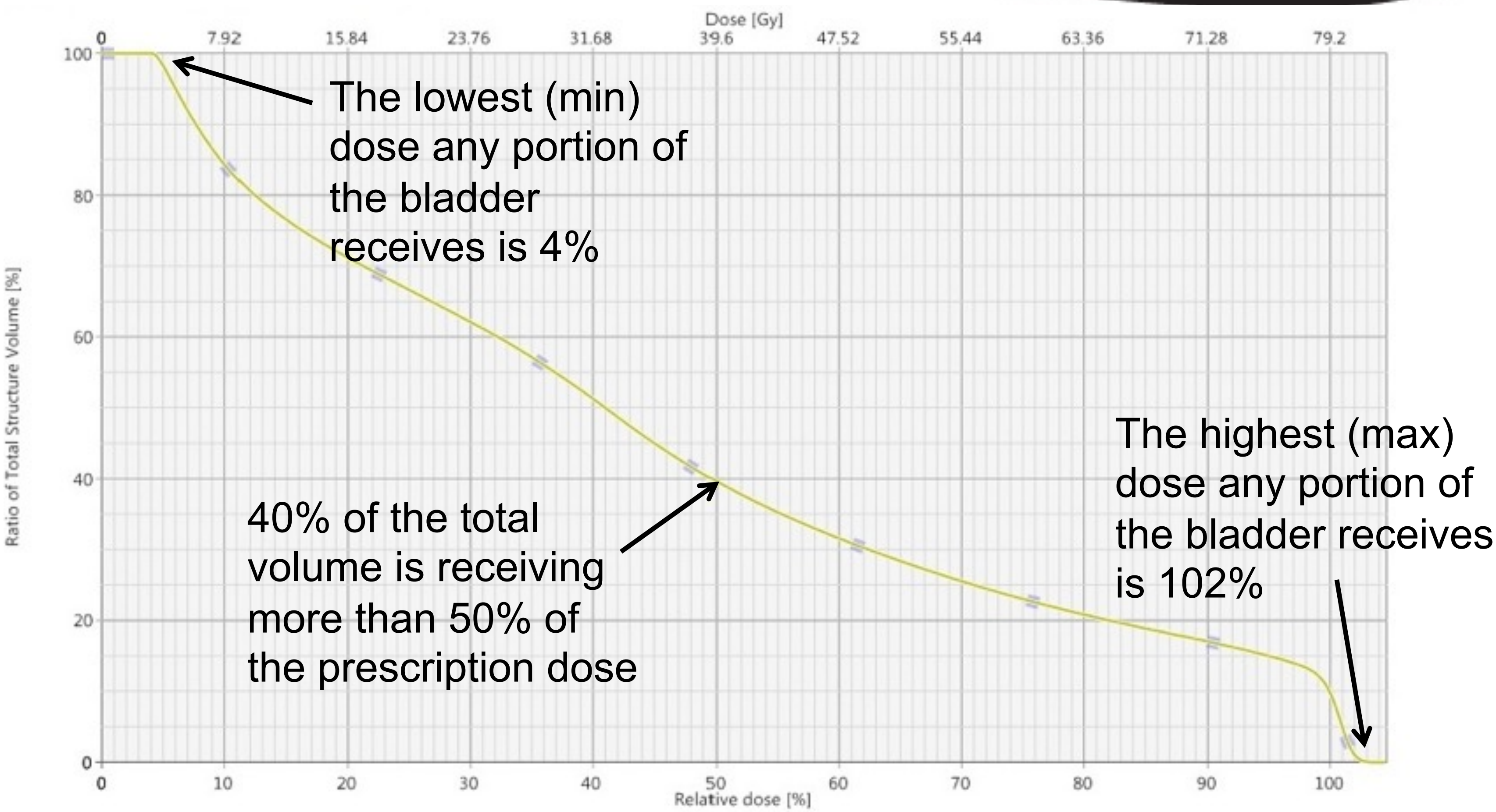
The DVH

- A Dose-Volume Histogram (DVH) is a graphical representation of the percentage of dose received by a portion of the volume
- For a given treatment plan, the PTV and each organ has an associated DVH
- Is critical to defining the objective function

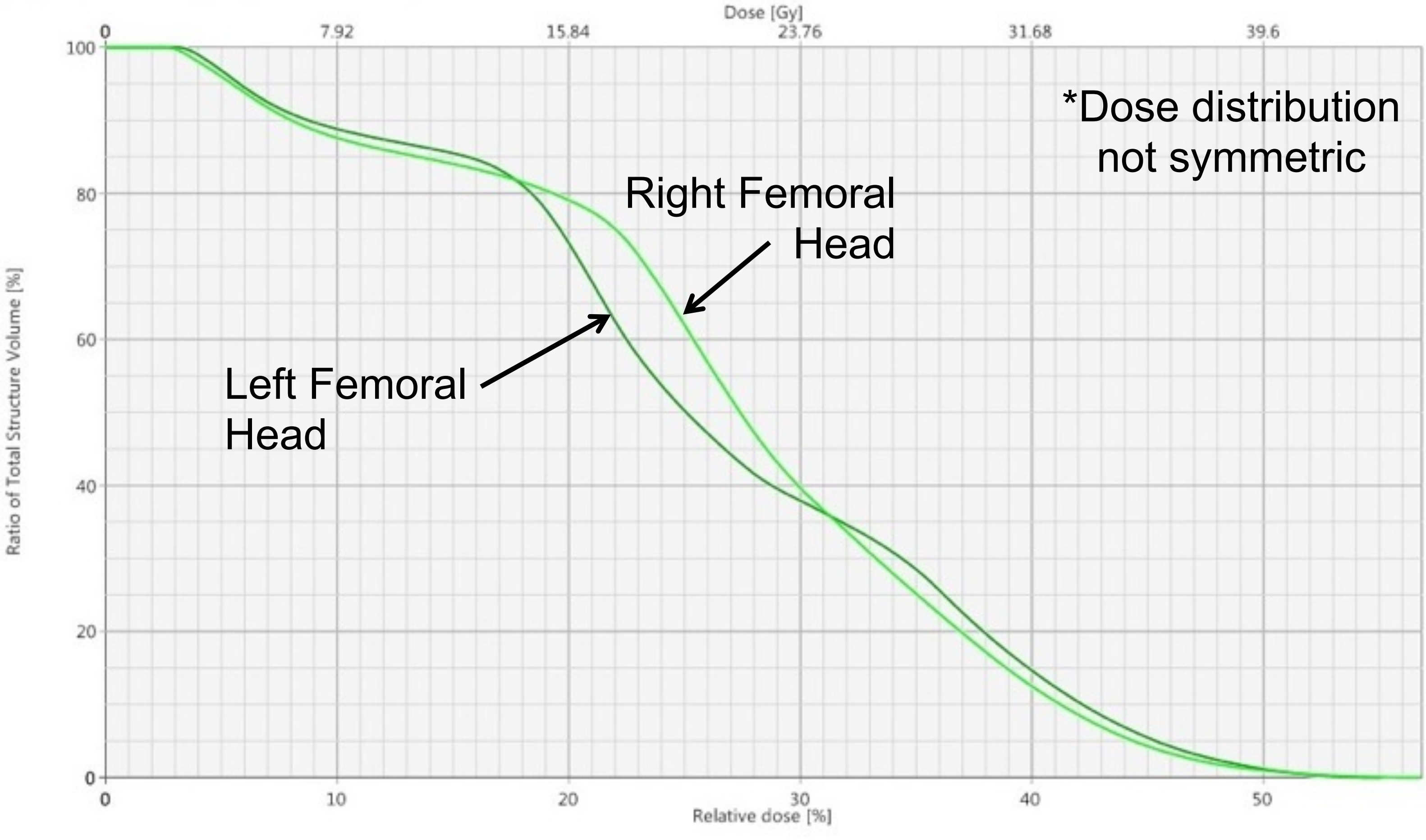
PTV DVH


Dose [Gy]
0
15.84
31.68
47.52
63.36
79.2
100
80
60
40
20
0
Ratio of Total Structure Volume [%]
0
20
40
60
80
100
Relative dose [%]
Approx. 15% of the PTV receives slightly less than the prescribed dose
Approx. 85% of the PTV receives slightly more than the prescribed dose

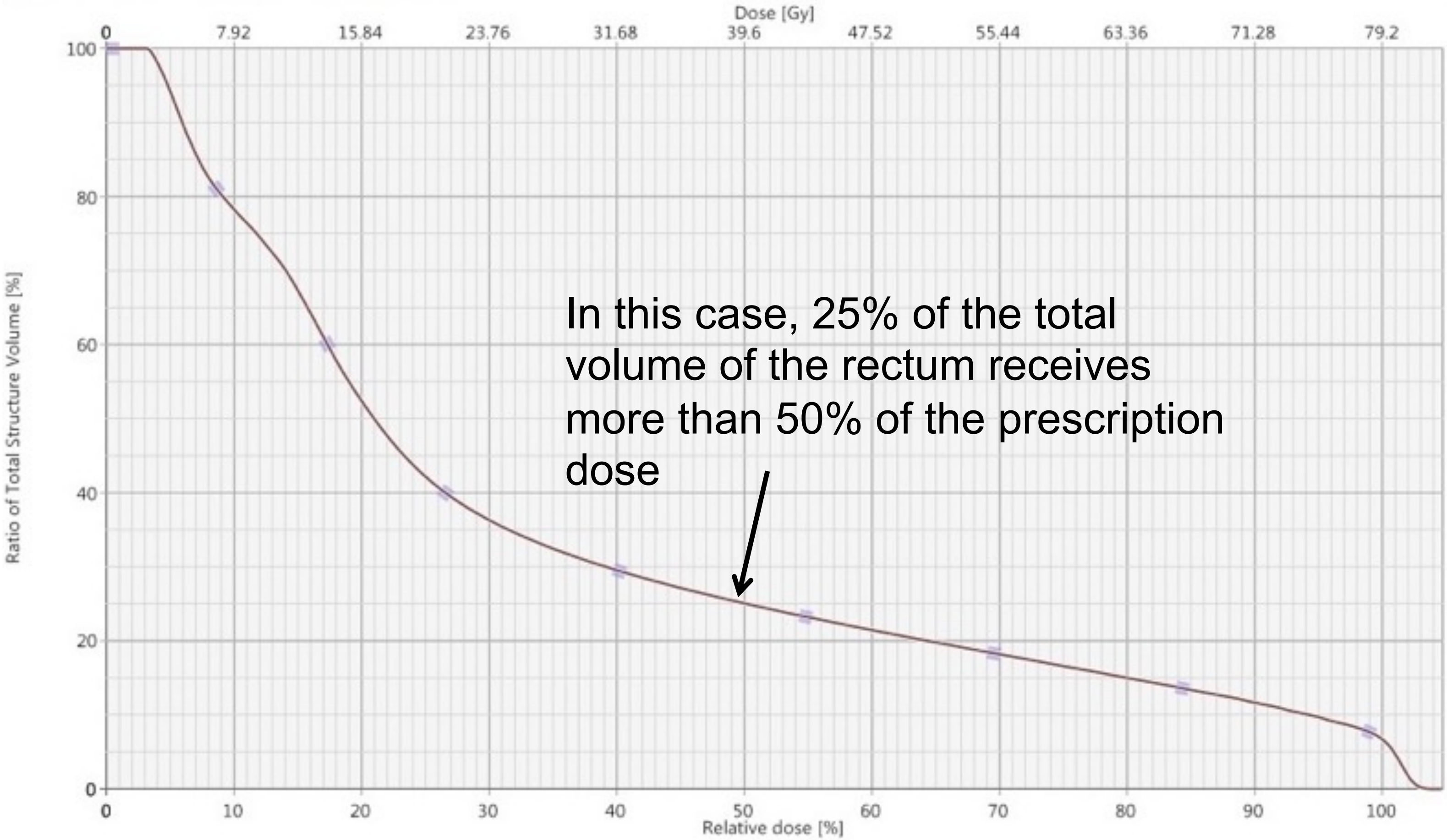
Bladder DVH


Dose [Gy]
0
7.92
15.84
23.76
31.68
39.6
47.52
55.44
63.36
71.28
79.2
Ratio of Total Structure Volume [%]
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
90
100
Relative dose [%]
The lowest (min) dose any portion of the bladder receives is 4%
40% of the total volume is receiving more than 50% of the prescription dose
The highest (max) dose any portion of the bladder receives is 102%

Fem. Heads DVH



Rectum DVH



Dose [Gy]
0
7.92
15.84
23.76
31.68
39.6
47.52
55.44
63.36
71.28
79.2
Ratio of Total Structure Volume [%]
100
80
60
40
20
0
Relative dose [%]
0
10
20
30
40
50
60
70
80
90
100
In this case, 25% of the total volume of the rectum receives more than 50% of the prescription dose

The Objective Function

$$F(\mathbf{w}) = \alpha\left(P_v - D_v(\mathbf{w})\right)^2 + \sum_i \sum_j \beta_i (\max[0, D_{ij}(\mathbf{w}) - C_{ij}])^2$$

w is a vector of beamlet weights or intensities

Minimizing $F(\mathbf{w})$ results in optimal IMRT treatment plan

The Target

$$F(\mathbf{w}) = \alpha\left(P_v - D_v(\mathbf{w})\right)^2 + \sum_i \sum_j \beta_i\left(\max\left[0, D_{ij}(\mathbf{w}) - C_{ij}\right]\right)^2$$

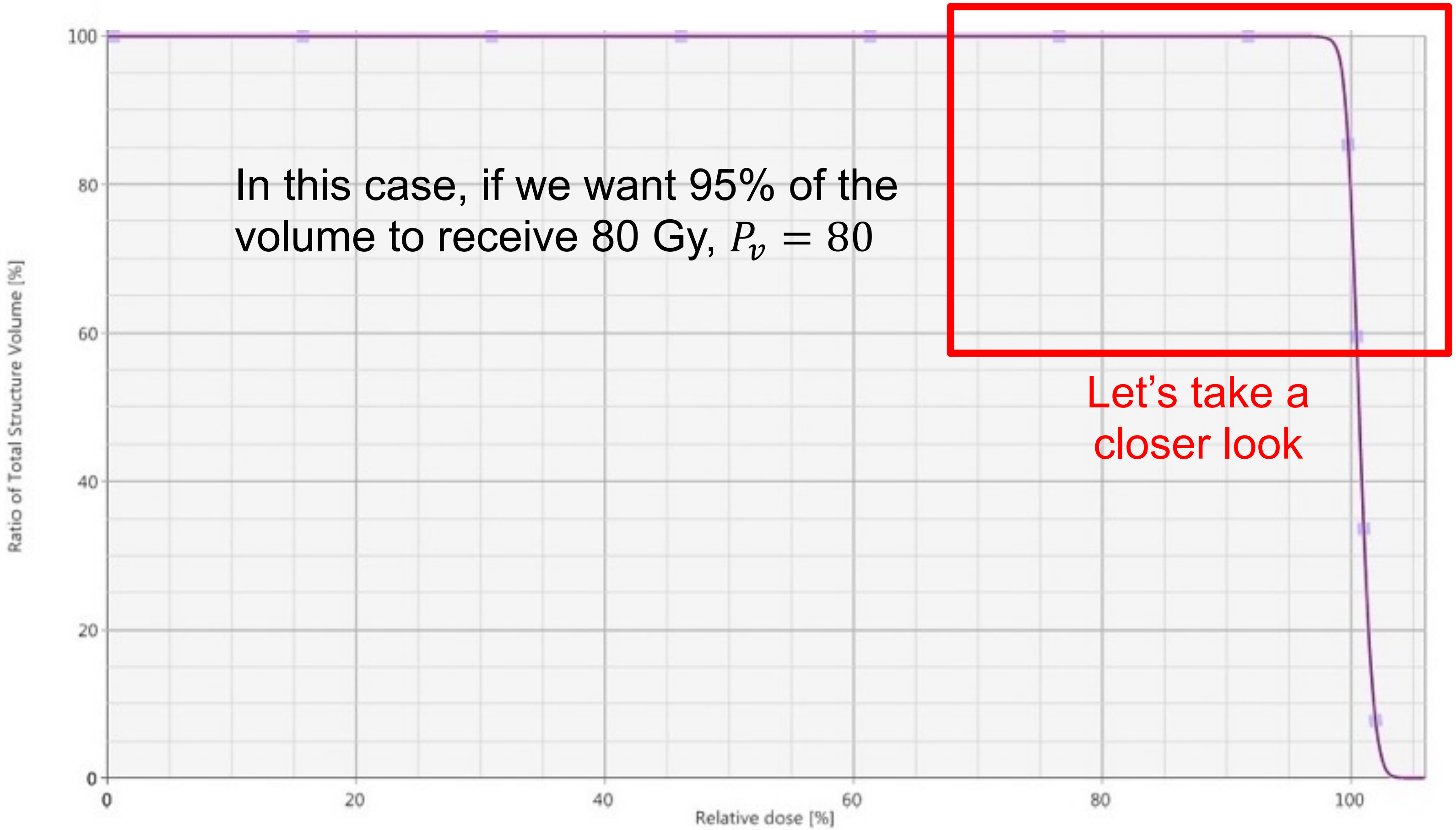
α=Priority of target dose (How important is it that this dose is fully administered?)

P_v=Dose prescribed to a given volume, v, of the target

$D_v(\mathbf{w})$=Dose actually received by volume v for weight vector $\mathbf{w}$

The Target

$$\alpha\left(P_v - D_v(\mathbf{w})\right)^2$$

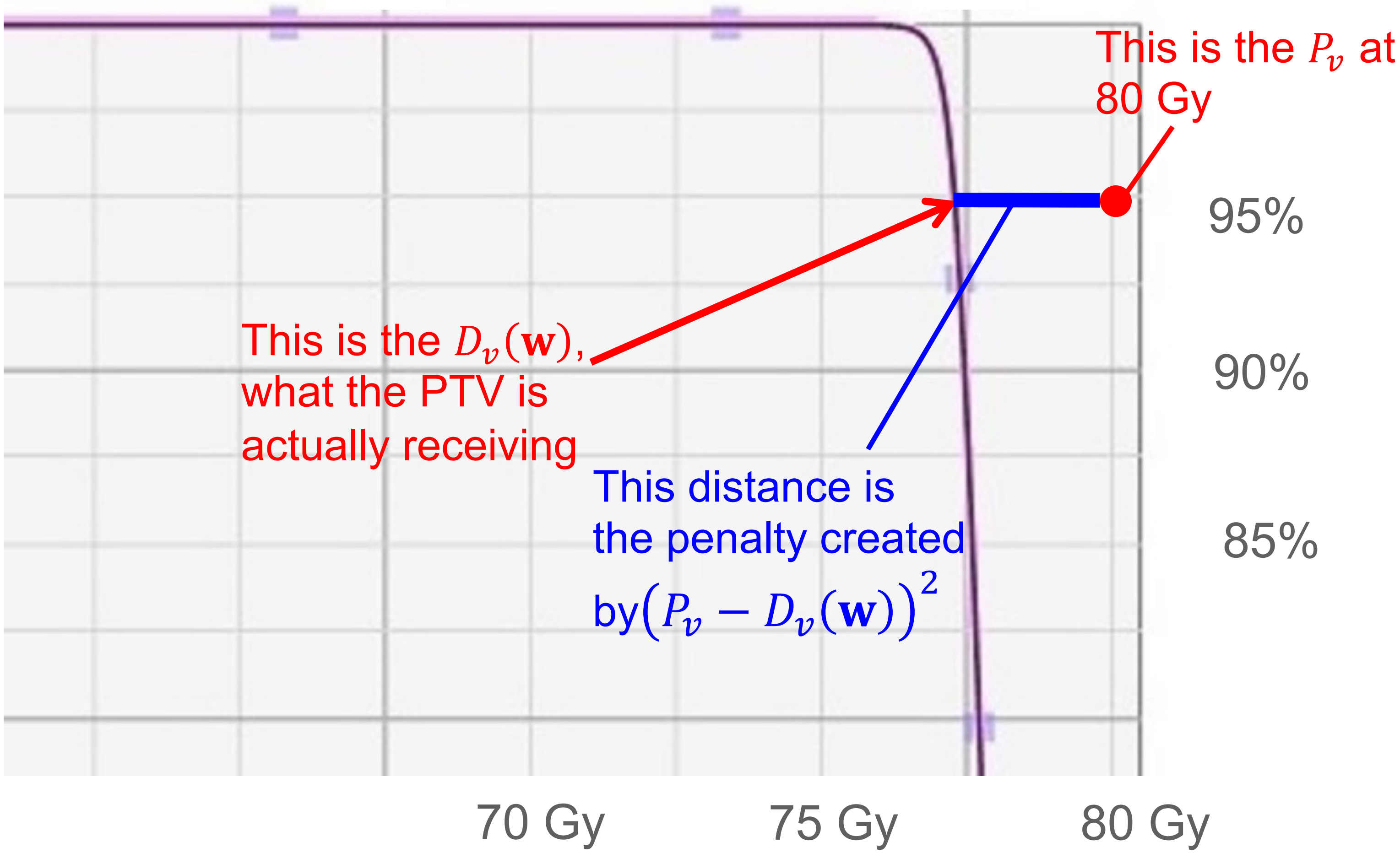


In this case, if we want 95% of the volume to receive 80 Gy, $P_v = 80$

Let's take a closer look

The Target

$$\alpha\left(P_v - D_v(\mathbf{w})\right)^2$$



Penalties

$$\alpha\left(P_v - D_v(\mathbf{w})\right)^2$$

- In clinical terms, this ensures that the dose received by the target is as close as possible to the dose prescribed

Organs at Risk (OARs)

$$F(\mathbf{w}) = \alpha\left(P_v - D_v(\mathbf{w})\right)^2 + \sum_i \sum_j \beta_i(\max[0, D_{ij}(\mathbf{w}) - C_{ij}])^2$$

$\sum_i$ =Sum over each OAR, eg. bladder = 1

$\sum_j$ =Sum over multiple objectives for a given OAR

β_i =Priority of OAR

C_{ij} =Objective dose

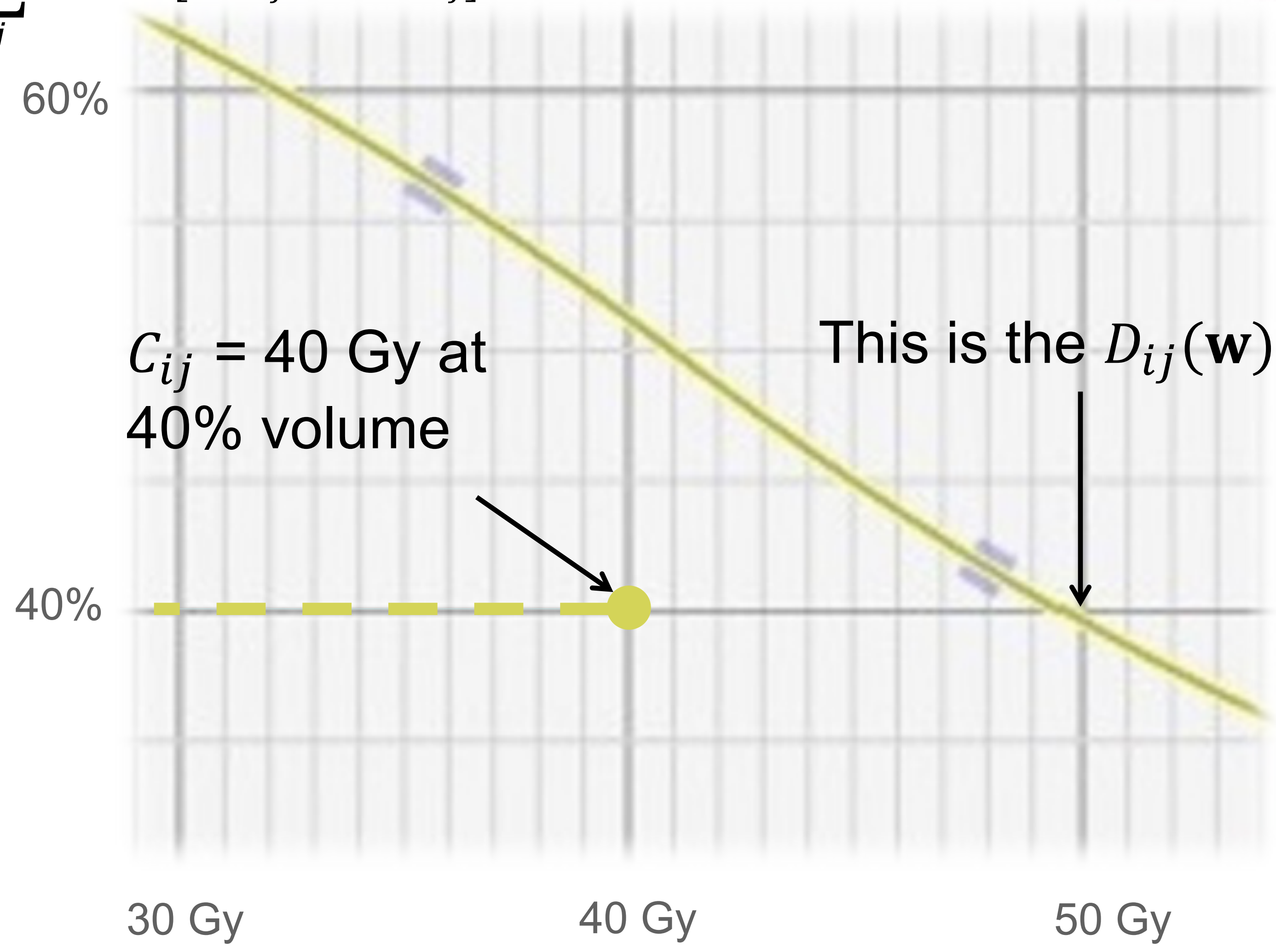
D_{ij} =Actual dose received by OAR

Organs at Risk (OARs)

$$\sum_i \sum_j \beta_i (\max[0, D_{ij}(\mathbf{w}) - C_{ij}])^2$$

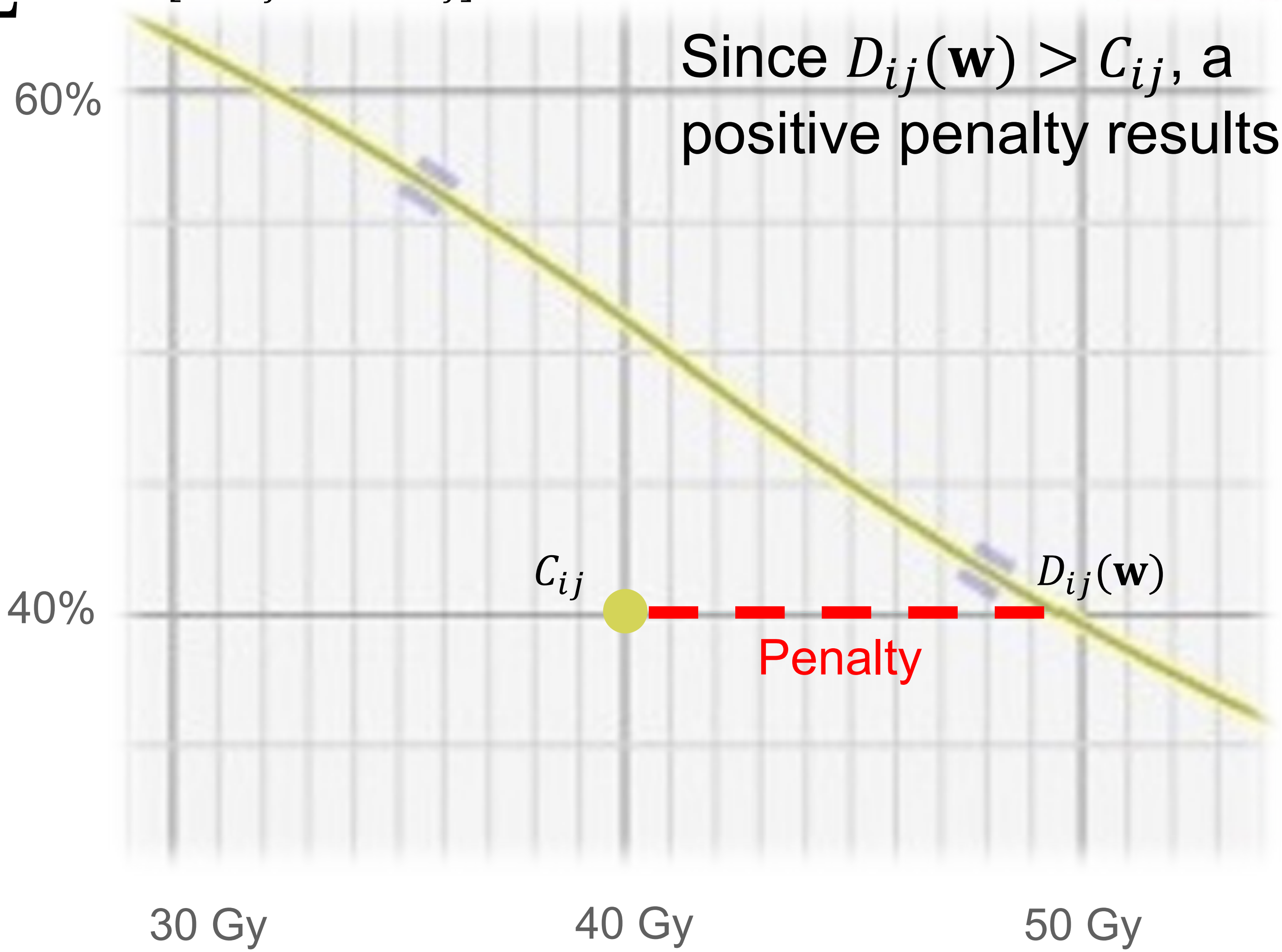
Organs at Risk (OARs)

$$\sum_i \sum_j \beta_i (\max[0, D_{ij}(\mathbf{w}) - C_{ij}])^2$$



Organs at Risk (OARs)

$$\sum_i \sum_j \beta_i \left(\max\left[0, D_{ij}(\mathbf{w}) - C_{ij}\right]\right)^2$$



Since $D_{ij}(\mathbf{w}) > C_{ij}$, a positive penalty results

Penalties

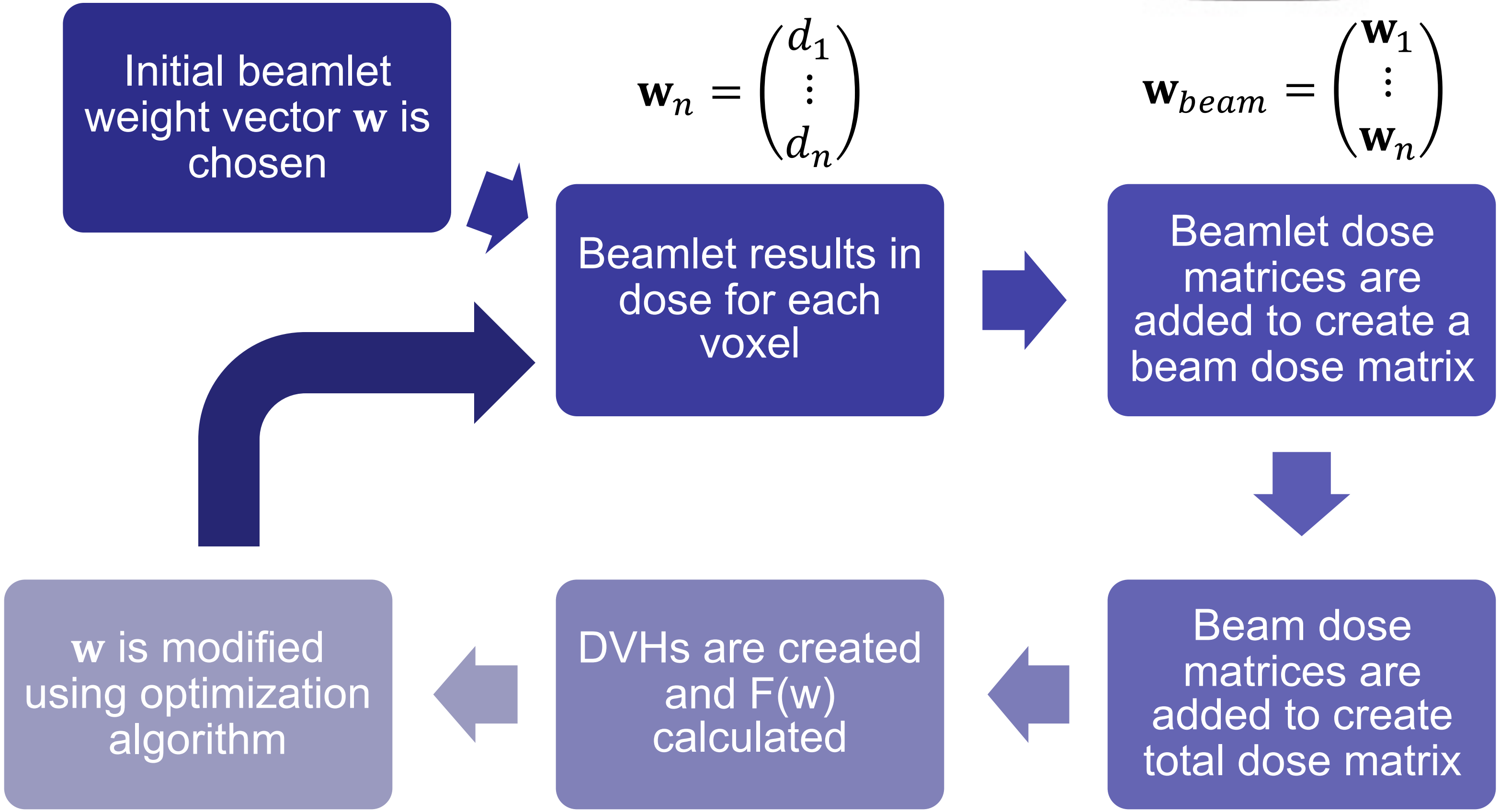
$$\sum_{i}\sum_{j}\beta_i(\max[0, D_{ij}(\mathbf{w}) - C_{ij}])^2$$

- There is no reward for $D_{ij}(\mathbf{w}) < C_{ij}$ because there is negligible clinical benefit to administering less than the objective dose to the OAR

The Process

$$\mathbf{w}_n = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}$$

$$\mathbf{w}_{beam} = \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_n \end{pmatrix}$$



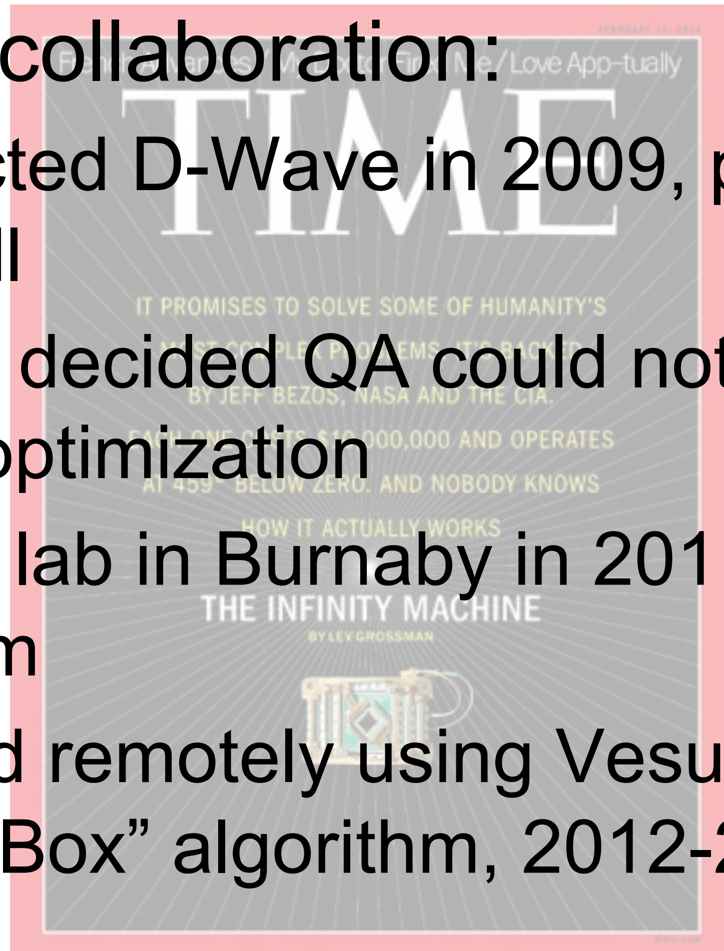
$$\mathbf{w}_{total} = \begin{pmatrix} \mathbf{w}_{beam\ 1} \\ \vdots \\ \mathbf{w}_{beam\ n} \end{pmatrix}$$

ROSWELL PARK CANCER INSTITUTE

D-Wave Systems

History of collaboration:

- Contacted D-Wave in 2009, put in touch with Bill
- Initially decided QA could not support IMRT optimization
- Visited lab in Burnaby in 2011 and revisited problem
- Worked remotely using Vesuvius chip and "Black Box" algorithm, 2012-2014



Publication in 2015

IOP Publishing | Institute of Physics and Engineering in Medicine Physics in Medicine & Biology

Phys. Med. Biol. **60** (2015) 4137–4148 doi:10.1088/0031-9155/60/10/4137

First application of quantum annealing to IMRT beamlet intensity optimization

Daryl P Nazareth and Jason D Spaans

Department of Radiation Medicine, Roswell Park Cancer Institute,
Buffalo NY 14263, US

E-mail: daryl.nazareth@roswellpark.org

Received 3 November 2014, revised 19 March 2015
Accepted for publication 2 April 2015
Published 1 May 2015

CrossMark

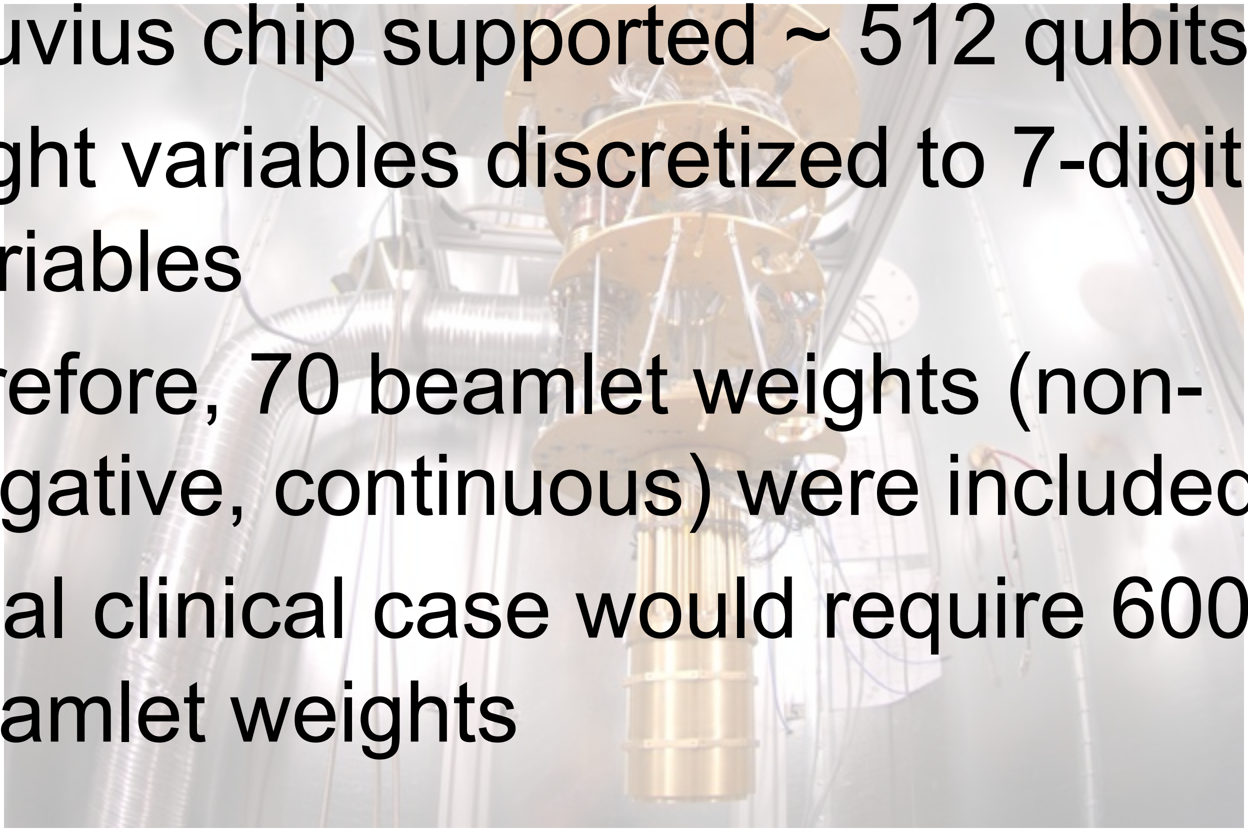
Abstract

Optimization methods are critical to radiation therapy. A new technology, quantum annealing (QA), employs novel hardware and software techniques to address various discrete optimization problems in many fields. We report on the first application of quantum annealing to the process of beamlet intensity optimization for IMRT.

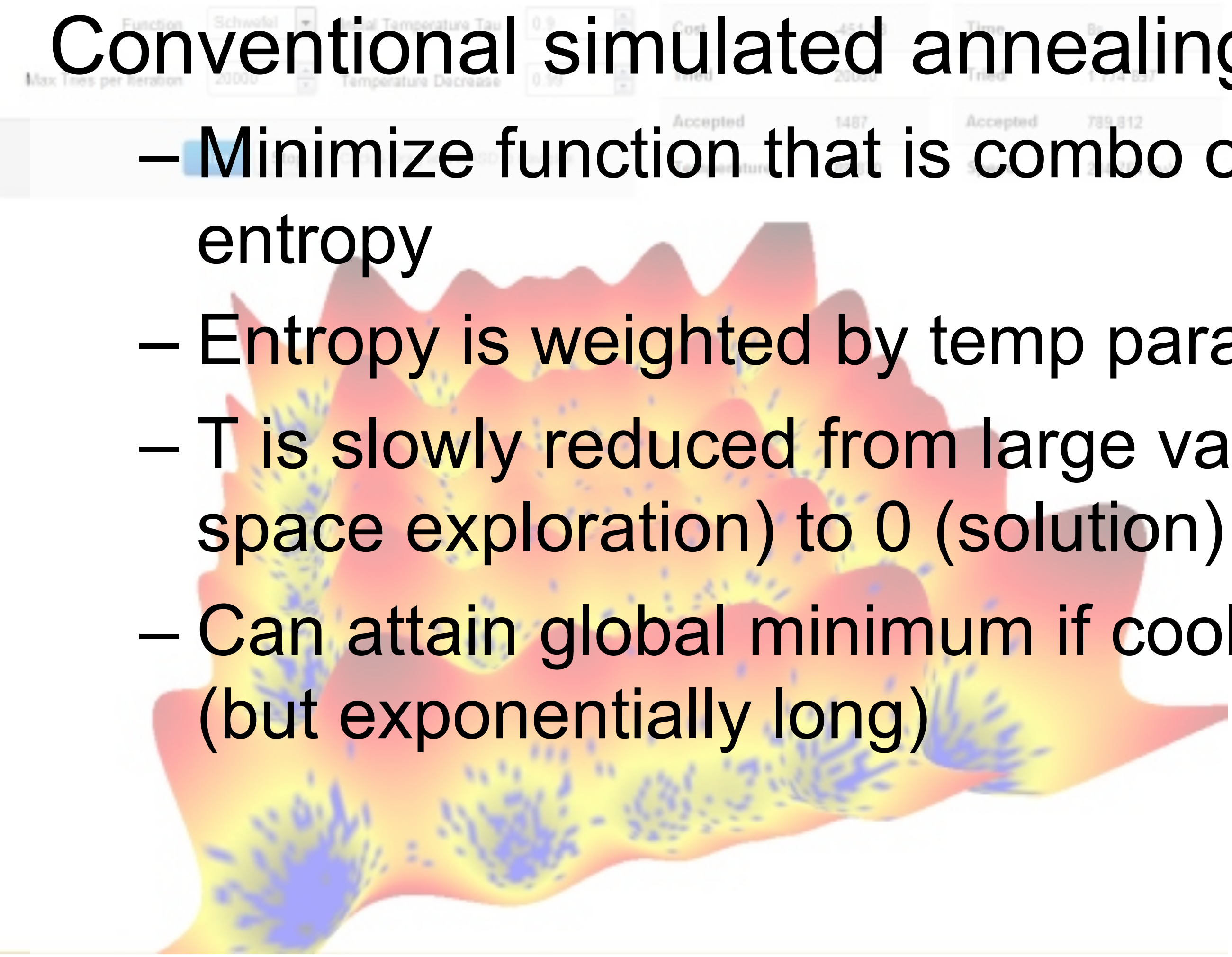
We apply recently-developed hardware which natively exploits quantum mechanical effects for improved optimization. The new algorithm, called

Applying QA

- Vesuvius chip supported ~ 512 qubits
- Weight variables discretized to 7-digit binary variables
- Therefore, 70 beamlet weights (non-negative, continuous) were included
- Actual clinical case would require 600-1000 beamlet weights



SA Algorithm



Conventional simulated annealing (SA) features:

- Minimize function that is combo of original plus entropy
- Entropy is weighted by temp parameter T
- T is slowly reduced from large values (search space exploration) to 0 (solution)
- Can attain global minimum if cooling slow enough (but exponentially long)

Evaluations

Three methods compared:

- Quantum annealing
- Simulated annealing
- Tabu search: popular heuristic used in combinatorial optimization

Methods were used to determine beamlet weights for two prostate bed cases

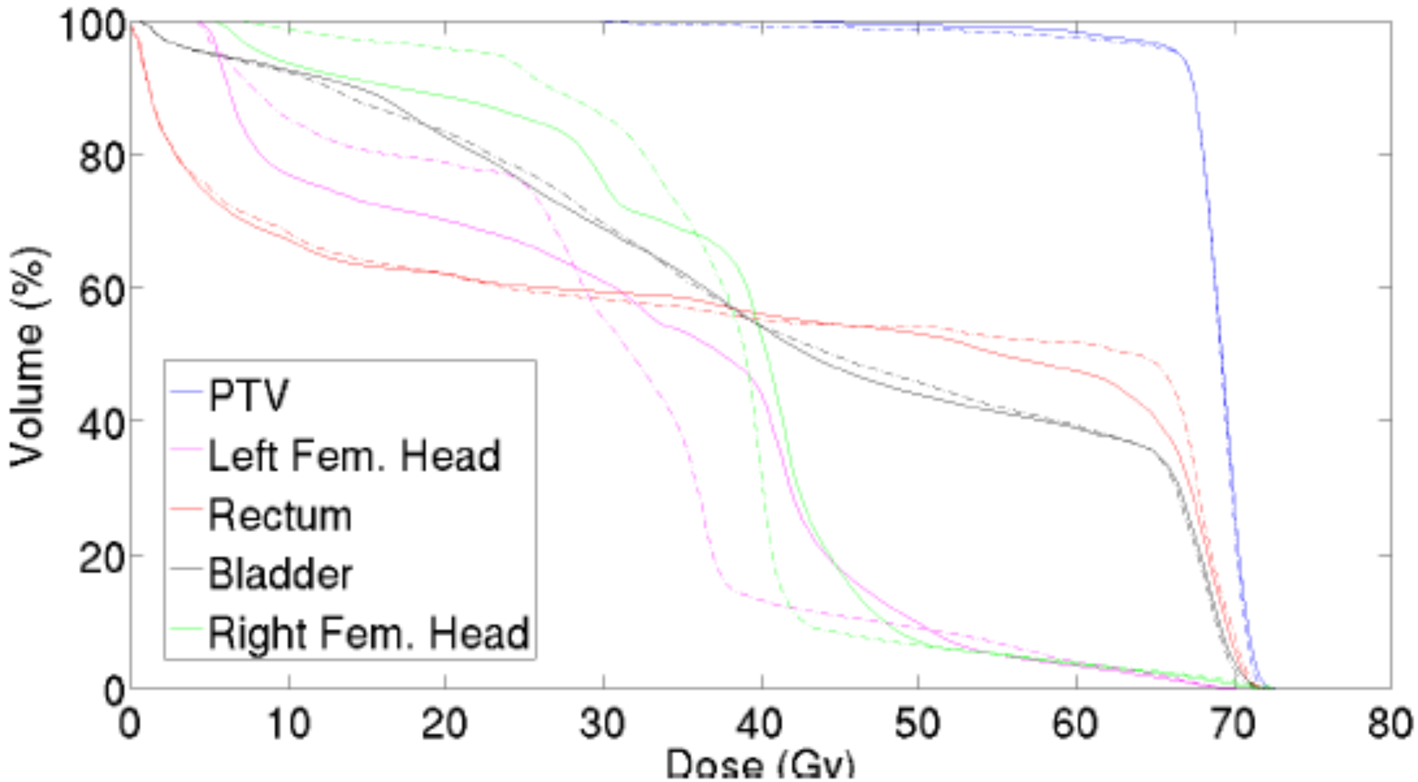
Each was run for 10^7 function evaluations and compared for speed and score

Results

Patient	Method	Evals/sec /core	Final Score
1	QA	9.3	16.9
1	SA	9.6	6.7
1	Tabu	4.3	10.0

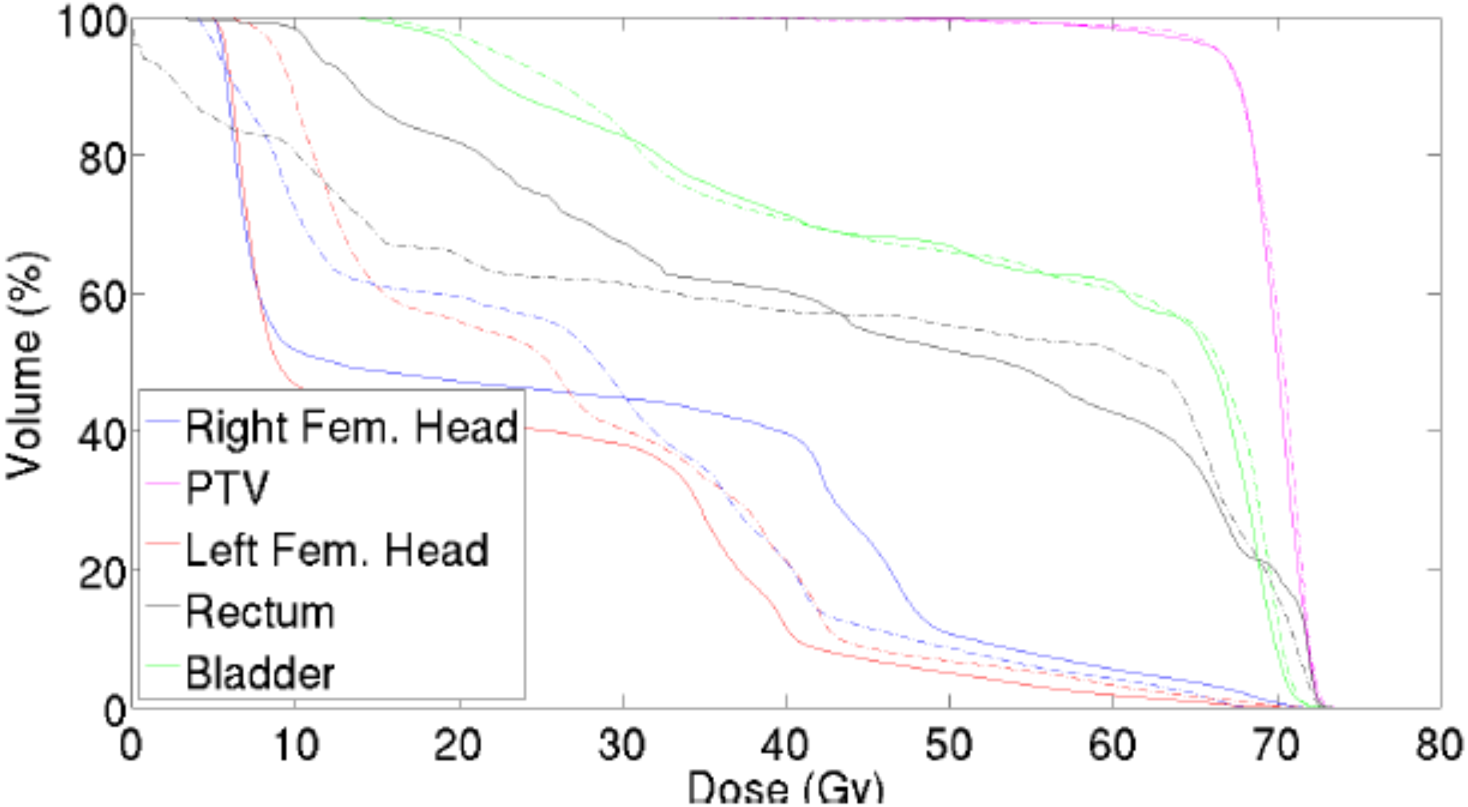
2	QA	15.4	70.7
2	SA	17.4	22.9
2	Tabu	6.3	120.0

DVHs



QA (solid) and SA (dashed) for Patient 1

DVHs

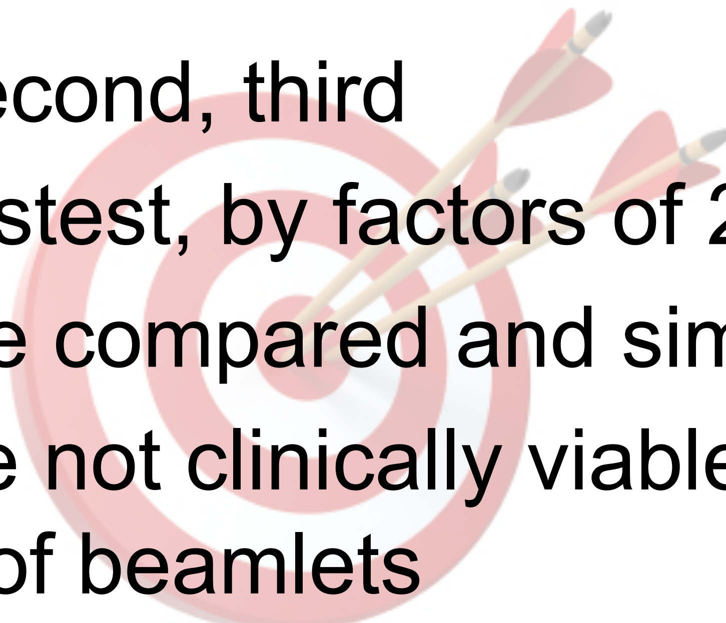


QA (solid) and Tabu (dashed) for Patient 2

Wall Clock Time

Patient	Method	Time
1	QA	1.00
1	SA	2.89
1	Tabu	3.23
2	QA	1.00
2	SA	2.67
2	Tabu	3.67

Results Summary



- SA produced best score for both patients
- QA was second, third
- QA was fastest, by factors of 2.7 – 3.7
- DVHs were compared and similar
- Plans were not clinically viable due to small number of beamlets

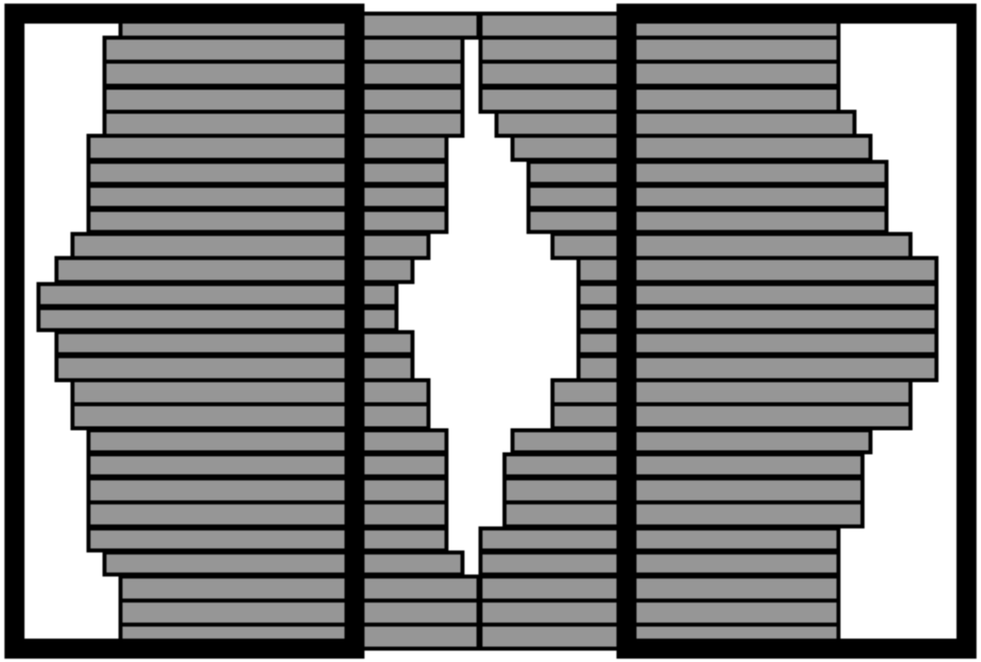
Future Work – VMAT

VMAT

VMAT

VMAT Treatment

VMAT Optimization



Conclusions

- This is first application of QA to IMRT optimization
- Compared QA to SA and Tabu
- Evaluated using clinical DVH-based objective functions
- QA hardware will rapidly scale in size
- Further research on application of QA to VMAT may offer promising returns

Thank You!

YouTube Embeds

Linear Accelerator

IMRT Treatment

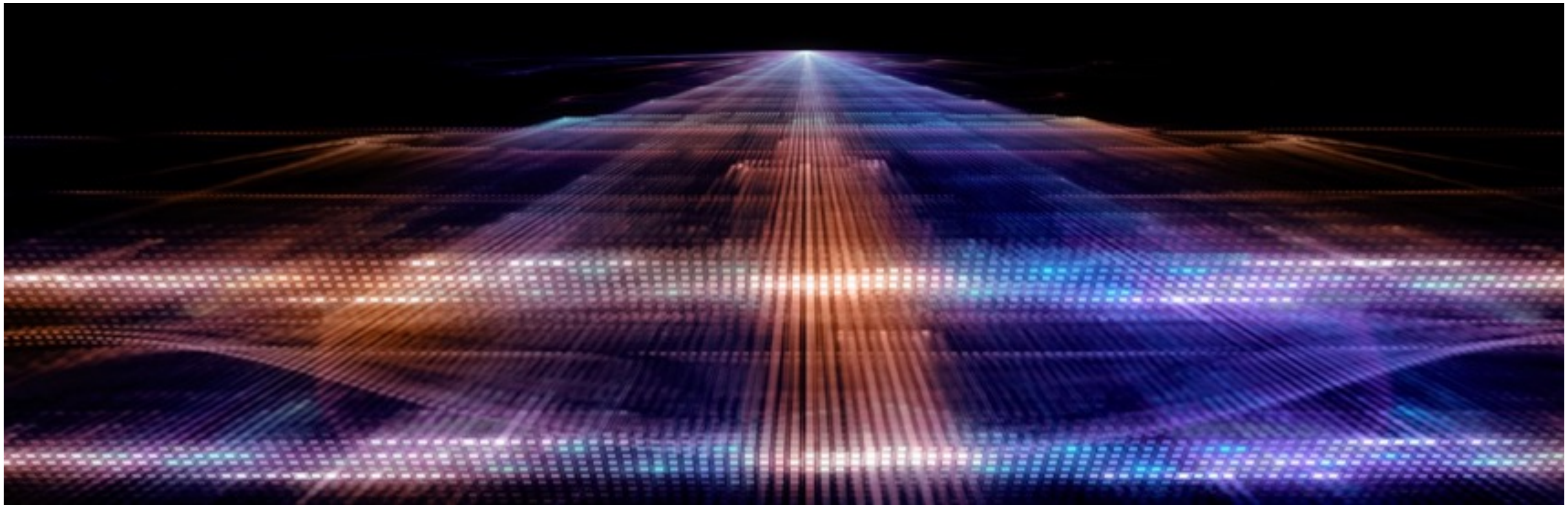
VMAT Treatment

D:Wave
The Quantum Computing Company™

QUBITS

Thoughts and Experiences

Five years of Quantum Programming



Pre-history: Conversation with Geordie Rose

- Sometime in the fall of 2011, after flubbing my first phone interview with GR, I was granted a second chance.

- I remember two questions that he asked:

What is a support vector machine?

What are the odds that a book appearing on the New York Times best seller list in the next ten years will have been written by a machine?

experience

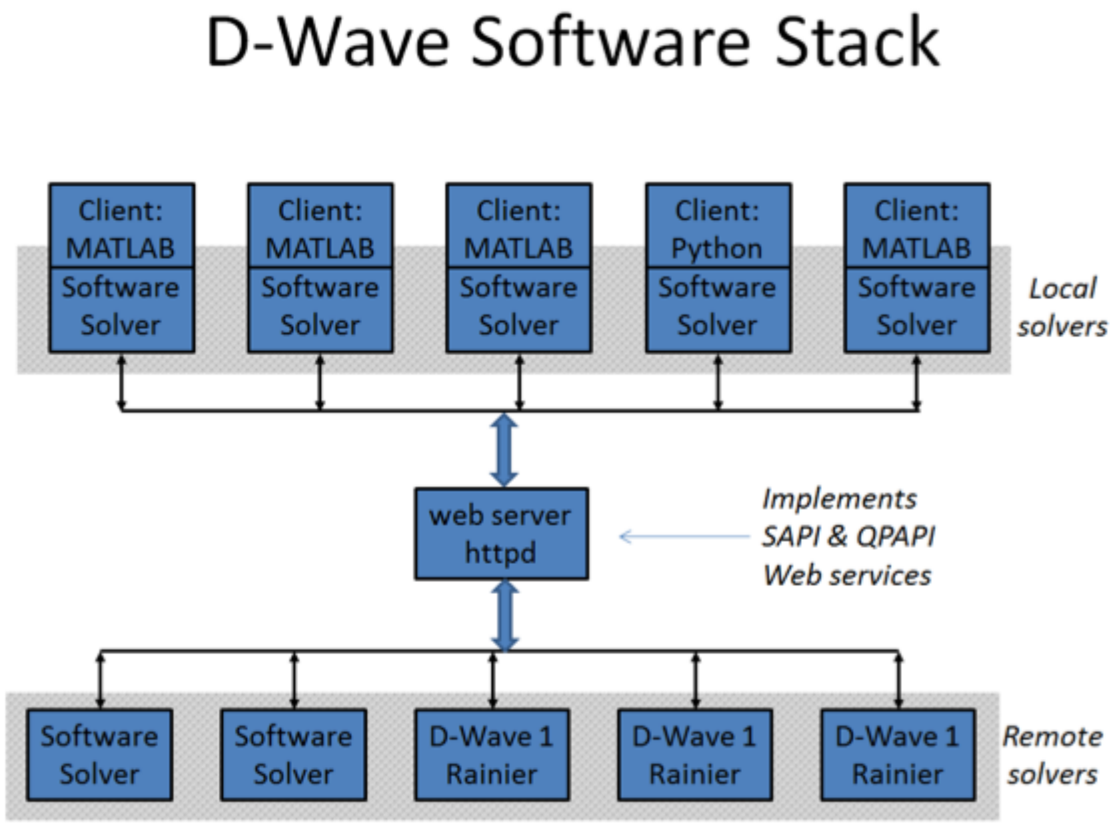
Day 1


NASA 2012
QFT 1.0
First NASA Quantum Future Technologies Conference
January 17–21, 2012
NASA Ames Research Center • Moffett Field, California

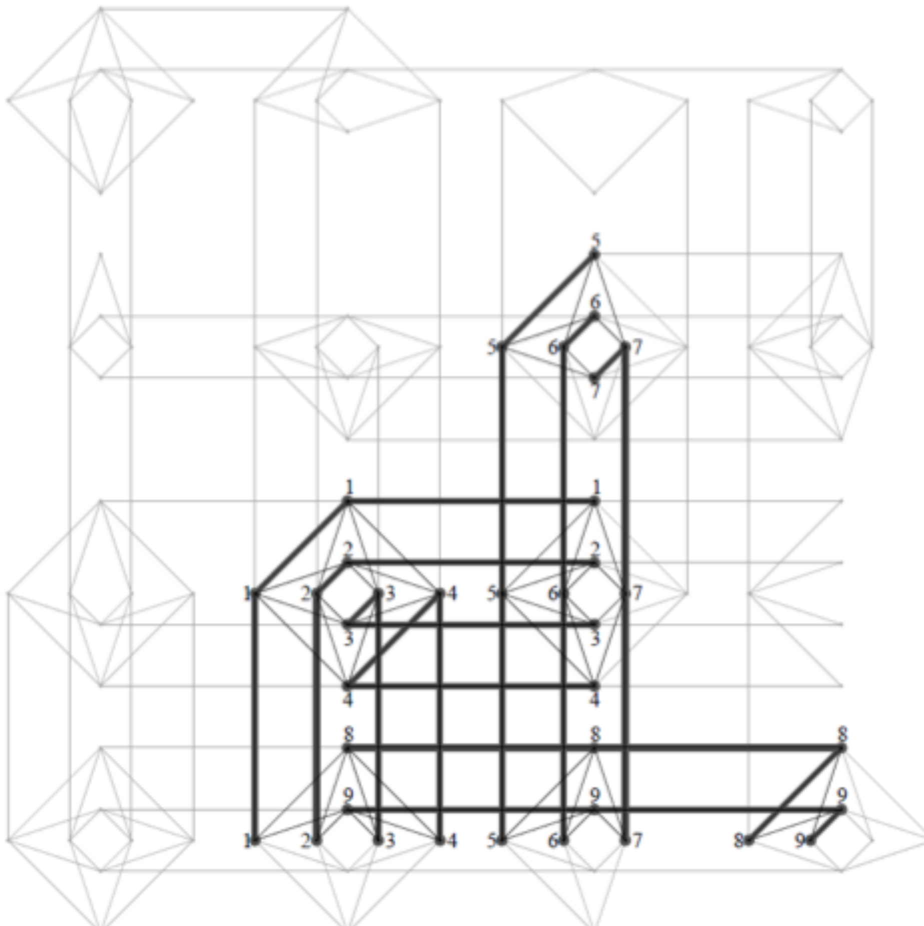
Day 14: First things first

```
emacs@DDAHL-T410
File  Edit  Options  Buffers  Tools  Help

 **************** Monday January 30, 2012 ****************

In Burnaby

----

Have downloaded emacs 23.3 onto the laptop.

----

Have downloaded the Matlab pack.

----

The URL for the machine is:

    http://appsqa.internal.dwavesys.com

The Solver API is at:

    http://appsqa.internal.dwavesys.com/sapi

The QP API (QP = Quantum Processor) is at:

    http://appsqa.internal.dwavesys.com/qpapi

-1\---  Notes.txt      Top L1      (Text Fill)--------------------------------
Mark set
```

Day 15: Early software architecture

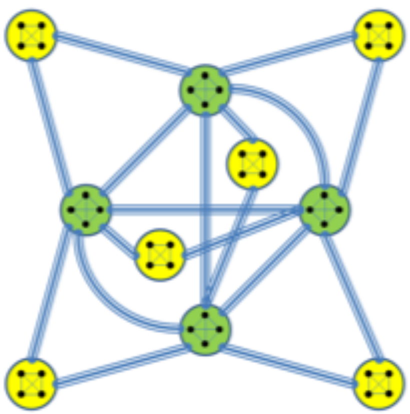


Day 15: Chimera, circa 2012



Day 21 – 22: Hadamard matrices, manually

4x4 Hadamard: QUBO interactions



Problem variables		4 vars
Ancillary variables		4 vars
Quadratic interactions		16 terms 6 terms
Totals: Variables : 4x10 = 40 Interactions : 18x16+10x6 = 348		

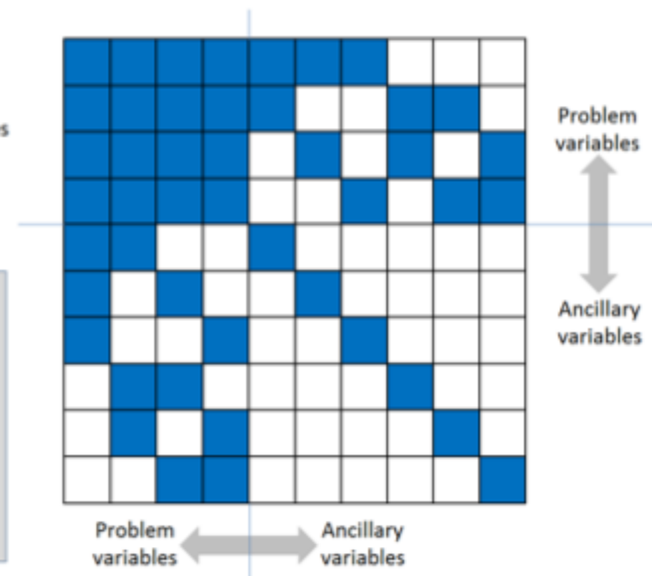
Block structure of QUBO

= 4x4 block of variables

Maximum number of interactions:
40 choose 2 = 780

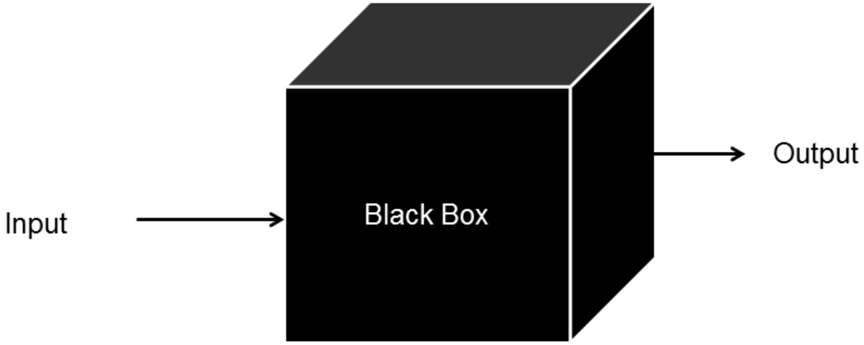
Actual number Of interactions:
348

Relative density:
348/780 = 45%



Day 23: Depression sets in

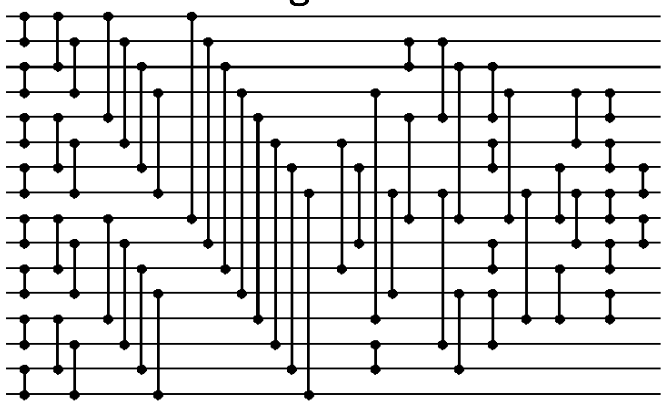
Day 24: BlackBox restores hope



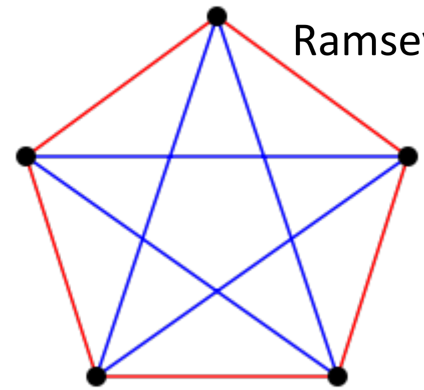
Internal behavior of the code is unknown

Day 25 – ∞: Problems. Lots of Problems.

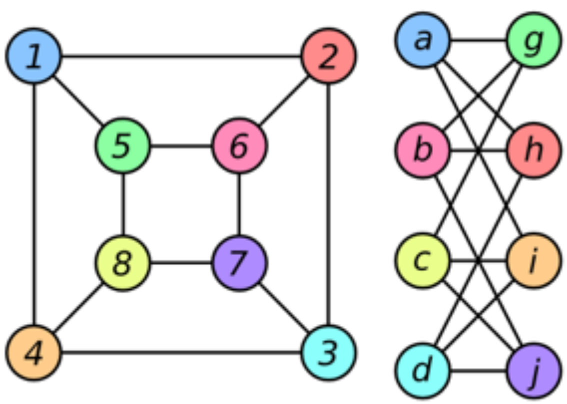
sorting networks



Ramsey theory

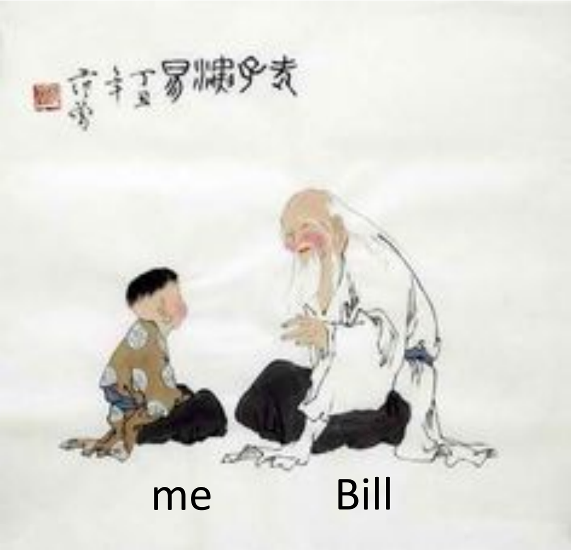


graph isomorphism



travelling salesman problem

Day 210 – 212: Learning from the Master

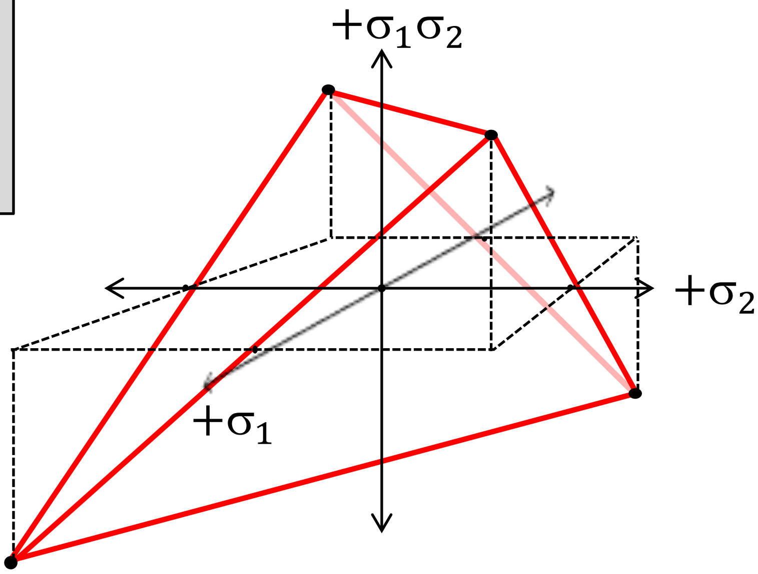


Day 250: Polytopes for Adiabatic QC

Chimeratope(L,M,N) –or– CH(L,M,N)

- L = half number of spins in the unit cell
- M = number of rows
- N = number of columns

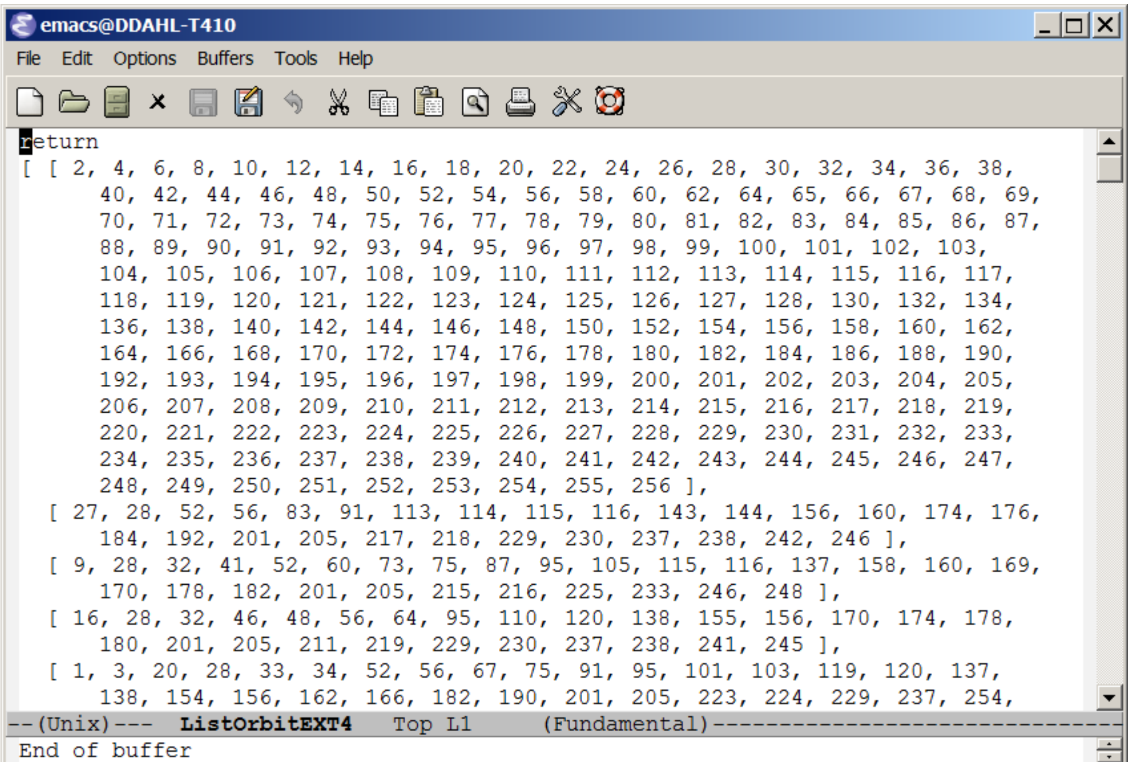
Chimeratope(1,1,1)



Day 270: Automorphism Groups of Chimera

tiling	size	factorization	group
1x1	1152	2^7 * 3^2	S(2) x S(4) x S(4)
1x2	27648	2^10 * 3^3	S(2) x S(4) x S(4) x S(4)
1x3	663552	2^13 * 3^4	S(2) x S(4) x S(4) x S(4) x S(4)
1x4	15925248	2^16 * 3^5	S(2) x S(4) x S(4) x S(4) x S(4) x S(4)
2x1	27648	2^10 * 3^3	S(2) x S(4) x S(4) x S(4)
2x2	2654208	2^15 * 3^4	D(4) x S(4) x S(4) x S(4) x S(4)
2x3	31850496	2^17 * 3^5	S(2) x S(2) x S(4) x S(4) x S(4) x S(4) x S(4)
2x4	764411904	2^20 * 3^6	S(2) x S(2) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4)
3x1	663552	2^13 * 3^4	S(2) x S(4) x S(4) x S(4) x S(4)
3x2	31850496	2^17 * 3^5	S(2) x S(2) x S(4) x S(4) x S(4) x S(4) x S(4)
3x3	1528823808	2^21 * 3^6	D(4) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4)
3x4	1.8345885696e10	2^23 * 3^7	S(2) x S(2) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4)
4x1	15925248	2^16 * 3^5	S(2) x S(4) x S(4) x S(4) x S(4) x S(4)
4x2	764411904	2^20 * 3^6	S(2) x S(2) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4)
4x3	1.8345885696e10	2^23 * 3^7	S(2) x S(2) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4)
4x4	8.8060251341e11	2^27 * 3^8	D(4) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4) x S(4)

Day 574: Conquering the Unit Cell

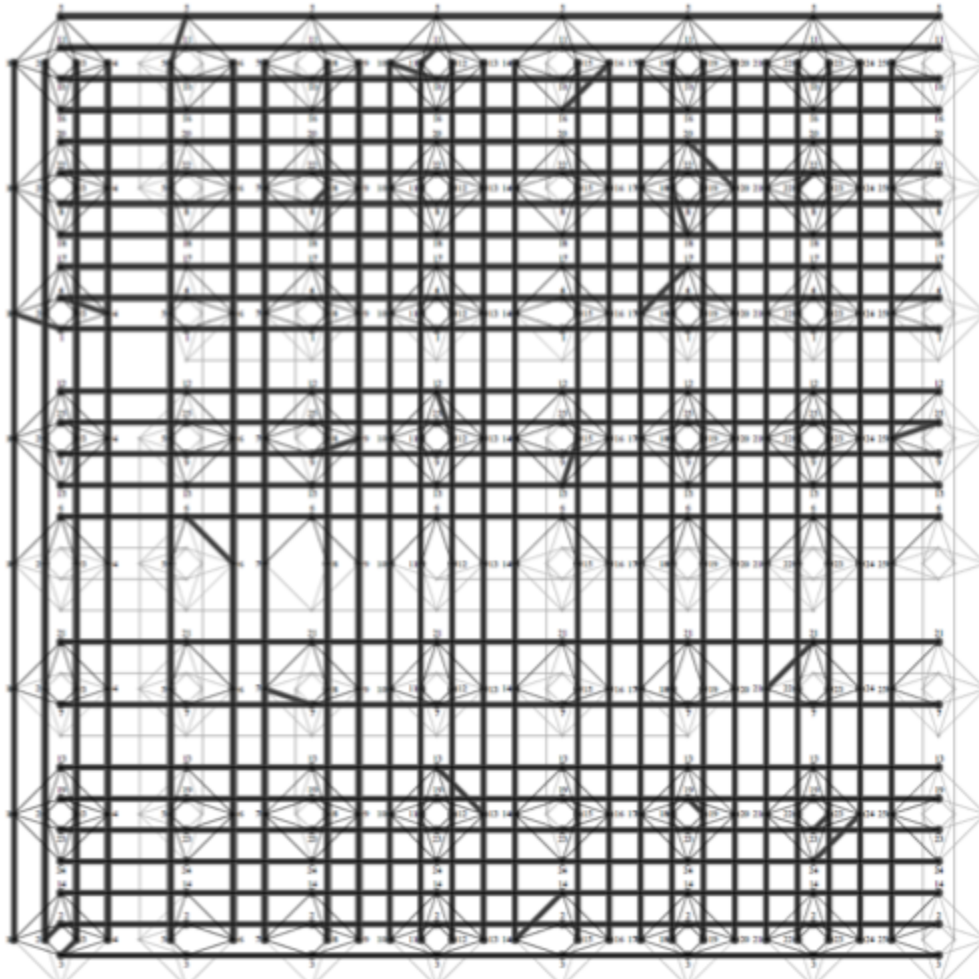


```
emacs@DDAHL-T410
File  Edit  Options  Buffers  Tools  Help
return
[ [ 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38,
      40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 65, 66, 67, 68, 69,
      70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87,
      88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103,
      104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117,
      118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 130, 132, 134,
      136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162,
      164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190,
      192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205,
      206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219,
      220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233,
      234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247,
      248, 249, 250, 251, 252, 253, 254, 255, 256 ],
  [ 27, 28, 52, 56, 83, 91, 113, 114, 115, 116, 143, 144, 156, 160, 174, 176,
      184, 192, 201, 205, 217, 218, 229, 230, 237, 238, 242, 246 ],
  [ 9, 28, 32, 41, 52, 60, 73, 75, 87, 95, 105, 115, 116, 137, 158, 160, 169,
      170, 178, 182, 201, 205, 215, 216, 225, 233, 246, 248 ],
  [ 16, 28, 32, 46, 48, 56, 64, 95, 110, 120, 138, 155, 156, 170, 174, 178,
      180, 201, 205, 211, 219, 229, 230, 237, 238, 241, 245 ],
  [ 1, 3, 20, 28, 33, 34, 52, 56, 67, 75, 91, 95, 101, 103, 119, 120, 137,
      138, 154, 156, 162, 166, 182, 190, 201, 205, 223, 224, 229, 237, 254,
--(Unix)---  ListOrbitEXT4    Top L1     (Fundamental)------------------------
End of buffer
```

Mathieu Dutour Sikiric

Institut Ruđer Bošković

Day 370: Vesuvius



Day 395: Big B


Linux Networx


UNITED STATES AIR FORCE
GATE 1 C
WRIGHT - PATTERSON AIR FORCE BASE

Day 455 – ∞: Training

LOCKHEED MARTIN

USRA Universities Space Research Association

Google

D-Wave Quantum Computer Training

April 3 -5, 2013

Location: University of Southern California, Information Sciences Institute (ISI)
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292

Attendees: Various Australian Small-Medium Enterprises

OAK RIDGE National Laboratory

NAVAL AIR SYSTEMS COMMAND

NASA

UNIVERSITY OF MARYLAND BALTIMORE COUNTY UMBC 1966

U.S. NAVAL RESEARCH LABORATORY

Stanford University

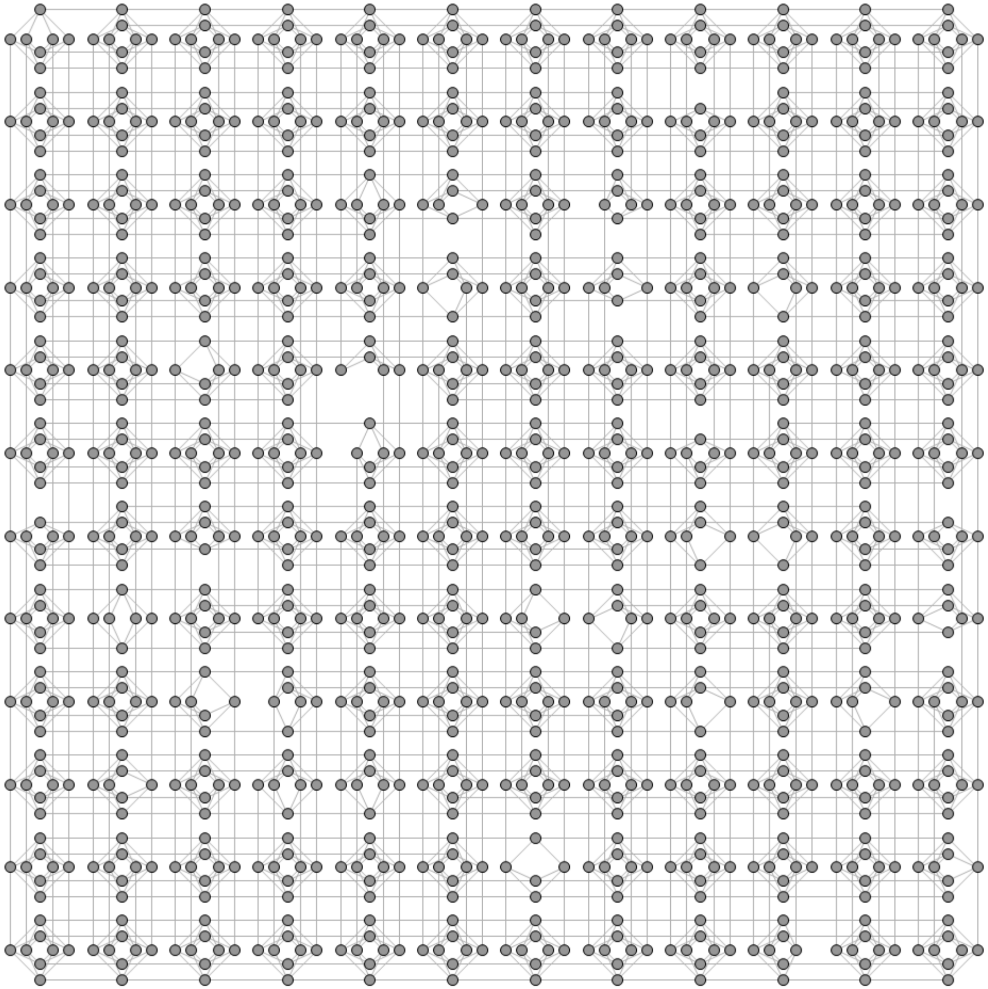
Los Alamos NATIONAL LABORATORY EST. 1943

TDS TEMPORAL DEFENSE SYSTEMS

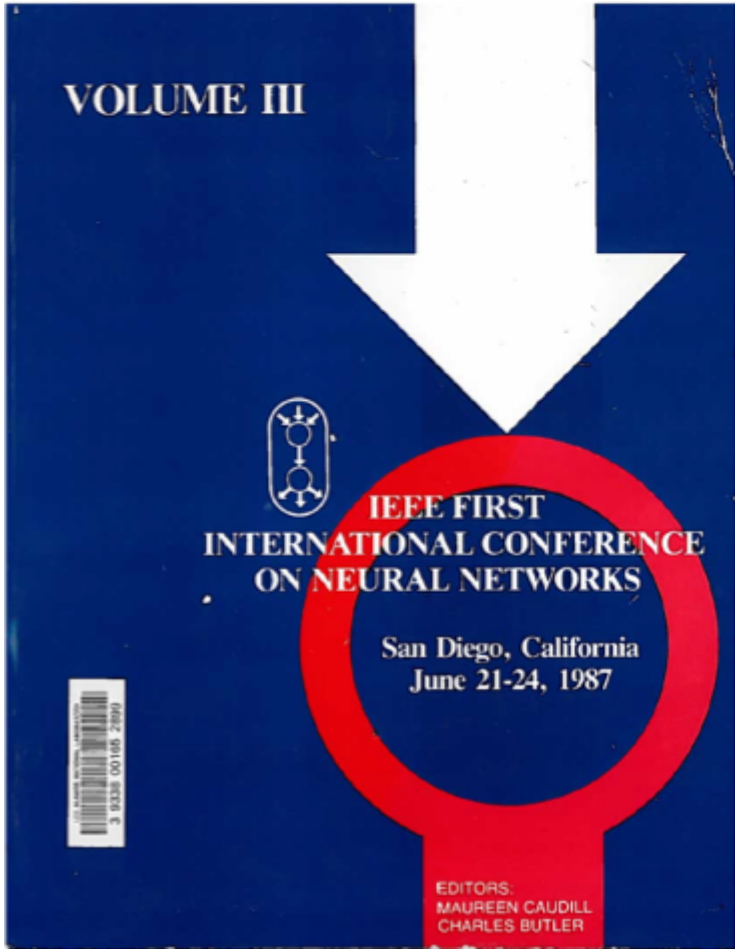
GENERAL DYNAMICS

AFRL

Day 720: Washington



Day −8760: Archaeology of Map Coloring



Neural Network Algorithm for an NP-Complete Problem: Map and Graph Coloring

Edward Denning Dahl, L-298

Lawrence Livermore National Laboratory
P. O. Box 808
Livermore, CA 94550

1. Introduction

The four color conjecture[1] states that any map drawn on a plane or sphere can be colored with four colors so that no two countries which share a border have the same color. The proof of this appealing conjecture took more than one hundred years, and on the order of 10^{10} computer operations.[2] Unfortunately, the proof of this conjecture is not constructive. In fact, there is no algorithm which is guaranteed to color an arbitrary planar map without essentially resorting to exhaustive search. The absence of such an algorithm extends to a class of problems which generalizes the task of coloring a planar map with four colors. In the general problem, the map is not necessarily planar, but may lie on a more complicated surface such as a torus with several holes. To complete the specification of the general problem, pick a positive integer K which sets the number of colors available for use in coloring the map. The problem is to decide whether a K-coloring of the map exists: that is, an assignment of colors to regions on the map so that regions sharing a border receive different colors.

Since 1971 and the proof of Cook's Theorem,[3] the notion of NP-Completeness has been made precise. For a problem to be NP-Complete it is required that one be able to find the solution to the problem in an amount of time polynomial in the problem size, provided one uses the very special model of computation represented by a non-deterministic Turing Machine. Secondly, an NP-Complete problem is at least as hard as any problem which satisfies the first criteria. Deciding whether a non-planar map is K-colorable is an NP-Complete Problem.[4]

John Hopfield has demonstrated[5] that a Neural Network can provide a heuristic technique for solving the Traveling Salesman Problem (TSP). In one version of the Traveling Salesman Problem one is given some number of cities, the pairwise distances between all the cities, and some fixed length L. One must then decide whether there is some tour through the cities which visits each one once, and has total length less than the bound L. This problem is NP-Complete,[6] and thus equivalently difficult to the Map K-colorability problem.

It is natural to wonder whether a Neural Network solution exists for the Map K-colorability problem, given that both it and the TSP are equivalently hard and that a Neural Net solution exists for the latter problem. We demonstrate that a Neural Network solution does exist for the Map K-colorability problem. The connectivity of the neurons in the net follows simply from the connectivity of the regions in the map. The dynamics of the

---

* Work performed under the auspices of the U. S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

1 K. O. May, *The Origin of the Four Color Conjecture*. Isis, vol. 56, pp. 346-348 (1965).

2 T. L. Saaty and P. C. Kainen, *The Four-Color Problem*. McGraw-Hill, Inc. (1977).

3 S. A. Cook, *Proc. 3rd Ann. ACM Symp. on Theory of Computing*. New York, 151-158 (1971).

4 R. M. Karp, *Complexity of Computer Computations*. R. E. Miller and J. W. Thatcher (eds.), Plenum Press, New York, 85-103 (1972).

5 J. J. Hopfield and D. W. Tank, *Biol. Cyber.*, Vol. 52, 141 (1985).

6 M. R. Garey and D. S. Johnson, *Computers and Intractibility*. Freeman and Company, San Francisco (1979).

D-Wave
The Quantum Computing Company™

QUBITS

Day 580: Map coloring on a quantum computer

Canada regions/shell script

(Not to scale)

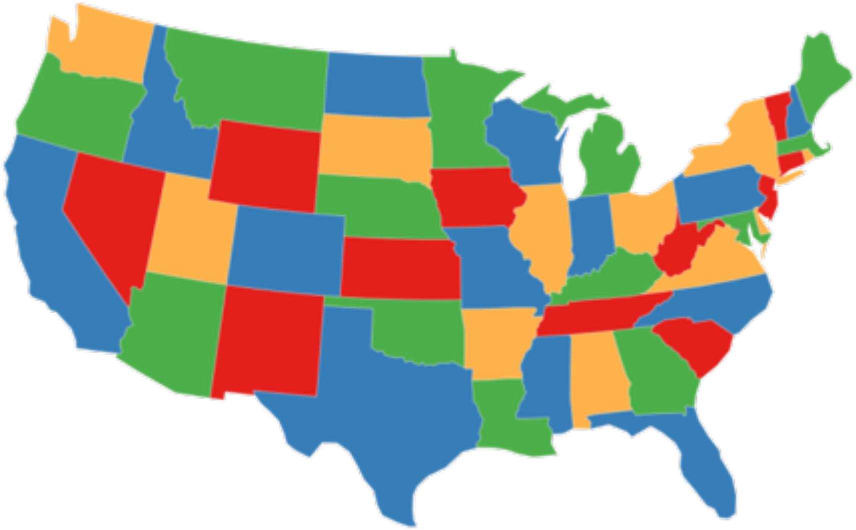
# of colors	Needle	Haystack	N/H
3	1728	$3^{13} = 1.6\text{x}10^{6}$	0.0011
4	653184	$4^{13} = 6.7\text{x}10^{7}$	0.0097

D-Wave The Quantum Computing Company™

QUBITS

Day 610: Static decomposition

US states/MATLAB

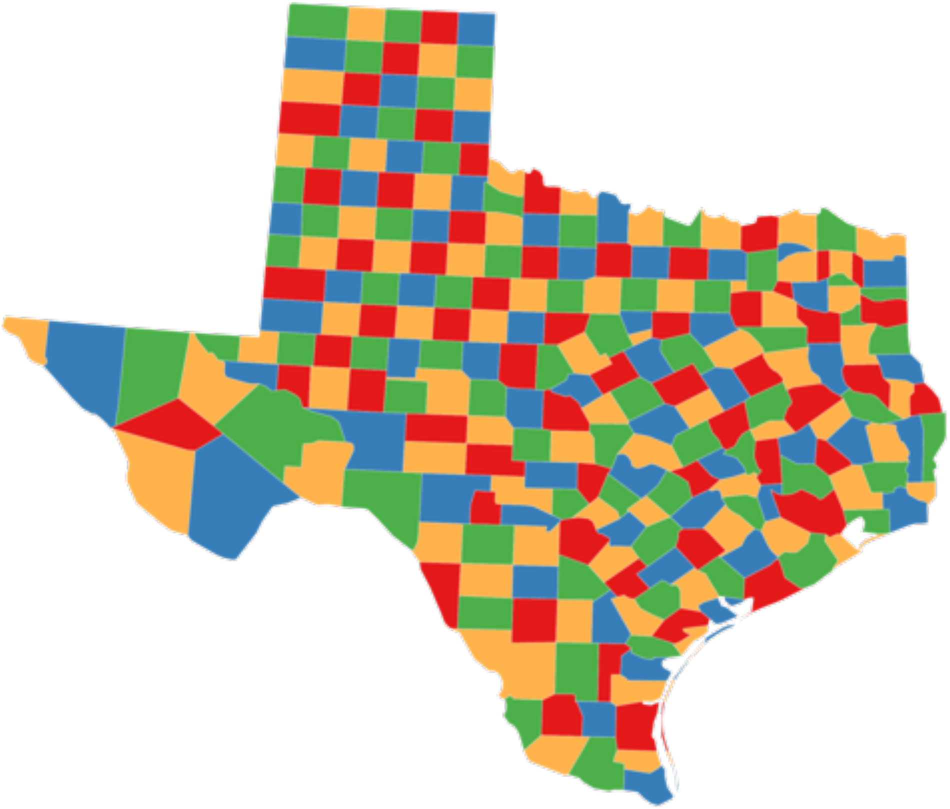


# of colors	Needle	Haystack	N/H
3	0	$3^{49} = 2.4\text{x}10^{23}$	0
4	25623183458304	$4^{49} = 3.2\text{x}10^{29}$	$8\text{x}10^{-17}$

D-WAVE The Quantum Computing Company™

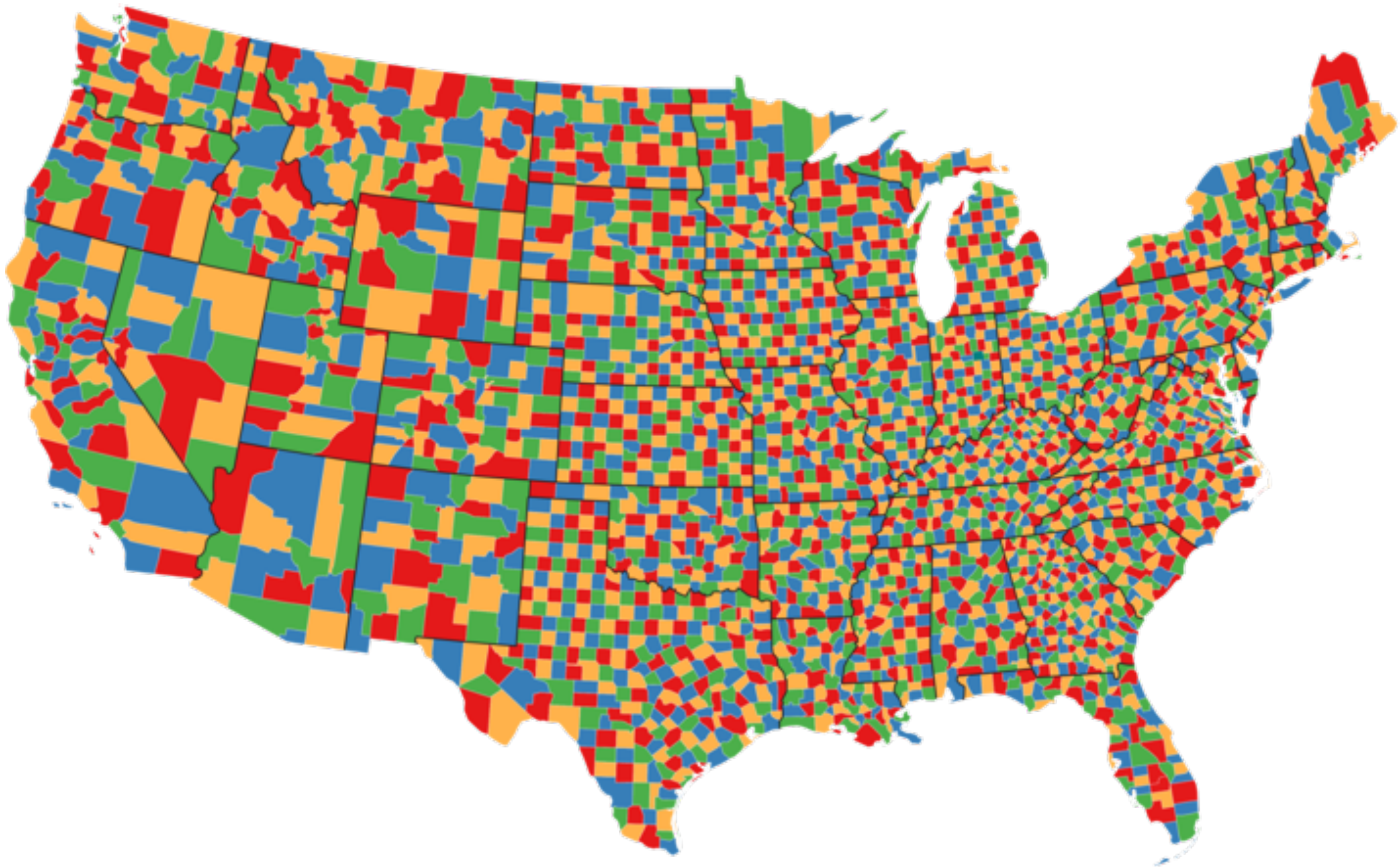
QUBITS

Day 720: Dynamic decomposition / qbsolv



254 counties in Texas

Day 900: Dynamic decomposition / qbsolv



3108 US counties

Day 735: DEQO – the predecessor of ToQ

Deqo: A Direct Embedding Quantum Optimizer

E. D. Dahl, D-Wave Systems

January 24, 2014

Abstract

Deqo is a prototype compiler for the D-Wave System. It implements a simple language designed for constraint satisfaction problems (CSP) defined over both boolean and small integer variables. Deqo assumes its CSP has locality: in other words, it consists of a set of constraints each of which involves a small number of available variables. The compilation model proceeds by applying a sequence of transformations which move the problem representation closer to the native quantum machine instruction (QMI) of the D-Wave System. Each transformation is described in the context of a simple example. We conclude by discussing possible extensions to the compilation model.

Day 950 : Map coloring made easy / ToQ

C

```c
void setup_unit_cell(int row, int col)
{
  int i, j;

  if (cell_region[row][col] == UNDEF)
    return;

  /* STEP 1: turn on one of C qubits */

  for (i=0; i<C; ++i)
    {
      weight[DW_QUBIT(row,col,'L',i)] += -0.5;
      weight[DW_QUBIT(row,col,'R',i)] += -0.5;
    }

  for (i=0; i<C; ++i)
    for (j=0; j<C; ++j)
      if (i != j)
        strength[DW_INTRACELL_COUPLER(row,col,i,j)] += 1;

  /* STEP 2: create chains */

  for (i=0; i<C; ++i)
    {
      weight[DW_QUBIT(row,col,'L',i)] += 1;
      weight[DW_QUBIT(row,col,'R',i)] += 1;
      strength[DW_INTRACELL_COUPLER(row,col,i,i)] += -2;
    }
```

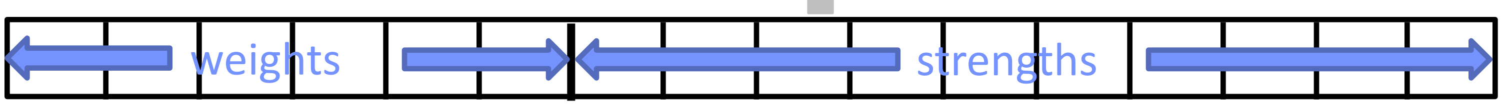
Snippet (28 of 596 LOC)

ToQ

```
mbool: 1, 4, @AB
mbool: 1, 4, @BC
mbool: 1, 4, @MB
mbool: 1, 4, @NB
mbool: 1, 4, @NL
mbool: 1, 4, @NS
mbool: 1, 4, @NT
mbool: 1, 4, @NU
mbool: 1, 4, @ON
mbool: 1, 4, @QC
mbool: 1, 4, @SK
mbool: 1, 4, @YT

assert: @AB != @BC
assert: @AB != @NT
assert: @AB != @SK
assert: @BC != @NT
assert: @BC != @YT
assert: @MB != @NU
assert: @MB != @ON
assert: @MB != @SK
assert: @NB != @NS
assert: @NB != @QC
assert: @NL != @QC
assert: @NT != @NU
assert: @NT != @SK
assert: @NT != @YT
assert: @ON != @QC
```

entire program

QMI :



D-Wave The Quantum Computing Company™

QUBITS

Day 1165: Sudoku

		3		9	2			
4				3			1	
2	7							
	1		3					8
	5		1	6	7		3	
3					8		6	
							5	3
	3			8				9
			6	2		1		

bash

```
Documents> qbsolv -h

        qbsolv -i infile [-o outfile] [-m] [-T] [-n] [-S SubMatrix] [-w]
                [-h] [-v verbosityLevel] [-V] [-q]

DESCRIPTION
        qbsolv executes a quadratic unconstrained binary optimization
        (QUOB) problem represented in a file, providing bit-vector
        result(s) that minimizes (or optionally, maximizes) the value of
        the objective function represented by the QUBO.  The problem is
        represented in the QUBO(5) file format and notably is not limited
        to the size or connectivity pattern of the D-Wave system on which
        it will be executed.
                The options are as follows:
        -i infile
                The name of the file in which the input QUBO resides.  This
                is a required option.
        -o outfile
                This optional argument denotes the name of the file to
                which the output will be written.  The default is the
```

emacs@DDAHL-T410

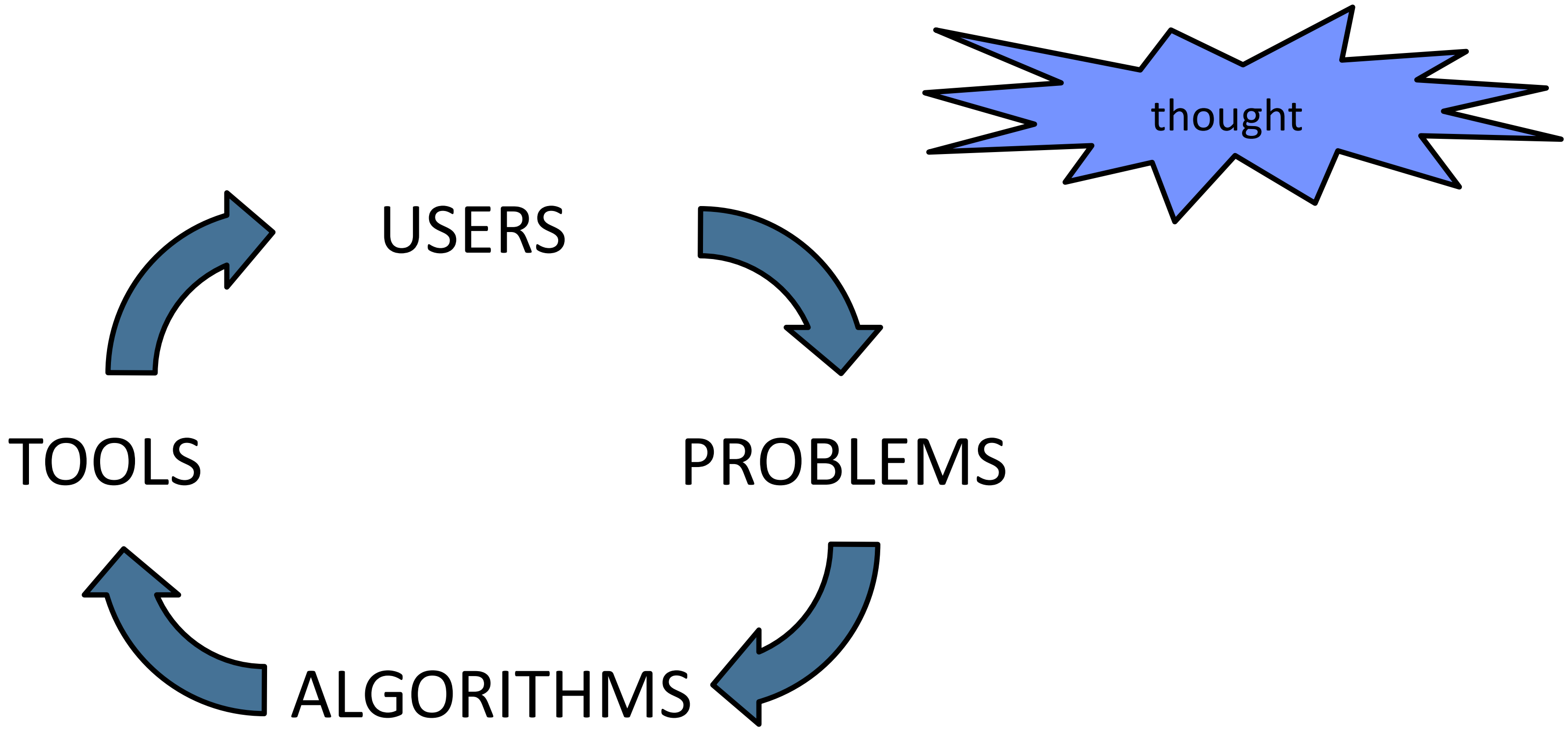
File Edit Options Buffers Tools Sh-Script Help

```
#!/bin/bash

sudoku=$1

if [[ -z $sudoku ]]
then
    echo "usage: driver.bash <sudoku-file>"
    exit 1
fi

echo "**************** Create QUBO for blank grid ****************"
s1.bash > s1.out

echo "**************** Identify fixed variables ****************"
s2.bash < $sudoku > s2.out

echo "**************** Compute reduced QUBO ****************"
s3.bash > s3.out

echo "**************** Identify free variables ; Enumerate variables **********
******"
s4.bash < s3.out > s4.out
--\---  driver.bash    Top L1     (Shell-script[bash])----------------------------
Indentation setup for shell type bash
```

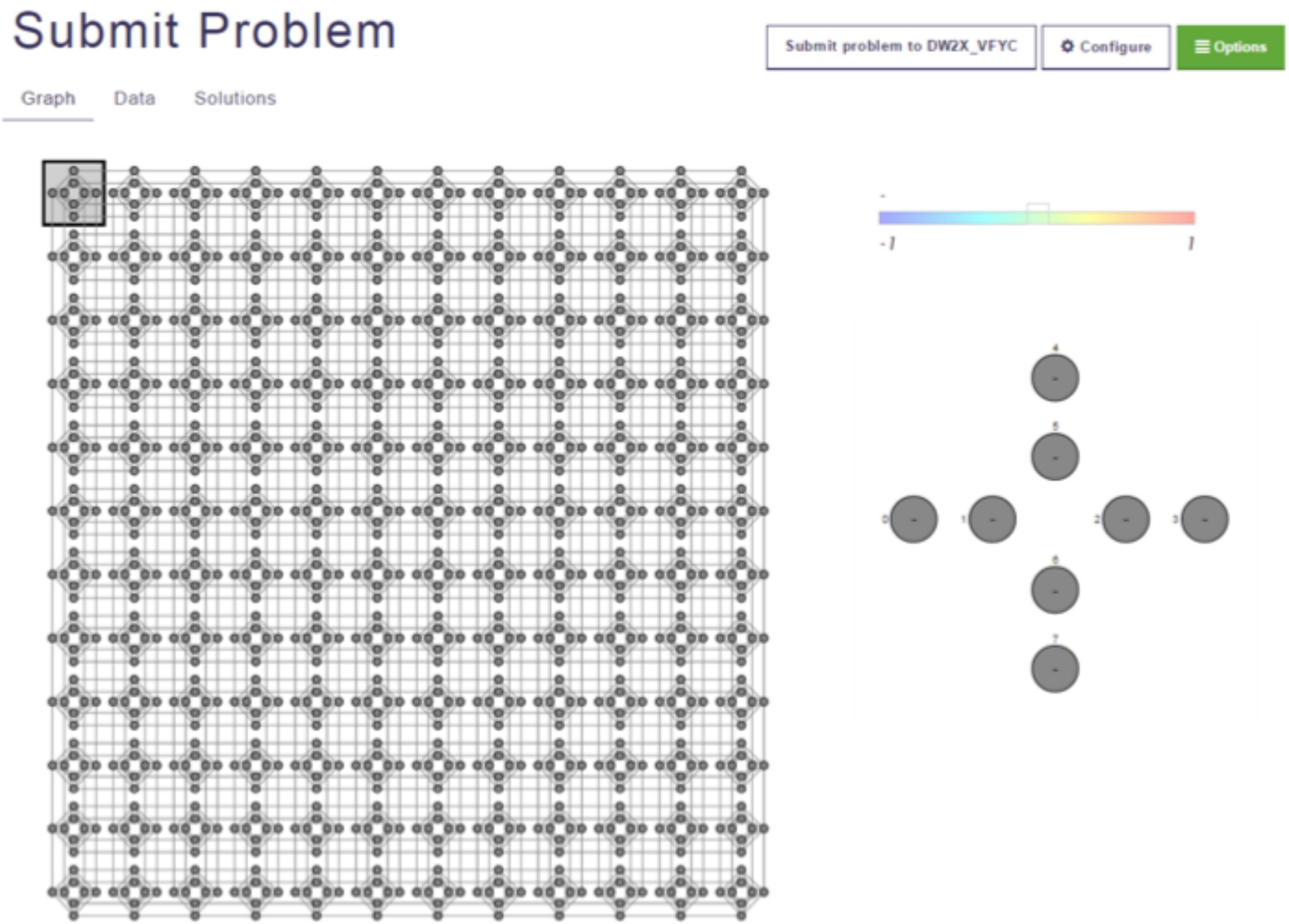
D:WAVE The Quantum Computing Company™

QUBITS

Day $\pi * 365$: The Virtuous Cycle



Day 1642: Virtual Full Yield



D-Wave The Quantum Computing Company™

QUBITS

Any Day, All Day

Problem	Technique
Hadamard matrices	Direct embedding
Ramsey lower bounds & more	BlackBox (Qsage)
Travelling Salesman Problem	BlackBox, QUBO, Parallel Update
Quadratic Assignment Problem	BlackBox, QUBO
Cyclic Ordering	Blackbox / Sorting network
Graph Isomorphism	Blackbox / Sorting network
Map Coloring	Various
Hello World	SAPI
Sudoku	qbsolv
Factoring	qbsolv

Perspectives about what might work best

STONEBRAKER ALGORITHM

Code section 2.

```
Until (it works) {
        Come up with a new idea;
        Prototype it with the help of superb computer scientists;
        Persevere, fixing whatever problems come up; always remembering that it is never too late to throw
        everything away;
        }
```

Likely copyright violation
DOI:10.1145/2869958

The D-Wave Trinity
Optimization & Constraint Satisfaction Problems
Machine Learning
Sampling

Day 1721: Today