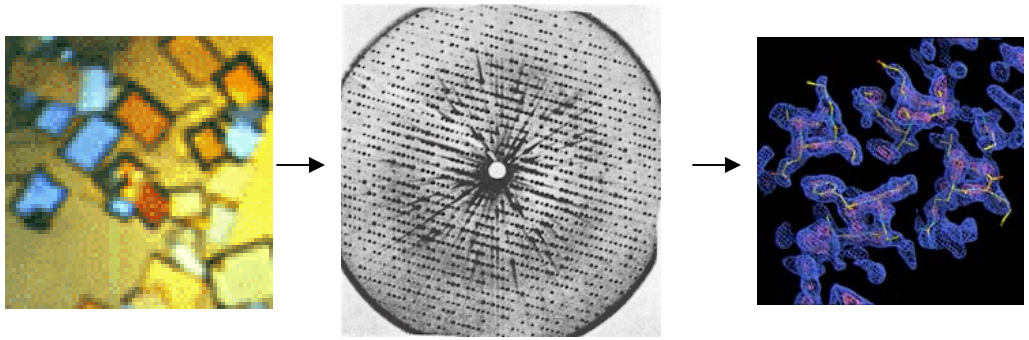
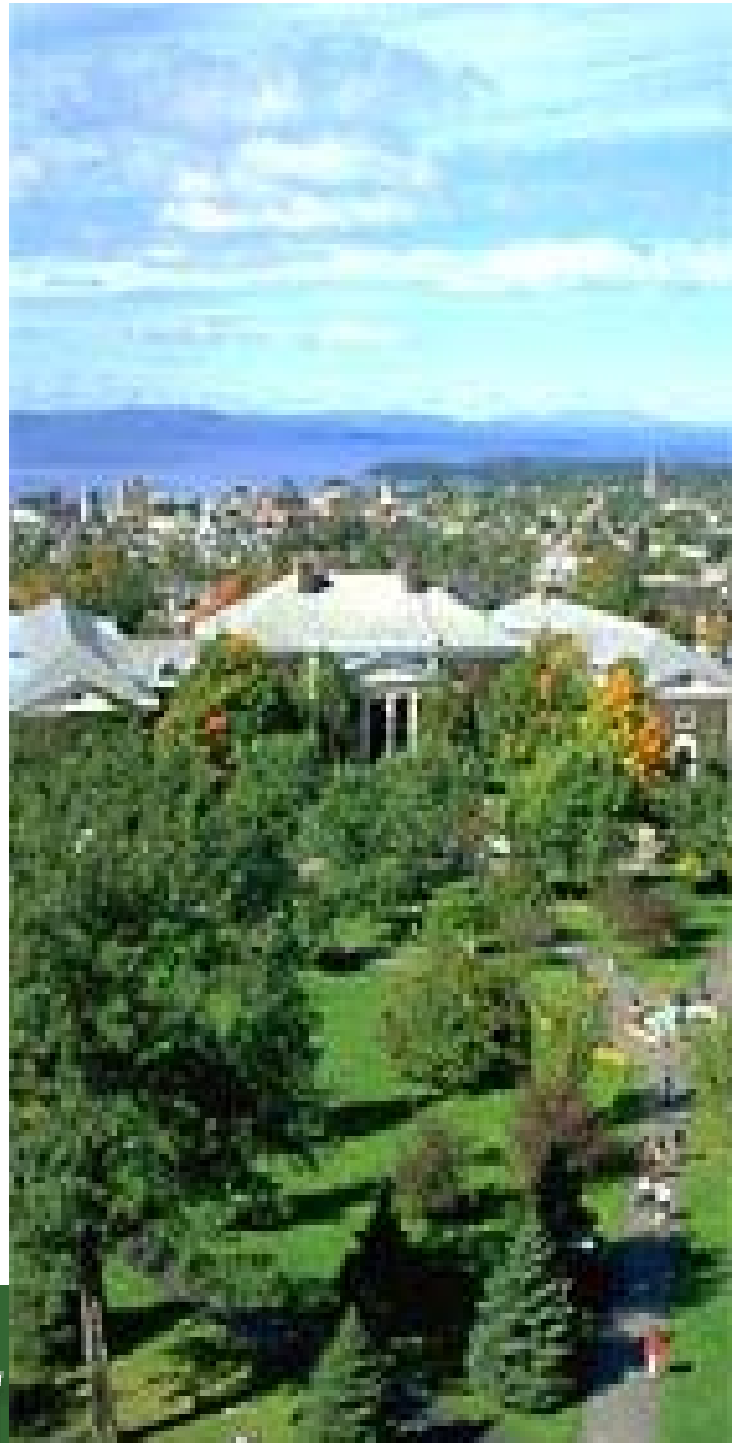
Crystallographic Case Study in an Interdisciplinary Evolutionary Computation Course



Margaret J. Eppstein
Department of Computer Science
James P. Hoffmann
Department of Botany



THE UNIVERSITY *of* VERMONT

Desired EC Course Outcomes:

1. Instill an understanding of Evolutionary Computation (EC) basics (theory and implementation)
2. Develop ability to use EC to solve real world problems
3. Stimulate interest/ability to understand current EC literature
4. Stimulate interest/ability to conduct research in EC
5. Develop excitement/appreciation for inter-disciplinary research
6. Gain experience working on inter-disciplinary, multi-level teams
7. Improve technical communication skills (written/oral)

Desired Research Outcomes:

See if EC is a feasible approach for locating heavy atoms in protein crystals (for crystallographic phasing via isomorphic replacement)

a. Try various EC approaches to see which, if any, is most promising

b. Solve scaled down version of problem

c. Perform scaling studies

d. Separate issues of efficacy of EC search strategy from complications such as noise in the data

Course Demographics:

Major	*#*
Computer Science	7
Mechanical Engineering	2
Civil&Environmental Engineering	1
Biomedical Engineering	1
Biological Sciences	1
Natural Resources	1
Microbiology & Molecular Genetics	1

Female	*Male*
4	10

Seniors	*MS student*	*PhD student*
5	5	4

# prior CS courses	*# students*
0	4
1	2
2 or more	8

"Prerequisites are computer programming (either through coursework or experience) and probability theory. A basic understanding of genetics & evolution is desirable, but not required. Multi-disciplinary teams will match students with complementary backgrounds."

Course Organization:

•Overview of EC (GA, ES, GP, theory) •Combination of lectures and hands-on laboratories •Targeted programming and non-programming assignments (theory and practice of EC) Culminate with comprehensive mid-term examination	1st 7.5 Weeks (Breadth)
•Case Study of Real-World Crystallography Problem •Read and discuss papers from the EC literature Culminate with mini-symposium and GECCO style papers	2nd 7.5 weeks (Depth)

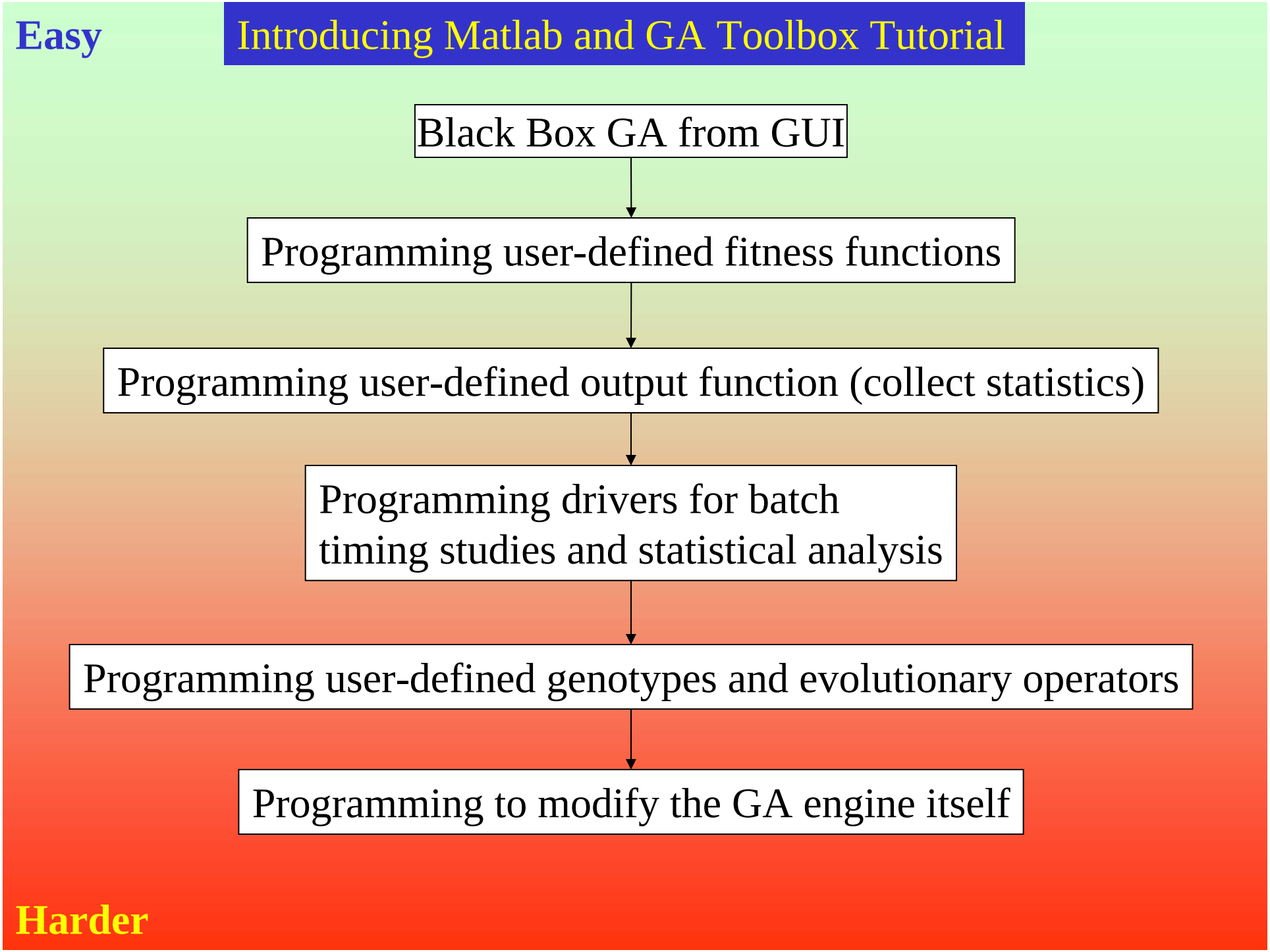
First 7.5 weeks: Overview of EC

Text Support	*Topics*	*Chapters*
A. E. Eiben and J.E. Smith, "Introduction to Evolutionary Computing", Springer, NY, (2003).	**Genetic algorithms** **Evolution strategies** **Genetic programming** **theory**	1-4,6,11
D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley, MA, (1989).	**Schema theory**	2
F. Rothlauf, "Representations for Genetic and Evolutionary Algorithms", Physica-Verlag, Heidelberg, (2002).	**Representation theory**	2,3,5

Software Support
Matlab v. 7 and GADS Toolbox
Schwefel's Evolution Strategies Code
GPLAB (public domain Matlab code)

Easy

Introducing Matlab and GA Toolbox Tutorial



Harder

Second 7.5 weeks: Current Literature

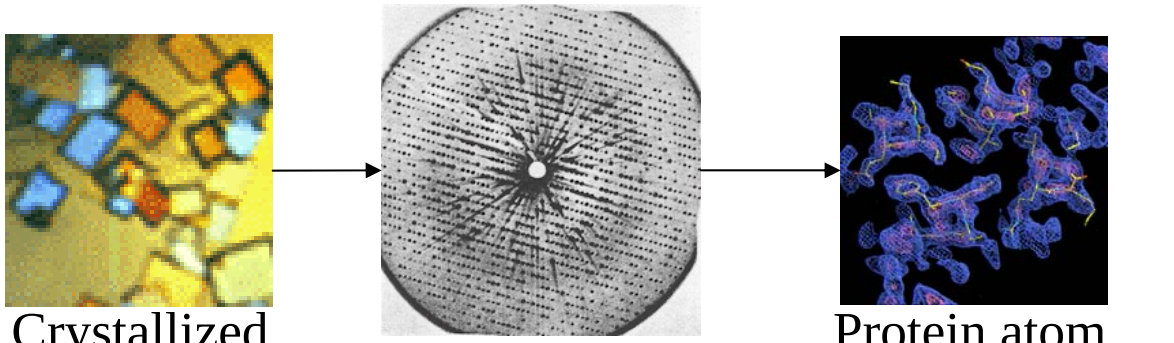
- Graduate students assigned to "lead" various papers
- Leaders prepared discussion questions (DQ) 1 week in advance of discussion
- All students required to hand in hardcopy of DQ answers before discussion commenced
- Papers chosen for breadth and potential relevance to case study
  - Memetic: case study
  - Memetic: Baldwinian vs. Lamarkian evolution
  - Self-adaptation of mutation rates
  - Mixing: P_c vs. Selection Intensity
  - Fitness sharing and niching for multimodal landscape
  - Island model for multi-modal and dynamic landscape
  - Fast messy GA
  - Gene expression messy GA
  - Probabilistic model building
  - Evolution of complexity (Alife)
  - Dominance & diploidy for dynamic Fitness
  - Co-evolution in a spatial EA

Second 7.5 weeks: Case Study

Criteria for Selecting Case Study:

1. Biologically-related real-world application
2. Problem generator for testing/validation
3. Alternate representations possible
4. Reasonably fast fitness function
5. Difficult, multi-modal search space
6. Solvable, at least for small problems
7. Controllable problem size for scaling studies

Criterion 1: Biologically-related real-world application

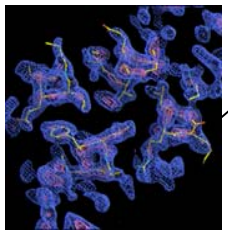


Determination of protein structure via x-ray crystallography.

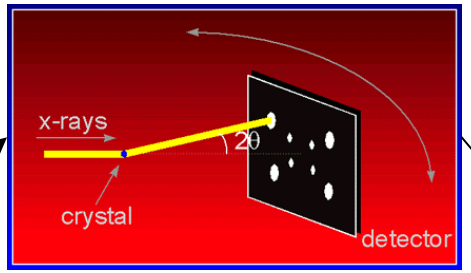
Instructor provided

- Background handout,
- Background lecture,
- Application-specific code.

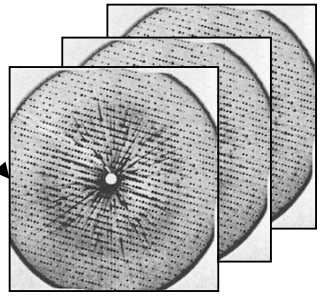
Crystallography is an Inverse Problem



“real space”
<x,y,z>



Physical equivalent of
Fourier transform



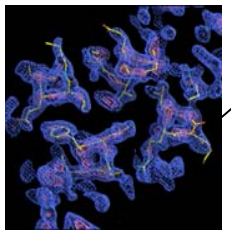
“reciprocal space”
<h,k,l>

Phases and
amplitudes

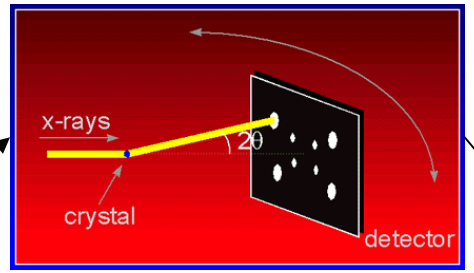
Fourier^{-1} transform

Can’t measure phases

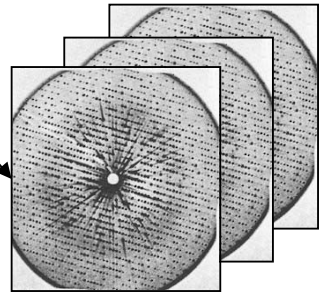
Unfortunately, it is an ***ill-posed*** Inverse Problem



"real space"
<x,y,z>



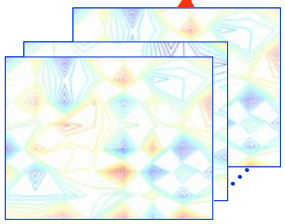
Physical equivalent of
Fourier transform



"reciprocal space"
<h,k,l>

Amplitudes only
(Phases set to 0)

Fourier^{-1} transform



"Patterson space"
<u,v,w>

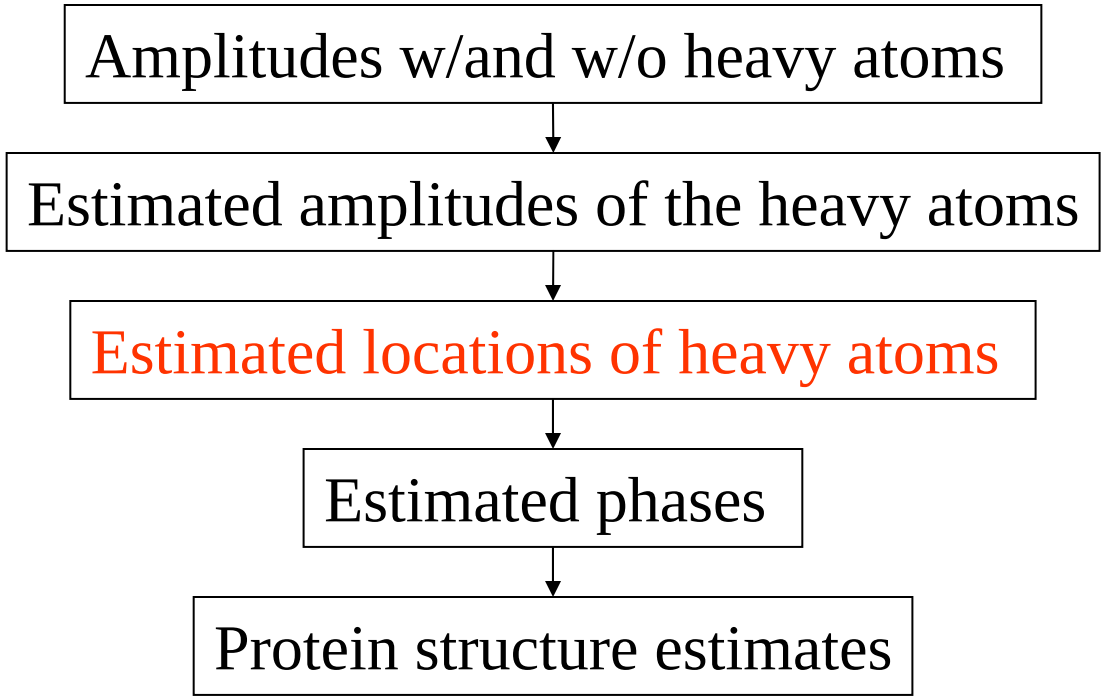
Can't measure phases

"Peaks" (high values) at locations <u,v,w> in the Patterson map correspond to *relative* distances between atoms in real space.

Crystallographic phasing by Isomorphous replacement

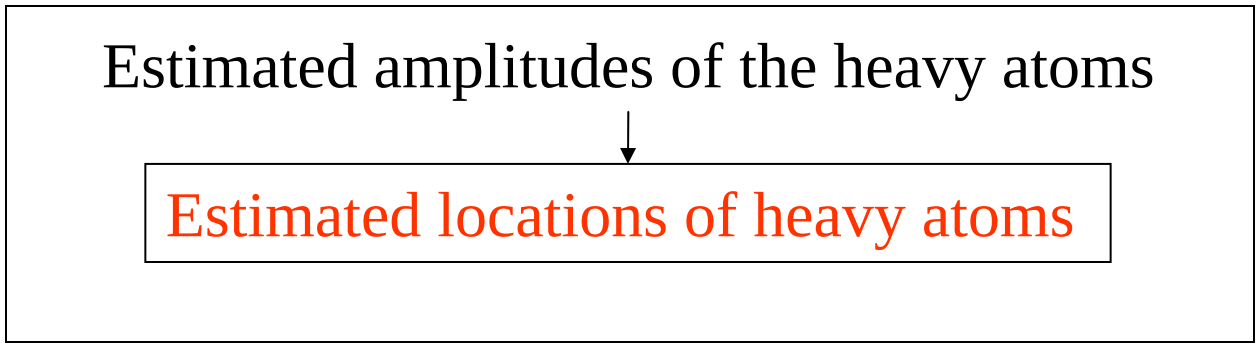
•Protein crystals doped with heavy atoms
(e.g., mercury or platinum)

•Heavy atoms perturb amplitudes of x-ray reflections



Case Study: locate heavy atoms

Key step in isomorphic replacement:

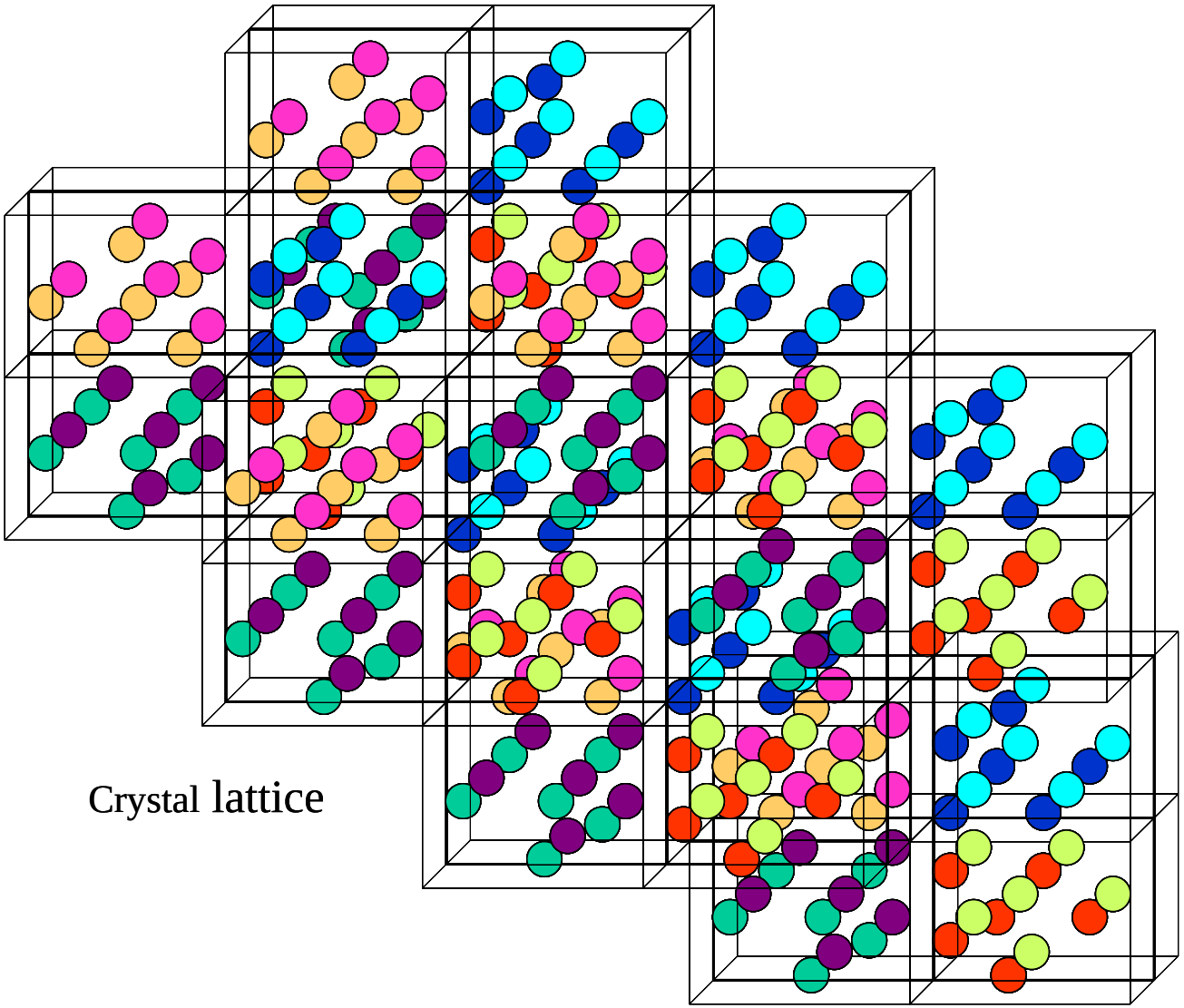


Resulting Optimization Problem:

The immediate computational problem is thus reduced to locating, in 3D space, the locations of a handful of heavy atoms, from a bunch of approximated heavy atom reflections.

This is still an ill-posed multi-modal inverse problem.

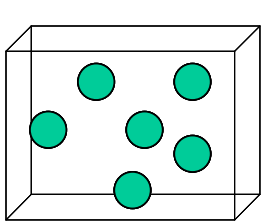
A crystal lattice is composed of many repeating identical unit cells.



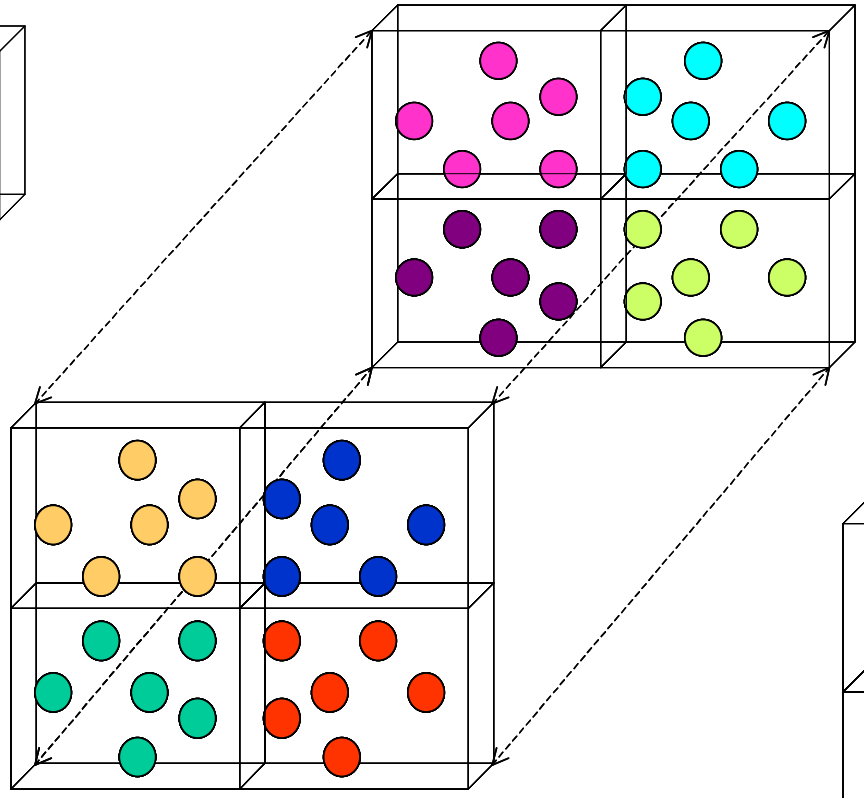
Crystal lattice

The Asymmetric Unit (AU) is the smallest building block of a crystal.

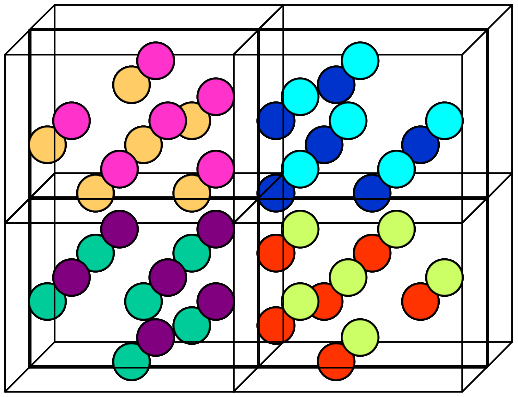
- Each atom in the AU is reflected about various axes of symmetry
- Symmetric AUs are grouped together to form a unit cell.



Asymmetric Unit (AU)



An example of 8-fold symmetry



Unit Cell

Crystals fall into different “space groups” based on their symmetry.

In this project we modeled the 4 2 2 space group

x_j	y_j	z_j
x	y	z
-x	-y	z
-x	y	-z
x	-y	-z
-y	-x	-z
y	x	-z
-y	x	z
y	-x	z

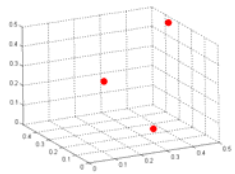
In space group 4 2 2, the AU is simply a cube with coordinates 0.0 to 0.5.

Each unique <x,y,z> location in the AU translates into eight <x,y,z> locations in the unit cell.

This results in $\binom{\#HA \cdot 8}{2}$ inter-atomic distances (peaks in Patterson space)

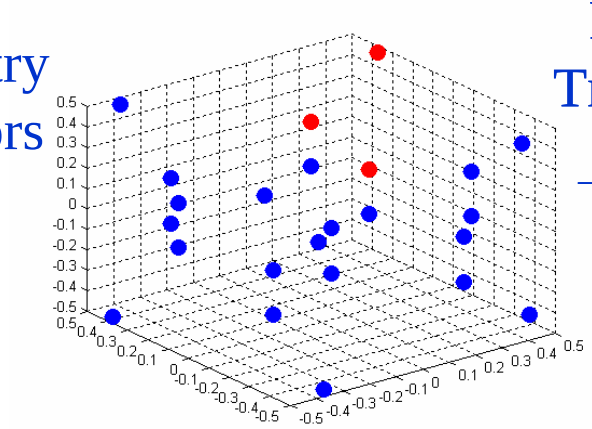
Criterion 2: Problem generator for testing/validation

Instructor developed/provided problem generator:



"True" heavy atom Locations in AU

Symmetry operators



Heavy atoms In unit cell

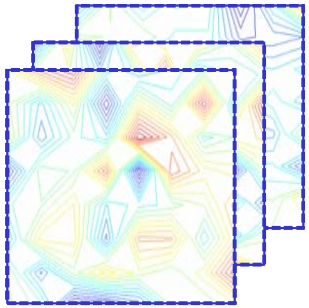
Fourier Transform

Phases and Amplitudes of 3D "Measured" Reflections

Throw away phases

Amplitudes of 3D Predicted Reflections (phases = 0)

Inverse Fourier Transform

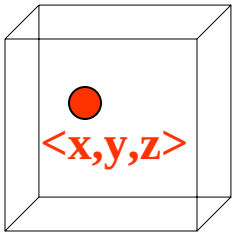


"True" Patterson Map

Criterion 3: Alternate representations possible

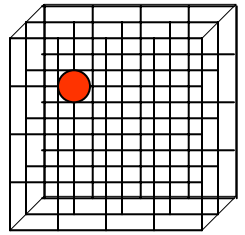
Various possible representations were suggested and debated in class discussion, including:

1. Real-valued <x,y,z> representation for each HA (fixed vs variable length chromosome)



2. Integer indeces of HA on a spatially discretized grid (fixed vs variable length chromosome)

3-D: <i,j,k>
1-D: <i>

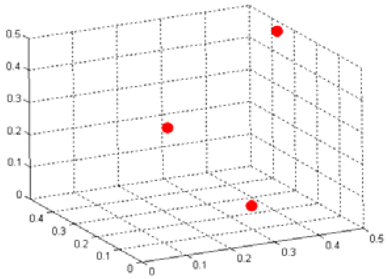


3. Binary representation for presence/absence of HA on grid **T/F**

Instructors led class to consensus for real-valued representation with fixed-length chromosome.

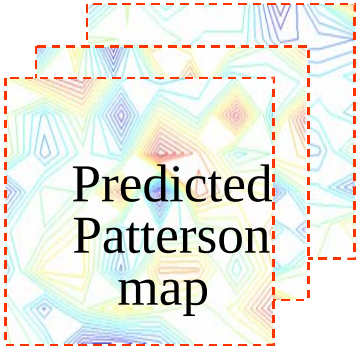
$$\mathbf{loc} = < x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_H, y_H, z_H >$$

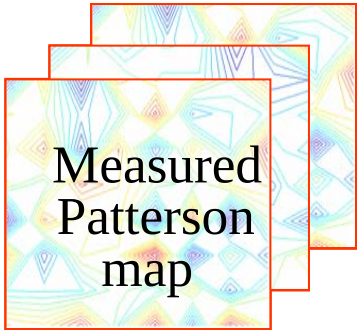
Criterion 4: Reasonably fast fitness function



Estimated <x,y,z> locations of HA

Predict Patterson Map





Compute correlation coefficient R between predicted and measured Patterson maps

R is “fitness”
$-1 \leq R \leq +1$
(+1 means perfectly correlated;
Must negate if minimizing.)

Criterion 4: Reasonably fast fitness function

Instructor developed/provided vectorized code for fitness evaluation:

$\mathbf{loc} = < x_1, y_1, z_1, x_2, y_2, z_2, ..., x_H, y_H, z_H >$ (HA locations in AU)

1) apply symmetry operators

2) $F_{hkl} = \sum_j f_j \exp\left(2\pi i\left(hx_j + ky_j + lz_j\right)\right)\exp\left(-\frac{B_j}{2}\sqrt{\frac{h^2}{a^2}+\frac{k^2}{b^2}+\frac{l^2}{c^2}}\right)$ (Discrete FT)

3) $\left|F_{000}^2\right| = 0.0$ (strip phases)

4) $P(u,v,w) = \frac{1}{V}\sum_h\sum_k\sum_l \left|F_{hkl}^2\right| \exp\left(-2\pi i\left(hu+kv+lw\right)\right)$ (Fast FT)

5) truncate all values more than 2 σ from mean

6) divide by RMS

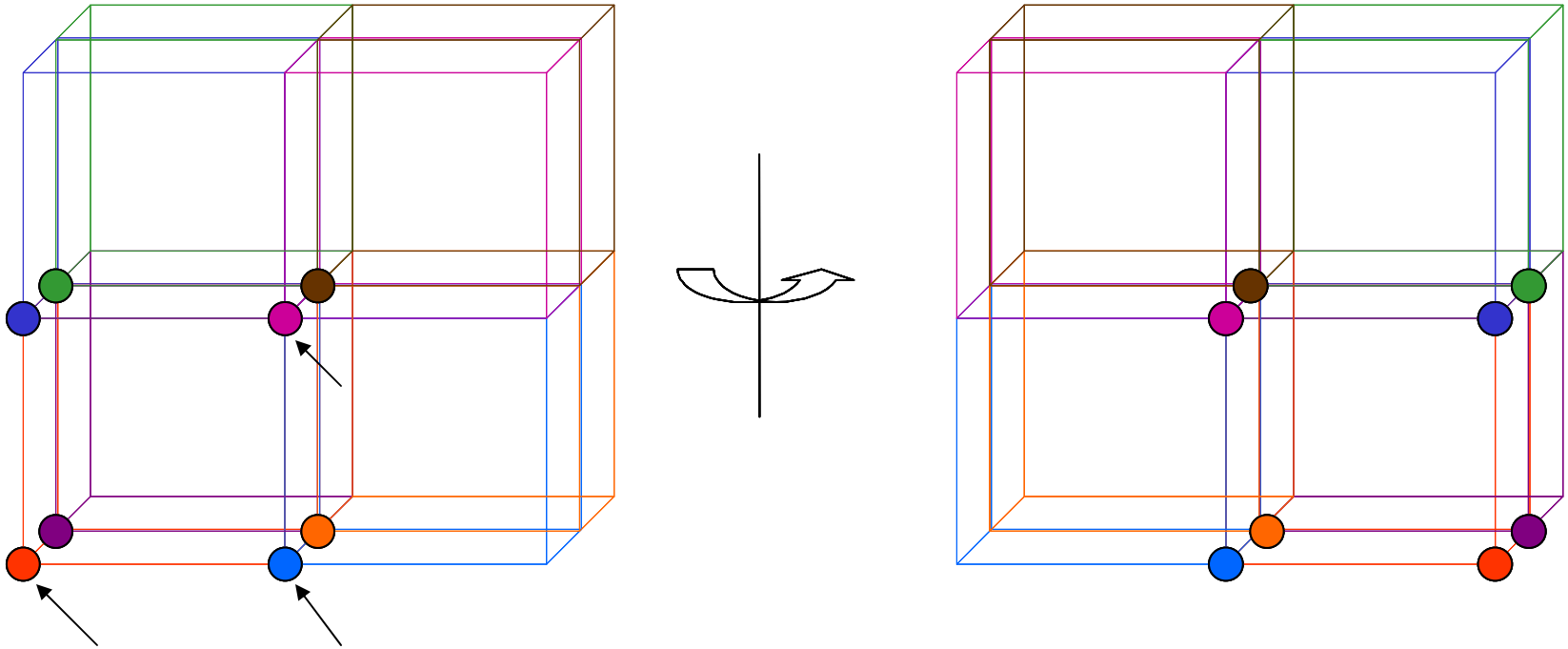
7) R=correlation(estimated Patterson, measured Patterson)

8) minimize -R

small crystals (20 angstrom cubes), few reflections (512),

~0.015 seconds on 2.2 GHz Pentium IV per fitness evaluation

Criterion 5: Difficult, multi-modal search space



The origin of the AU can be defined in different places relative to the crystal lattice; Solutions from different origins yield identical Patterson maps.

Mirror image solutions (different “hands”) have the same inter-atomic distances, therefore same Patterson map.

- **In space group 4 2 2 there are 8 distinct hand-origin possibilities (8 global optima).**
- **Combining partial solutions from different hand-origin will yield low fitness.**
- **No way to a priori identify which hand-origin a given HA location is in.**

Criterion 6: Solvable, at least for small problems

Several simplifications were introduced to ensure problems were solvable:

1. Noise free synthetic data (no protein background)

2. Small crystals ensured 2.5 angstrom sampling resolution (despite small # reflections); blur factor of 10. Instructors tested fitness function for perturbed HA locations; hill-climbing possible within about 4 angstroms.

3. Small number of HA

4. Number of HA assumed known.

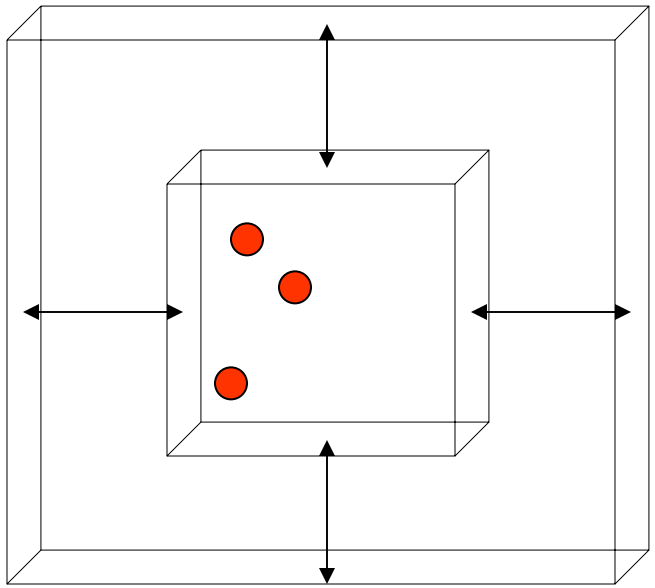
Enabled students to focus on effective EC search strategies for the problem.

Criterion 7: Controllable problem size for scaling studies

Teams were required to perform 2 scaling studies:

1. Vary # HA from 1 to at least 3 (I.e., 3 to 9 unknown reals); some teams searched for up to 10 HA (30 unknowns)

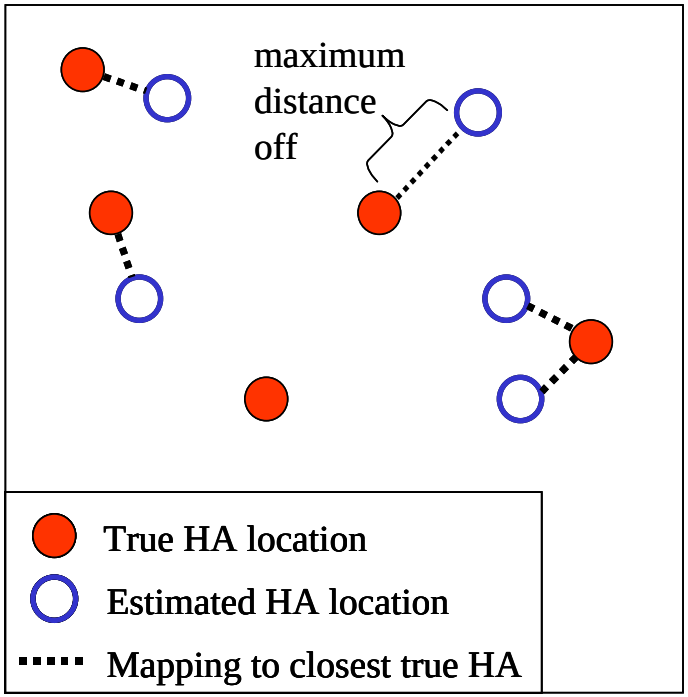
2. Vary range of search space from [0.15, 0.35] to [0.0, 0.5]



"True" HA locations restricted to inner range

Metrics for validating solution quality

Instructor developed/provided code for solution quality metrics:



- We computed the pair-wise Euclidean distances between each true atom and each estimated atom, for each of the 8 possible hands and origins,
- mapped each estimated HA to its closest true HA in each hand-origin,
- and selected the hand and origin of the true solution for which the sum of the mapped distances was the least.

1. **Number of HA found in best hand-origin**
2. **Maximum distance off in best hand-origin**

Individual design of ideas for search strategies

Prior to team formation, each student was directed to

•Read relevant sections of text regarding advanced EC design

Text Support	***Topics***	***Chapters***
A. E. Eiben and J.E. Smith, "Introduction to Evolutionary Computing", Springer, NY, (2003).	Adaptive parameter control Hybrid/memetic algorithms Meta-population structure	4,8,9,10

•Each student individually proposed possible evolutionary search strategies suitable for the problem.

Team formation

Four teams were formed, comprising 3-4 students each, balancing:

- Academic ability (based on homework and exam grades from the first half of the semester)
- Graduate and undergraduate standing
- CS and non-CS majors
- CS&E and Life-Science majors
- Personality/Leadership skills (based on class interactions from the first half of the semester).

Team/Individual progress and accountability incentives

1. Project grades: 50% team deliverables, 50% individual contributions/performance
2. Weekly target deadlines for project milestones
3. Weekly in-class team meetings/oral reports to “clients” (instructors)
4. Blue-books of individual progress, signed weekly
5. 1-page terminal summary of individual contributions, co-signed by all team members
6. Individual confidential terminal evaluation of contributions of team members
7. Confidential presentation evaluation forms of oral presentations of other teams

EC approaches selected by the 4 teams:

After team formation

teams discussed/debated member ideas and arrived at consensus for approach

EC approaches selected:

1. ISLAND: meta-population structure
2. AM: adaptive ES-like mutation
3. GAGA: 2-stage GA
4. GAPS: hybrid GA-Pattern Search

In addition to scaling studies, teams were required to test the efficacy of some aspect specific to their approach.

Parameter tuning, experimental design, and statistical analysis determined by each team.

EC approaches selected by the 4 teams:

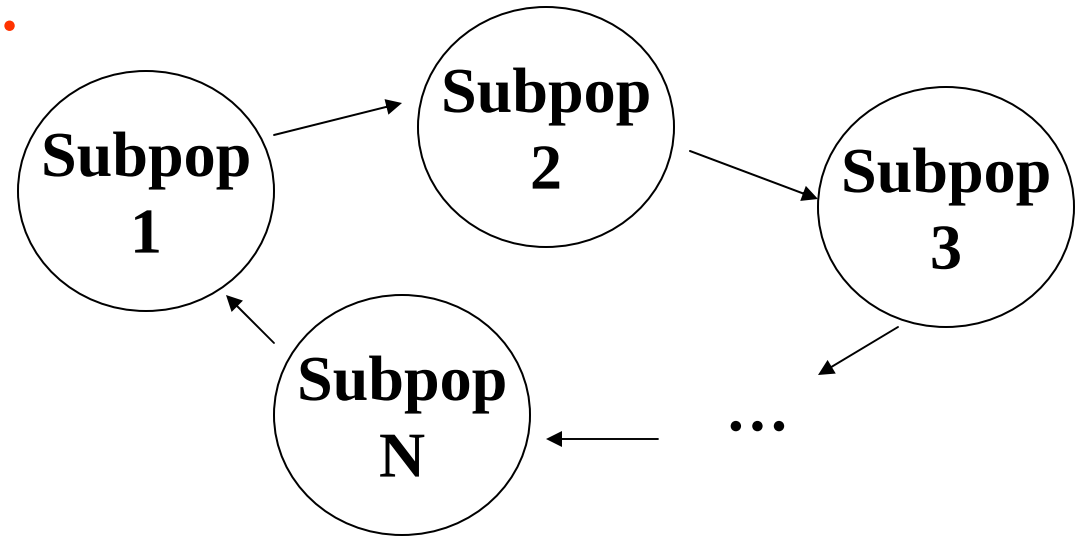
Team 1: ISLAND (Meta-population approach)

•Island model with uni-directional migration between island populations, connected in a ring topology.

•Single-point crossover

•Bounded Gaussian mutation with a deterministic reduction in step size with iteration, where out-of-range mutations were mapped back to the boundaries of the feasible reason.

Team 1 tested the effects of both number of islands and migration rate.



EC approaches selected by the 4 teams:

Team 2: AM (adaptive mutation)

•Implemented an adaptive mutation operator, as used in evolution strategies

•One uncorrelated mutation step size per decision variable.

•Unlike the other teams, they used uniform crossover.

Team 2 compared adaptive and non-adaptive mutation.

$$< x_1, y_1, z_1, ..., x_H, y_H, z_H \mid \underbrace{\sigma_1, ..., \sigma_H}_{\text{Adaptive mutation Step sizes}} >$$

$$\sigma_i \leftarrow \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)}$$

$$x_i \leftarrow x_i + \sigma_i \cdot N_i(0,1)$$

$$\tau' = \frac{0.5}{\sqrt{2H}}; \quad \tau = \frac{0.5}{\sqrt{2\sqrt{H}}}$$

EC approaches selected by the 4 teams:

Team 3: GAGA (2-stage Genetic algorithm)

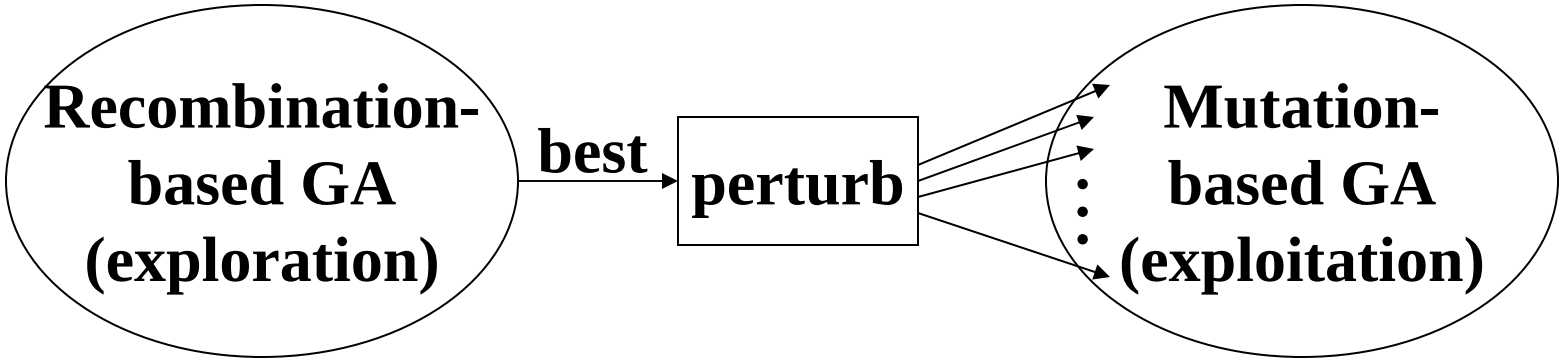
•Stage 1: recombination-based GA for exploration.

•Best solution from Stage 1 was repeatedly perturbed randomly to create the initial population for the Stage-2.

•Stage 2:mutation-based GA, for exploitation.

•Crossover and mutation operators were similar to those of Team 1.

Team 3 investigated the effect of the relative ratio of the population sizes in the two stages.



EC approaches selected by the 4 teams:

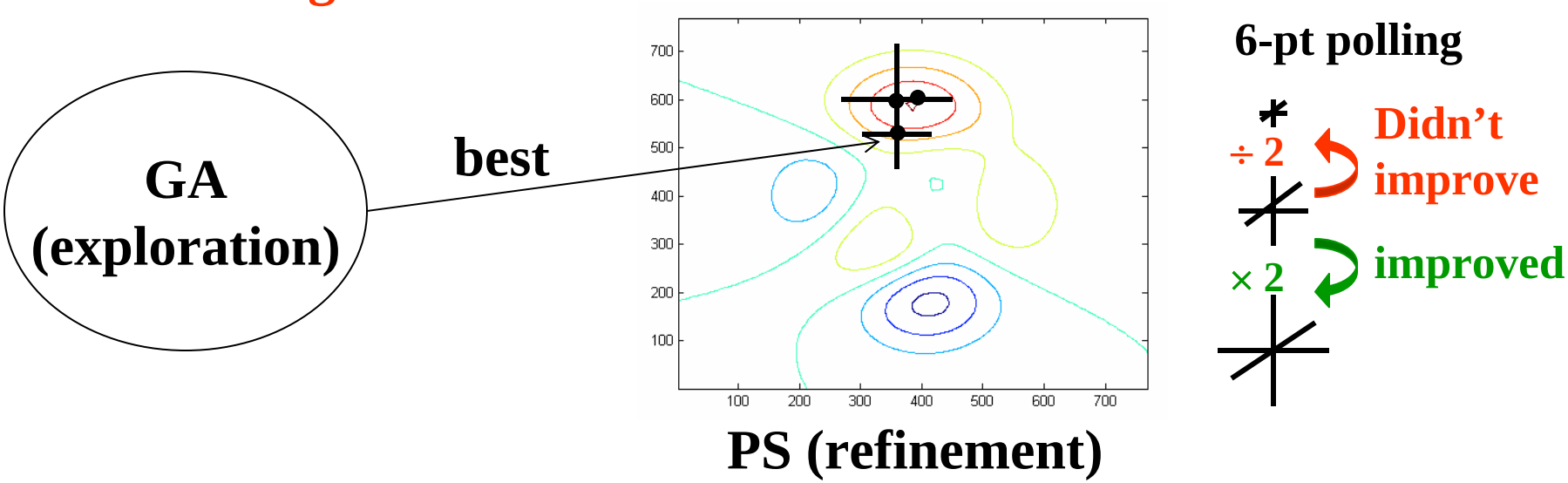
Team 4: GAPS (hybrid GA with Pattern Search)

•Stage 1: GA for stochastic exploration

•Stage 2: Pattern Search (PS) on single best GA solution, for deterministic refinement of the solution

•Crossover and mutation were like those of Team 1, with the exception that out-of-range mutations were randomly reset to feasible values.

Team 4 compared the results from a 1-stage GA to those of the 2-stage GAPS.



Team Deliverables:

1. Documented working code and results for
   - GA with and w/o custom features
   - Scaling study for #HA
   - Scaling study for size of search space in AU
2. GECCO-style manuscript
   - Detailed technical writing guidelines provided by instructors
   - GECCO website for formatting
3. 30-minute oral presentation
   - Detailed guidelines for slide prep and oral delivery provided by instructors
   - Mini-symposium format (instructors as moderators)

Results of Team Projects:

All teams successfully completed all deliverables. ☺

Since experimental design (pop size, mutation rates, termination criteria, number of test problems, number of replicates, etc.) and statistical analysis were left up to the teams, results were not directly comparable based on team deliverables. ☹

Comparative Results (10 problems, 10 reps each):

method	*#HA found (out of 3)*	*% trials finding all 3 HA*	*Patterson Correlation R*	*Spatial resolution (angstroms)*	*CPU time per trial (min)*
GAPS	**2.95**	**95**	**0.998**	**0.334**	**28**
GAGA	2.90	90	0.996	0.528	42
ISLAND	2.73	73	0.978	1.198	4
AM	2.34	34	0.865	4.342	23

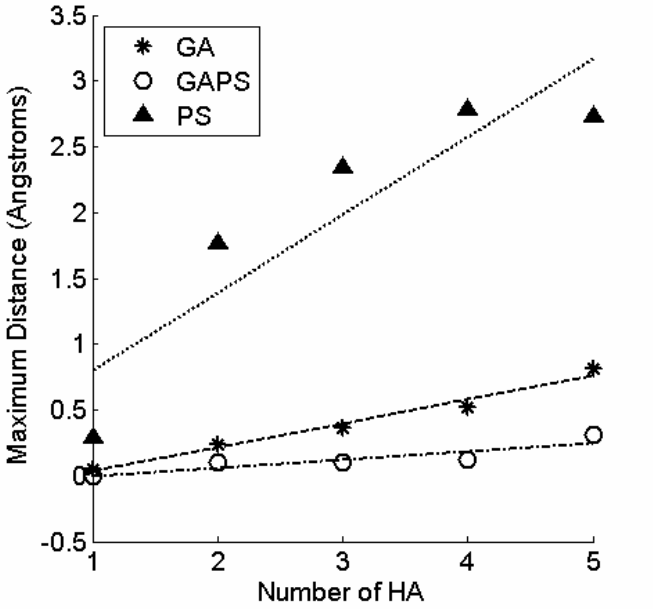
After the semester, we ran batch tests to directly compare the 4 algorithms using the same experimental design (with Josh Payne's help).

Conclusions regarding use of EC for finding HA:

1. Using adaptive mutation operator did not prove effective.
2. Using Island model with smaller subpopulations led to premature convergence in the subpopulations; larger total population would be required to see if islands can help search.
3. Using back-to-back hybrid Genetic Algorithm for global exploration followed by hill-climber for local refinement appears most promising.
4. Deterministic pattern search is a more effective hill-climber than a mutation-based GA.

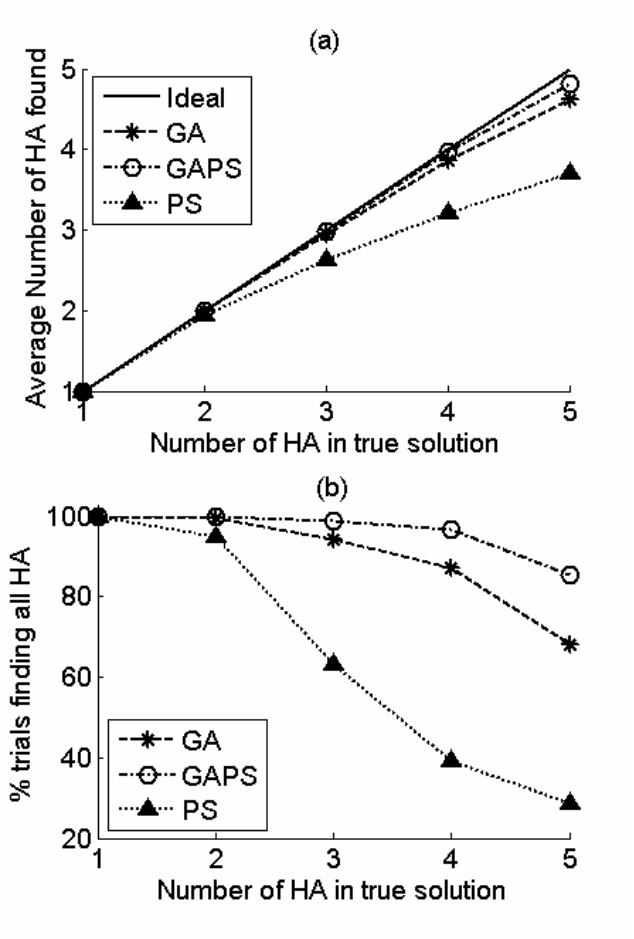
Follow-up Study with best approach (GAPS):

J.L. Payne and M.J. Eppstein, "A Hybrid Genetic Algorithm with Pattern Search for finding Heavy Atoms in Protein Crystals", accepted for Biological Applications Track of GECCO, 2005. (nominated for a Best Paper Award ☺)



$$popsize \sim 100 \cdot HA^{1.68}$$

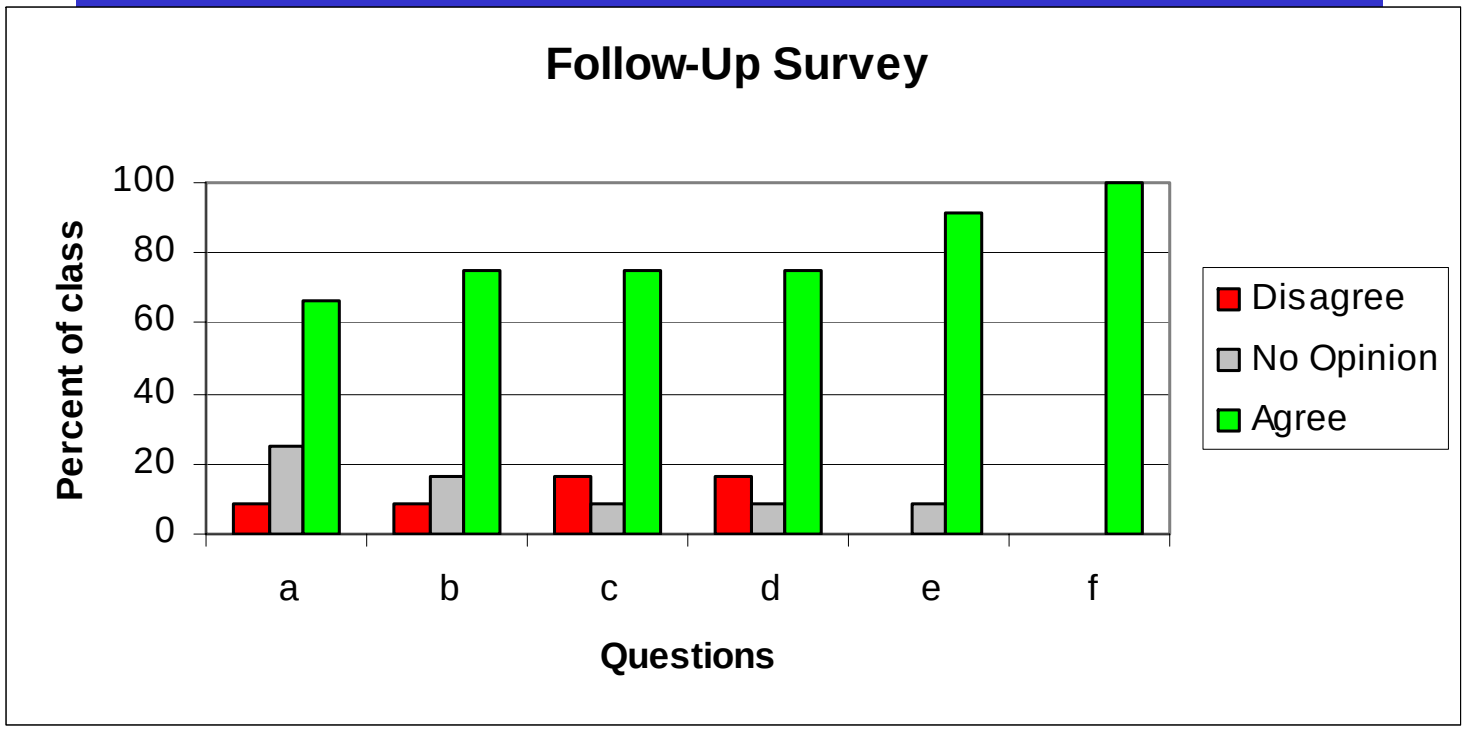
$$gens \sim 110 \cdot HA + 1.7$$



Conclusions regarding use of GAPS for finding HA:

1. Using a back-to-back hybrid Genetic Algorithm/Pattern search may be viable approach to finding heavy atoms.
2. Relaxing assumption that # HA is known could be tackled by:
    a. Estimating a subset of HA to fix hand-origin, then adding in and optimizing additional HAs until fitness decreases.
    b. Using variable-length chromosomes
    c. Estimating different replications with different numbers of HA and choosing the best.
3. Using real noisy data will require more sophisticated fitness function (e.g. maximum likelihood approach); separate issue.
4. Scaling up to more realistic problems (more heavy atoms, larger crystals, more reflections) is straightforward, but will increase computational burden (time less of an issue, but memory may be limiting).
5. Computational burden could be reduced by seeding population with likely candidates based on self-peaks.

Course Outcomes Assessed by Follow-up Survey



a. Course increased my interest in taking interdisciplinary coursework.

b. Course increased my interest in pursuing interdisciplinary research.

c. Course increased my interest in pursuing research in Evolutionary Computation.

d. My enthusiasm for project was enhanced by real-world nature of problem.

e. Using the Matlab GA toolbox facilitated focus on EC design.

f. Using the Matlab programming language allowed sufficient low-level customization of EC applications.

Summary of Key Pedagogy for incorporation of real-world Case Study in Introductory Evolutionary Computation course

1. **Course structure:** **breadth (lecture format/targeted assignments)→ depth (case-study, current literature)**
2. **Matlab:** **high-level ←→ low-level**
3. **Assignments:** **tutorial→targeted →research**
4. **Real-world problem:** **↑motivation**
5. **Team competition:** **↑motivation**
6. **Interdisciplinary teams:** **complementary backgrounds/strengths**
7. **Limited scope:** **doable w/in half semester (on laboratory PCs)**
8. **Scaling studies:** **projecting extendibility**
9. **GA-Toolbox and Instructor-provided modules:** **students focus on EC design**
10. **Team benchmarks:** **milestones, oral reports, deliverables**
11. **Individual accountability:** **initial ideas, logs, presentation, summary statements, team member evaluations**
12. **Detailed specifications/guidelines** **for deliverables**
13. **Conference style** **paper/presentation**

Acknowledgements

Development of this course was supported in part by

•An instructional incentive grant from the University of Vermont Center for Teaching and Learning, and

•A pilot award funded by DOE-FG02-00ER45828 awarded by the US Department of Energy through its EPSCoR Program.

We would like the thank all the students in the EC course for their enthusiastic participation:
N. Basha, C. Coughlin, E. Gaddis, M. Harpster, P. Hurd, C. Korecki, C. Mark, D. Pechenik, I. Spiro, J. Stinnett-Donnelly, D. Whitaker, C. Wolf, Y. Zhang, and especially J. Payne.

We would also like to acknowledge

•M.A. Rould, for lending his expertise in crystallography, and

•J. Gilbert, whose related research with Drs. Eppstein and Rould helped the instructors to formulate a doable problem for the course.