Unconstrained Optimization:
Direct Search Methods

AE 6310: Optimization for the Design of Engineered Systems

Spring 2017

Dr. Glenn Lightsey

Lecture Notes Developed By Dr. Brian German

Georgia Institute of Technology

Daniel Guggenheim School
of Aerospace Engineering

Direct Search Methods

Methods for which **derivatives are not used in the search.**

“We use the phrase ‘**direct search**’ to describe sequential examination of trial solutions involving comparison of each trial solution with the ‘best’ obtained up to that time together with a strategy for determining … what the next trial solution will be … search strategies which **employ no techniques of classical analysis** except where there is a demonstrable advantage in doing so.” (Hooke and Jeeves, 1961) [emphasis added]

Direct Search Methods

Advantages:

- Do not require gradient calculations
- Easy to implement/code, compared to line search methods
- Are sometimes more amenable to parallel computing

Disadvantages:

- Do not use the information available from gradient calculations
- May take more function calls to converge
- Not amenable to classical calculus-based analysis

Some Examples of Direct Search Algorithms

We will briefly discuss the following direct search methods:

- Grid search
- Random search
- Random walk
- Compass search
- Coordinate pattern search

There are *many* more! See (Kolda, 2003) for a nice review.

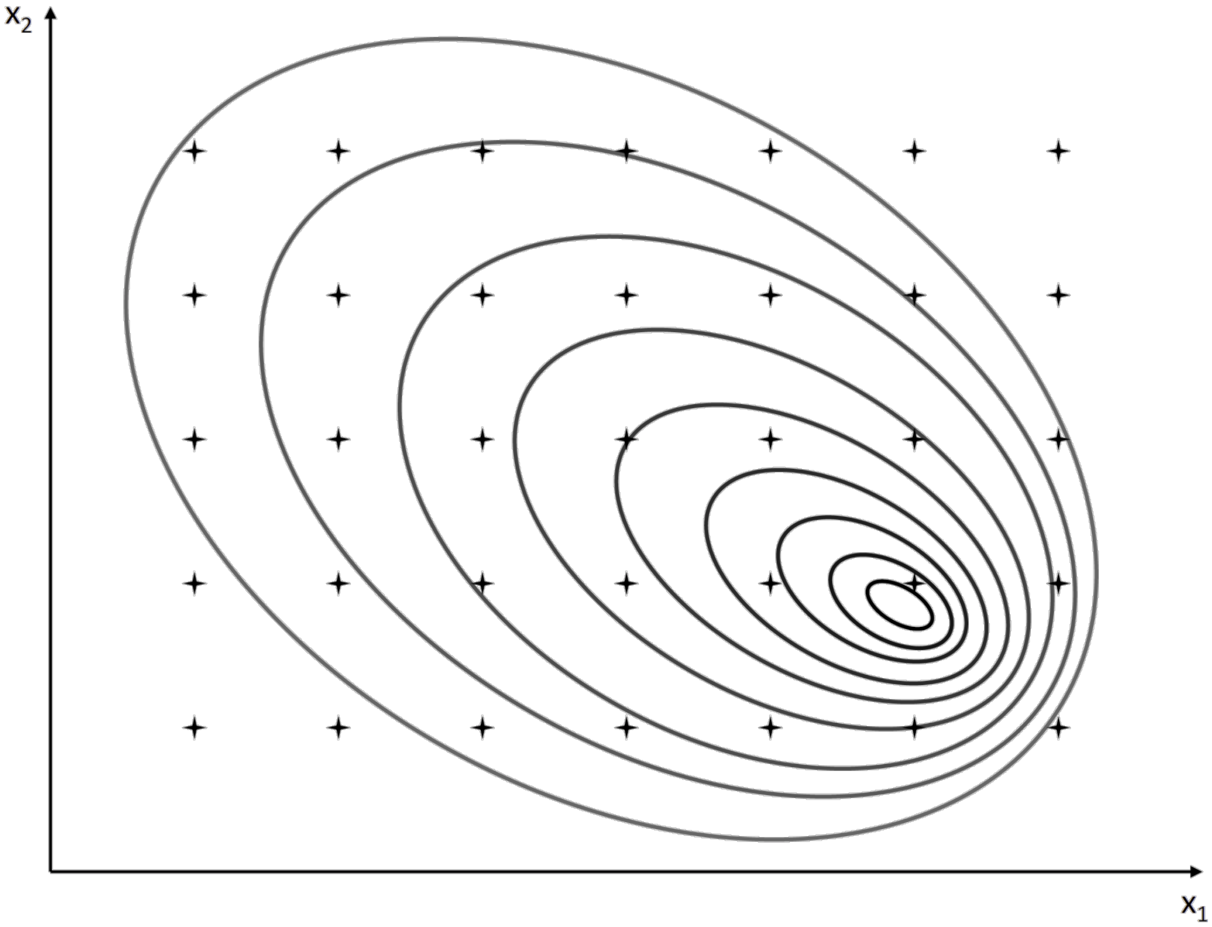
Grid Search

Searches a full-factorial discretization of the entire space.

General procedure:

1. Discretize the space as a full-factorial "grid" in all of the design variables.
2. Evaluate f at each point in the grid.
3. (Optional) Re-discretize the grid at a finer resolution in the region of the space with the smallest f from step 2.
4. (Optional) Evaluate f at each point in the finer grid.
5. (Optional) Repeat steps 3 through 4 until convergence.
6. Select the best point

Grid Search



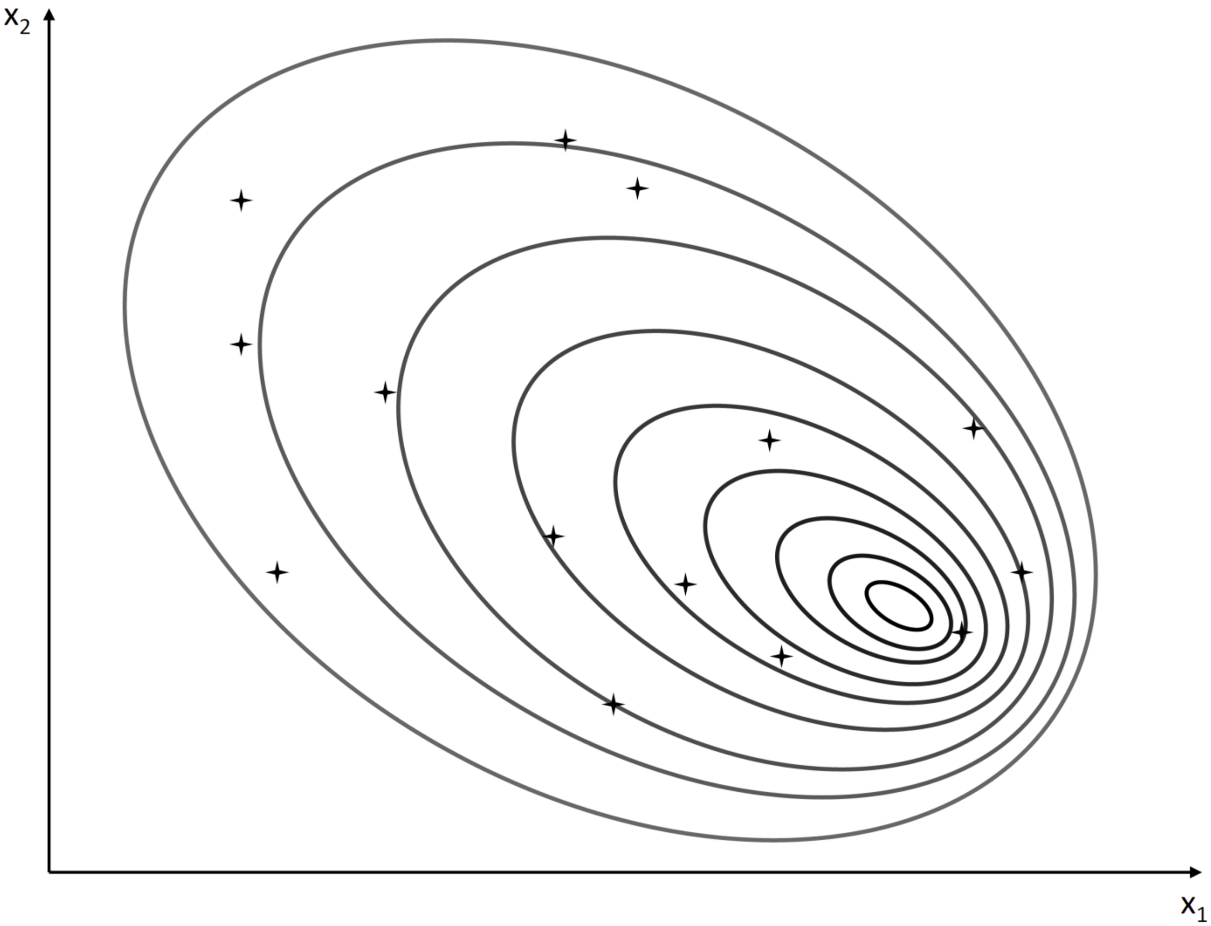
Random Search

Searches random points in a discretized design space.

General procedure:

1. Discretize the space (even just to floating point precision).
2. Generate a set of points by sampling each design variable from a probability distribution (typically a uniform distribution) to generate each point.
3. Evaluate f at the set of sampled points.
4. (Optional) Re-discretize the space at a finer resolution in the region with the smallest f from step 3.
5. (Optional) Evaluate f at each point in the finer discretization.
6. (Optional) Repeat steps 4 through 5 until convergence.

Random Search



Grid and Random Search Pros & Cons

Advantages:

- Easy to implement/code
- Reasonable for low dimensional problems
- Works with discrete design variables
- Good chance of finding a point near the global minimum
- "Embarrassingly parallel"

Disadvantages:

- Requires *many* function calls
- Especially inefficient for higher dimensional problems

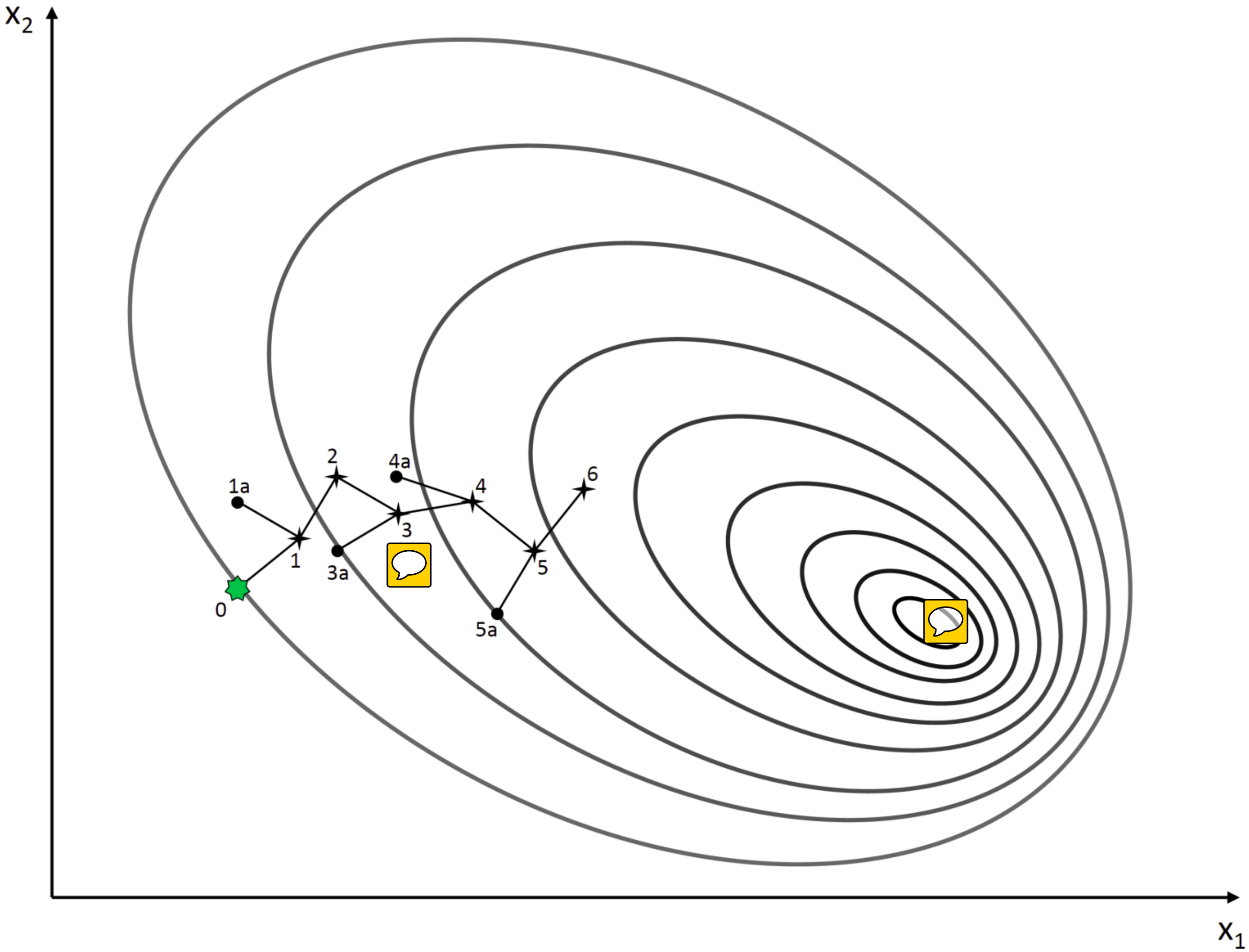
Random Walk

Starts from an initial point and searches in random directions with a fixed step size, stepping to the first point with improvement.

General procedure:

1. Pick initial point and evaluate f.
2. Take trial step in random direction.
3. Evaluate f at trial point.
4. If f does not improve, test another random direction.
5. Move to first trial point with improved objective function value.
6. If several directions have been tested without improvement, halve step size and repeat from step 2.

Random Walk



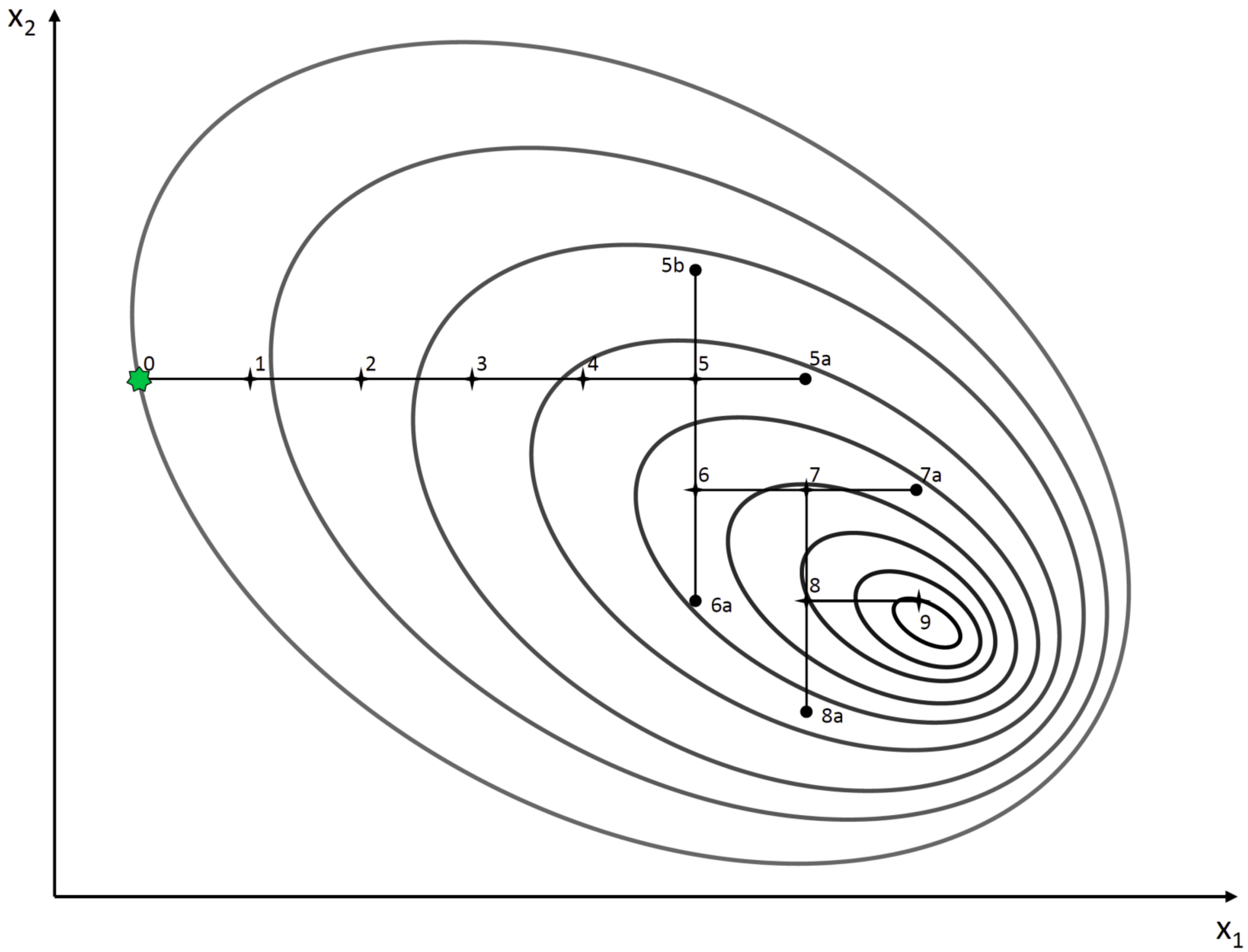
Compass Search

Starts from an initial point and searches in one univariate direction at a time with a fixed step size, stepping to the first point with improvement.

General procedure:

1. Pick initial point and evaluate f.
2. Take trial step in first univariate direction, e.g. x_1.
3. Evaluate f at trial point.
4. If f does not improve, test next univariate direction, e.g. x_2. If f does improve, retain search direction.
5. Move to first point with improved f value, step again in retained search direction, and repeat from step 3.
6. If all directions have been tested without improvement, halve step size and repeat from step 2.

Compass Search



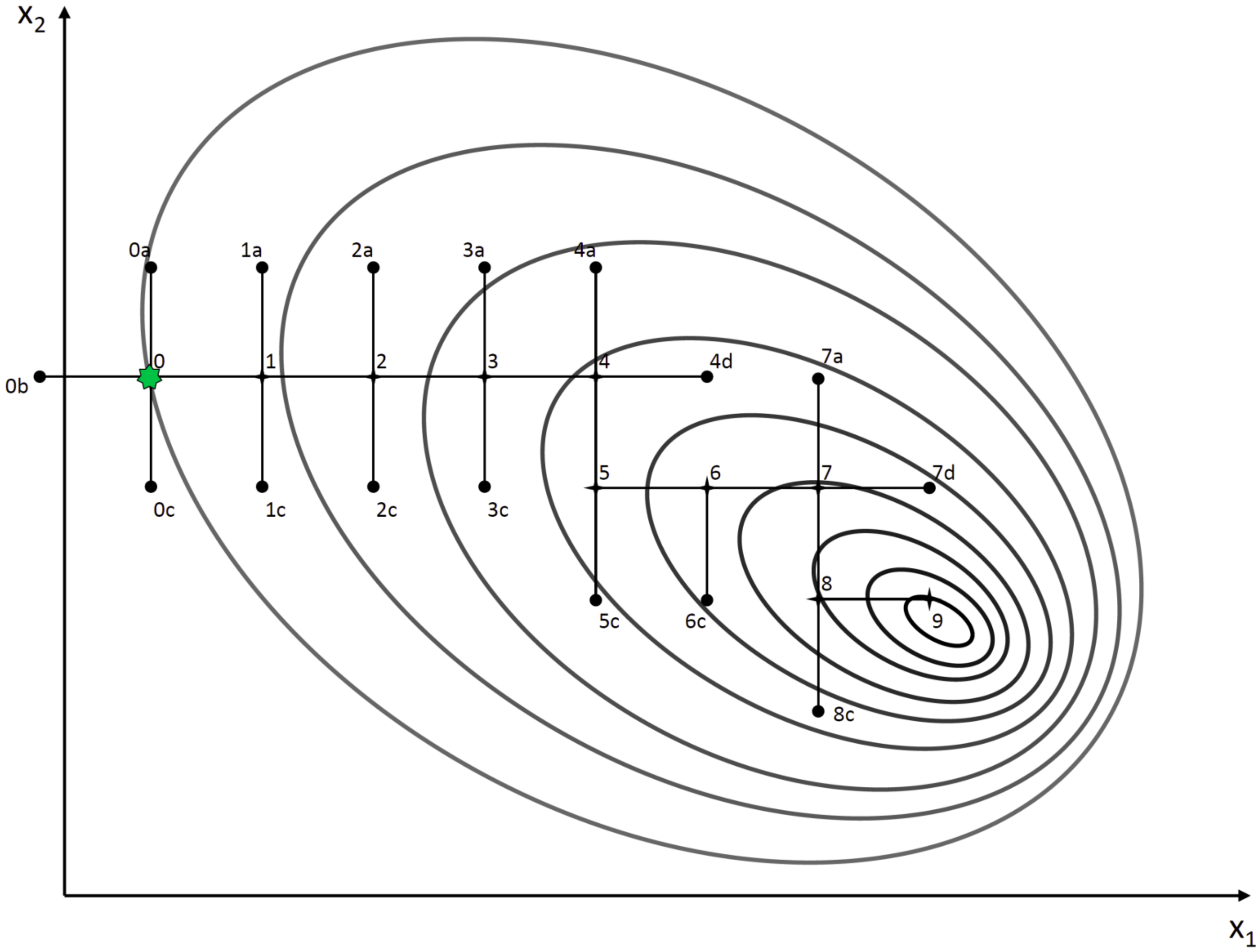
Coordinate Pattern Search

Starts from an initial point and searches in all univariate directions at the same time with a fixed step size, stepping to the point with most improvement.

General procedure:

1. Pick initial point and evaluate f.
2. Take trial step in all univariate directions, both (+) and (-), except for steps that would replicate points already evaluated.
3. Evaluate f at trial points.
4. Move to point with most improvement in f and repeat from step 2.
5. If all directions fail to improve, halve step size and repeat from 2.

Coordinate Pattern Search


x2
0a
1a
2a
3a
4a
0b
0
1
2
3
4
4d
7a
5
6
7
7d
0c
1c
2c
3c
8
9
5c
6c
8c
x1