

An Algorithm for Nonlinear Optimization: Nelder-Mead

Michael T. Stobb

University of California Merced, Merced, CA

November 25, 2013

Why Optimization?

Why Optimization?

- We wish to minimize the function

$$f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$$

Why Optimization?

- We wish to minimize the function

$$f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$$

- Other than continuity, no additional information about $f(x)$ is assumed

Why Optimization?

- We wish to minimize the function

$$f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$$

- Other than continuity, no additional information about $f(x)$ is assumed
- No constraints are assumed

Nelder-Mead in a Nutshell

Nelder-Mead in a Nutshell

- Take a polyhedra and evaluate the objective function at the vertices

Nelder-Mead in a Nutshell

- Take a polyhedra and evaluate the objective function at the vertices
- Construct a series of trial moves, replace the current worst point with the best of the new points

Nelder-Mead in a Nutshell

- Take a polyhedra and evaluate the objective function at the vertices
- Construct a series of trial moves, replace the current worst point with the best of the new points
- Repeat

Simplexes

Simplexes

- Simplexes are generalized triangles

Simplexes

- Simplexes are generalized triangles
- For n dimensions they have $n + 1$ vertices

Simplexes

- Simplexes are generalized triangles
- For n dimensions they have $n + 1$ vertices
- They are *not* necessarily regular

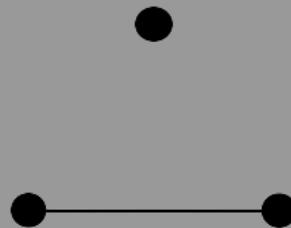
Simplexes

- Simplexes are generalized triangles
- For n dimensions they have $n + 1$ vertices
- They are *not* necessarily regular



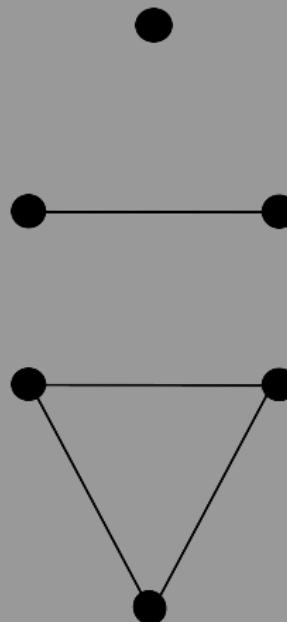
Simplexes

- Simplexes are generalized triangles
- For n dimensions they have $n + 1$ vertices
- They are *not* necessarily regular



Simplexes

- Simplexes are generalized triangles
- For n dimensions they have $n + 1$ vertices
- They are *not* necessarily regular



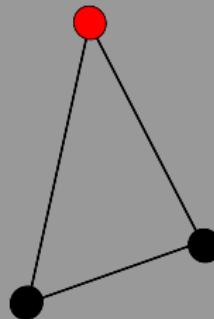
Simplex Transformations

Simplex Transformations

- Reflection

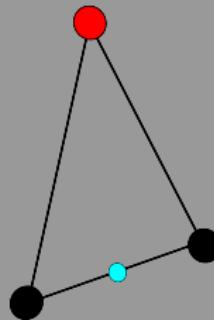
Simplex Transformations

- Reflection



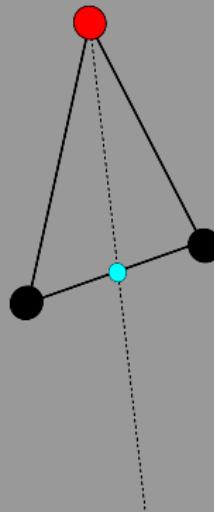
Simplex Transformations

- Reflection



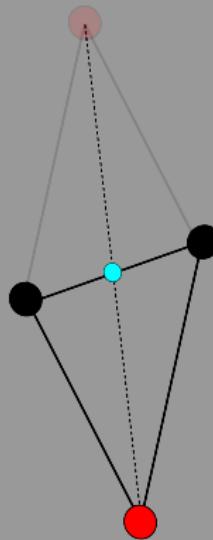
Simplex Transformations

- Reflection



Simplex Transformations

- Reflection

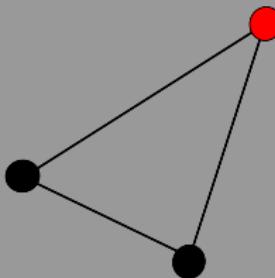


Simplex Transformations

- Reflection
- Expansion

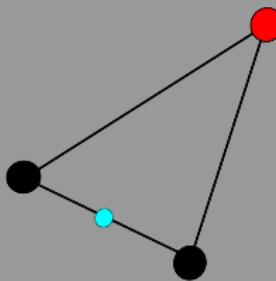
Simplex Transformations

- Reflection
- Expansion



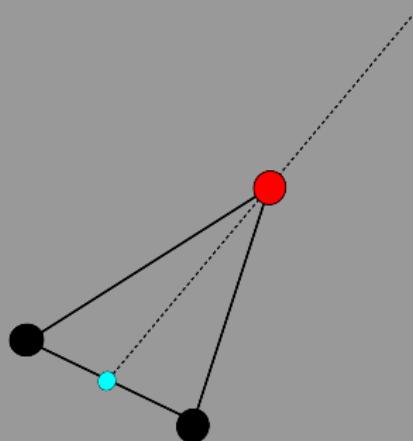
Simplex Transformations

- Reflection
- Expansion



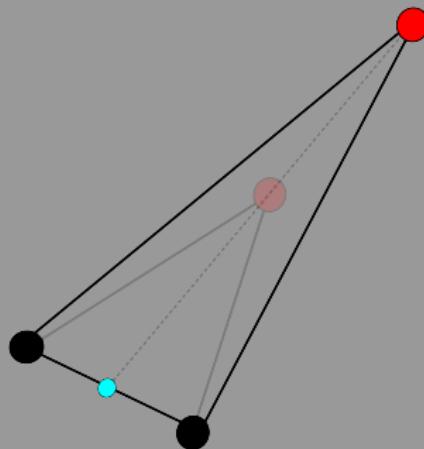
Simplex Transformations

- Reflection
- Expansion



Simplex Transformations

- Reflection
- Expansion

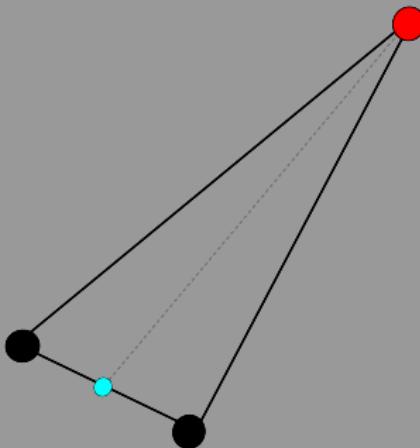


Simplex Transformations

- Reflection
- Expansion
- Contraction

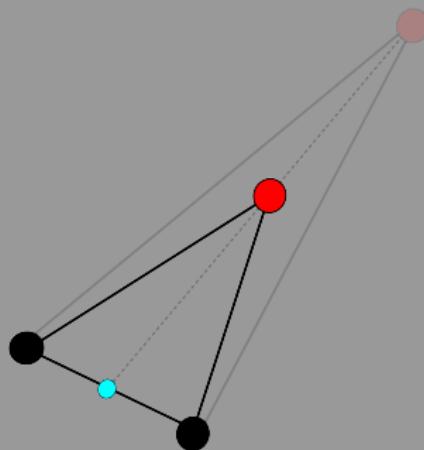
Simplex Transformations

- Reflection
- Expansion
- Contraction



Simplex Transformations

- Reflection
- Expansion
- Contraction

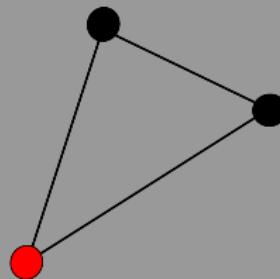


Simplex Transformations

- Reflection
- Expansion
- Contraction
- Shrink

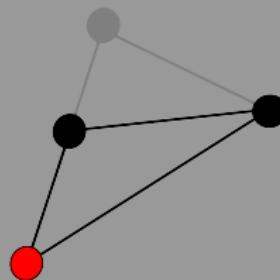
Simplex Transformations

- Reflection
- Expansion
- Contraction
- Shrink



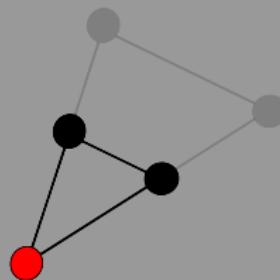
Simplex Transformations

- Reflection
- Expansion
- Contraction
- Shrink



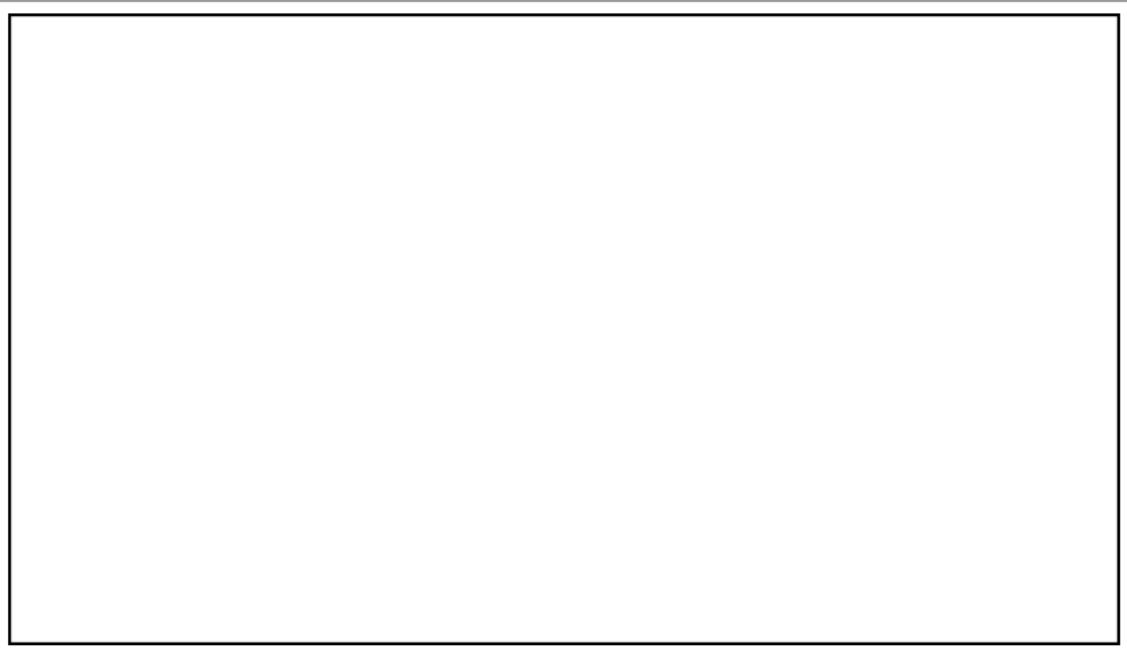
Simplex Transformations

- Reflection
- Expansion
- Contraction
- Shrink



Nelder-Mead Algorithm

Nelder-Mead Algorithm



Nelder-Mead Algorithm

Find worst point P_h

Reflect

Calculate new point P^*

Is P^* the best point?

Nelder-Mead Algorithm

Find worst point P_h
Reflect
Calculate new point P^*
Is P^* the best point?



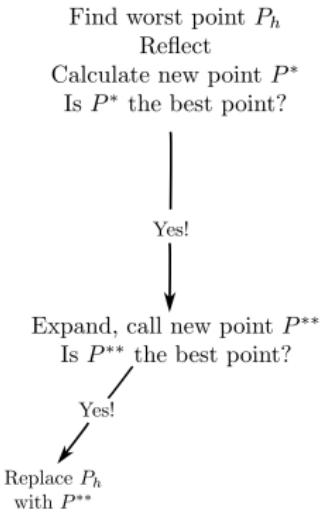
Nelder-Mead Algorithm

Find worst point P_h
Reflect
Calculate new point P^*
Is P^* the best point?

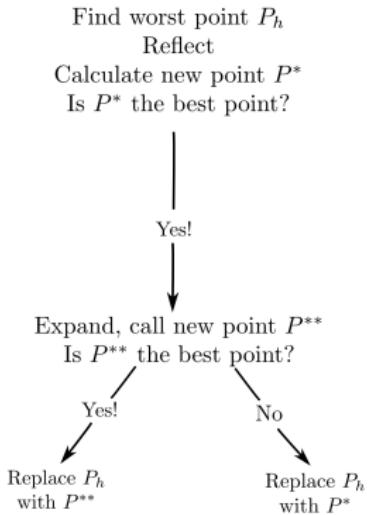


Expand, call new point P^{**}
Is P^{**} the best point?

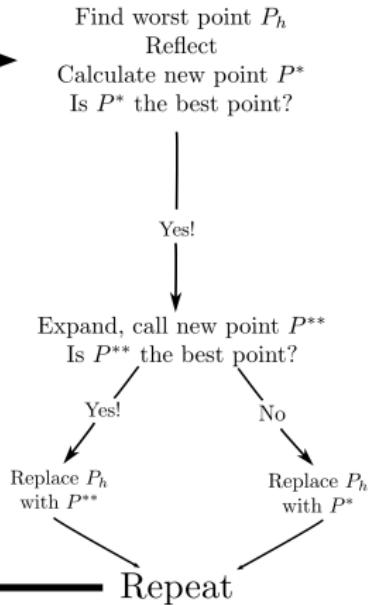
Nelder-Mead Algorithm



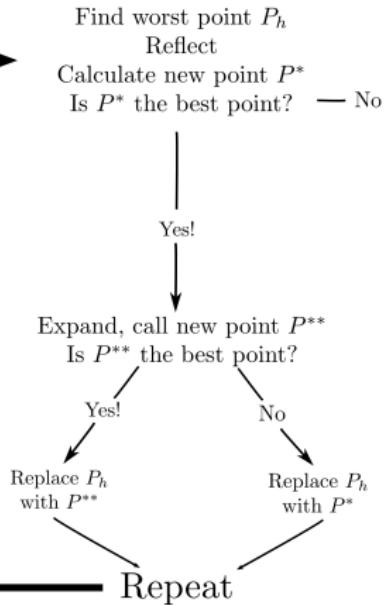
Nelder-Mead Algorithm



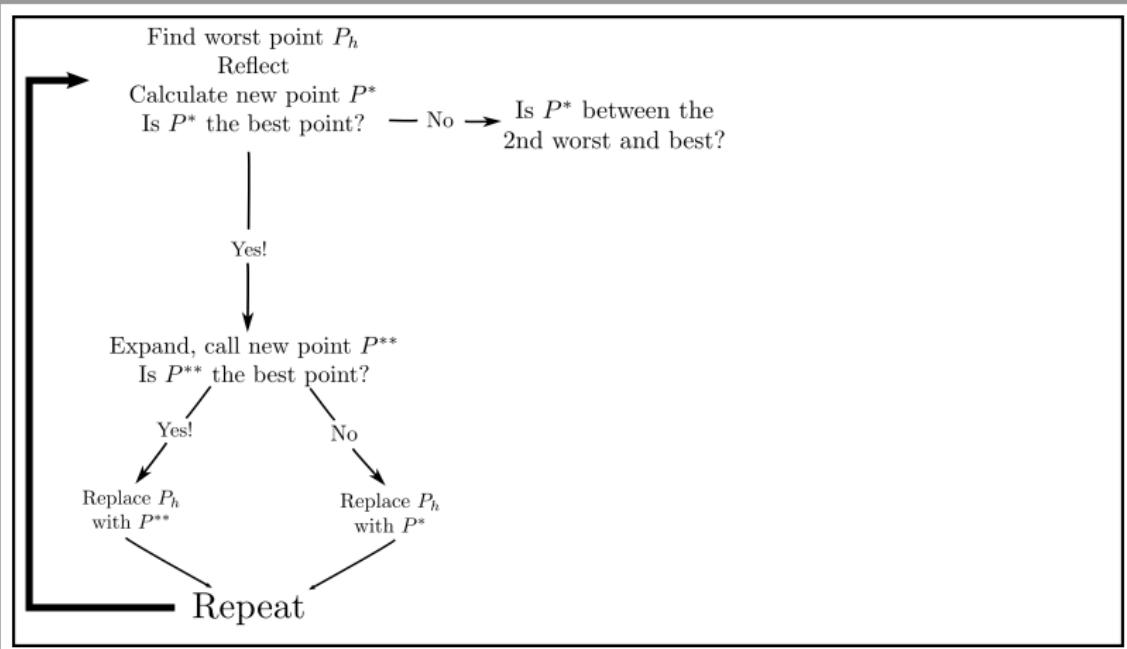
Nelder-Mead Algorithm



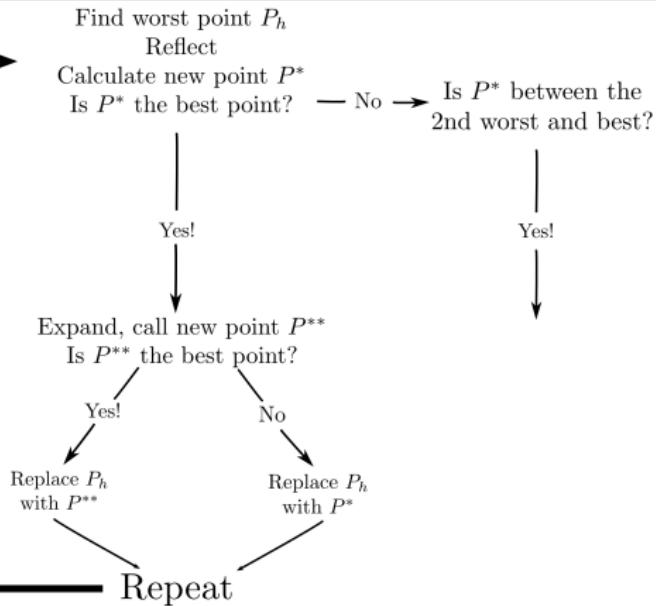
Nelder-Mead Algorithm



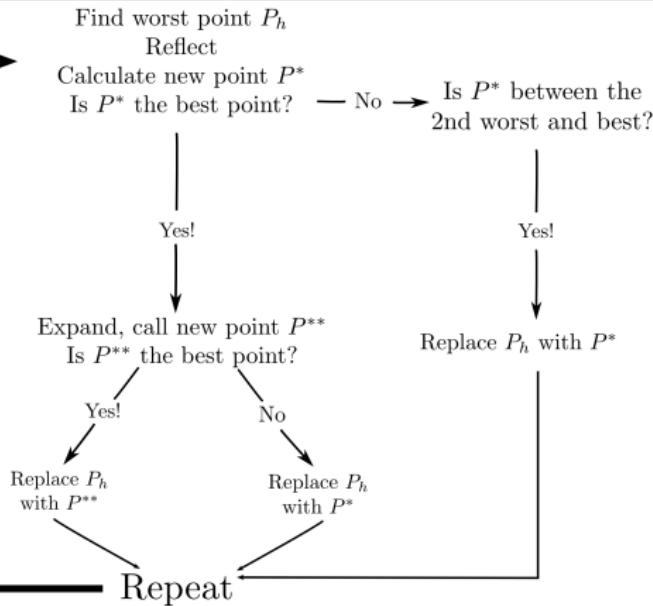
Nelder-Mead Algorithm



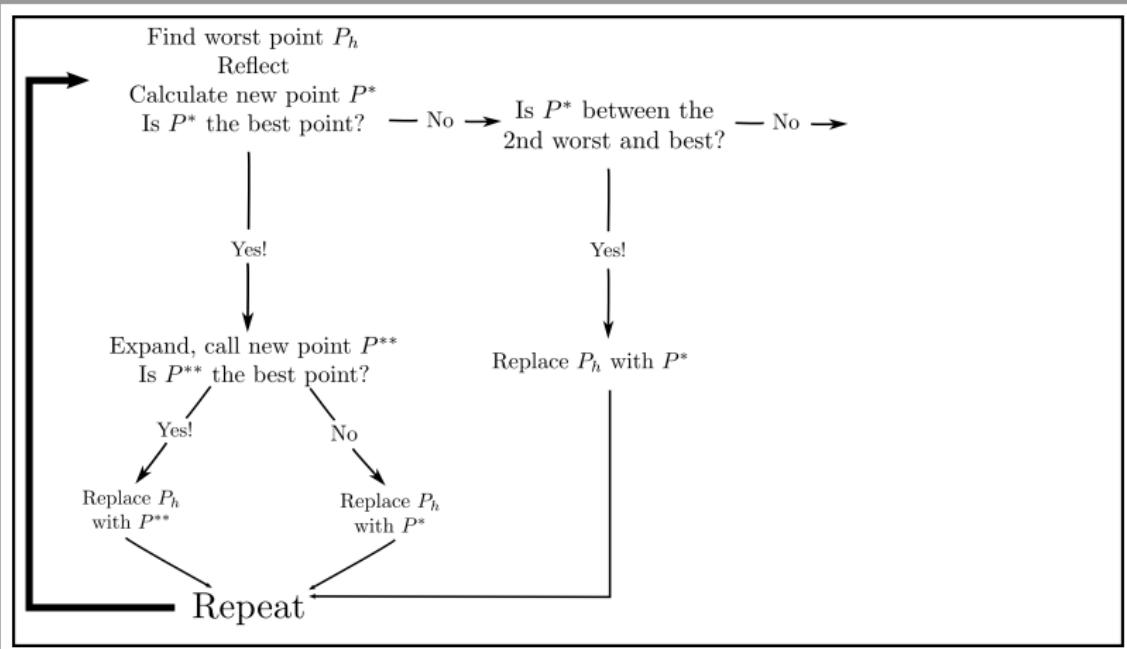
Nelder-Mead Algorithm



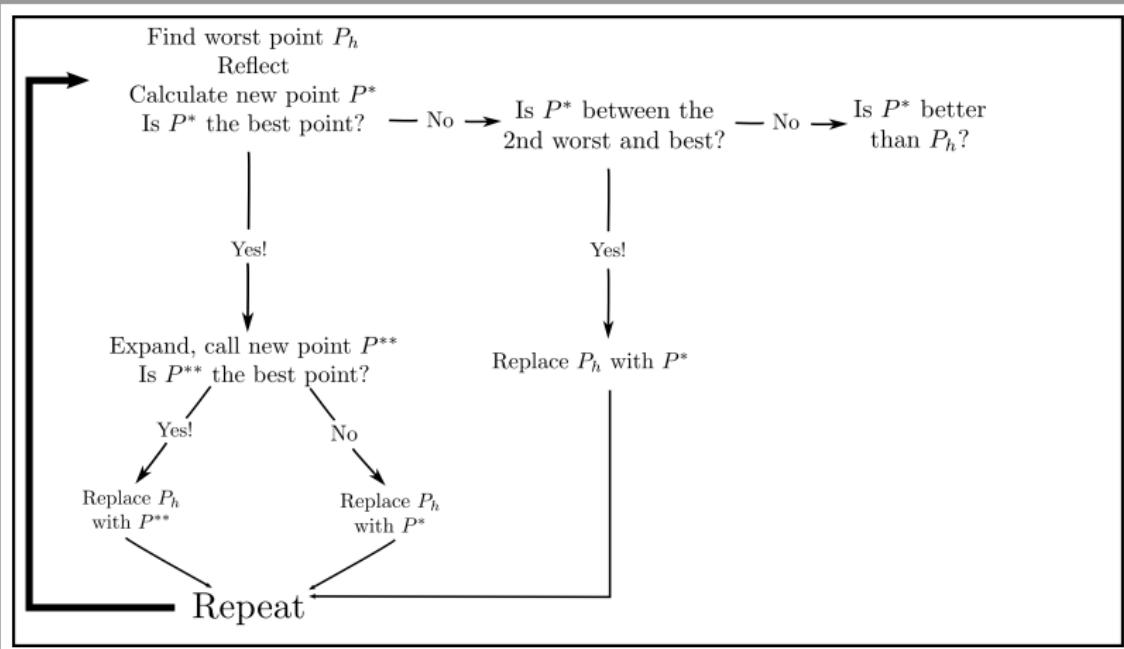
Nelder-Mead Algorithm



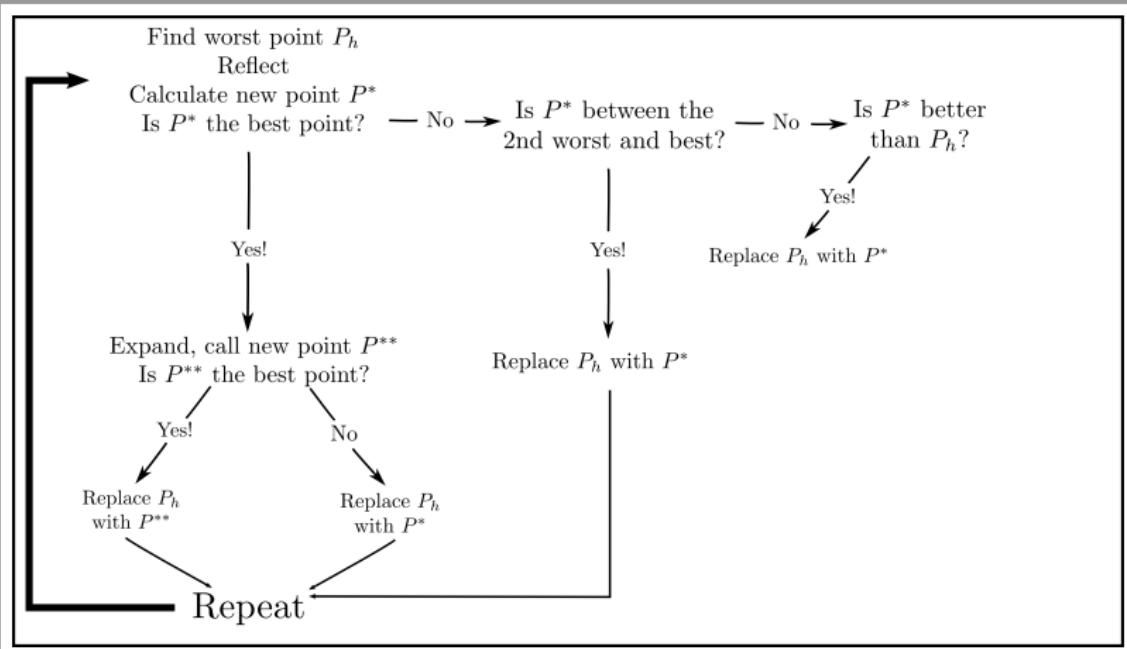
Nelder-Mead Algorithm



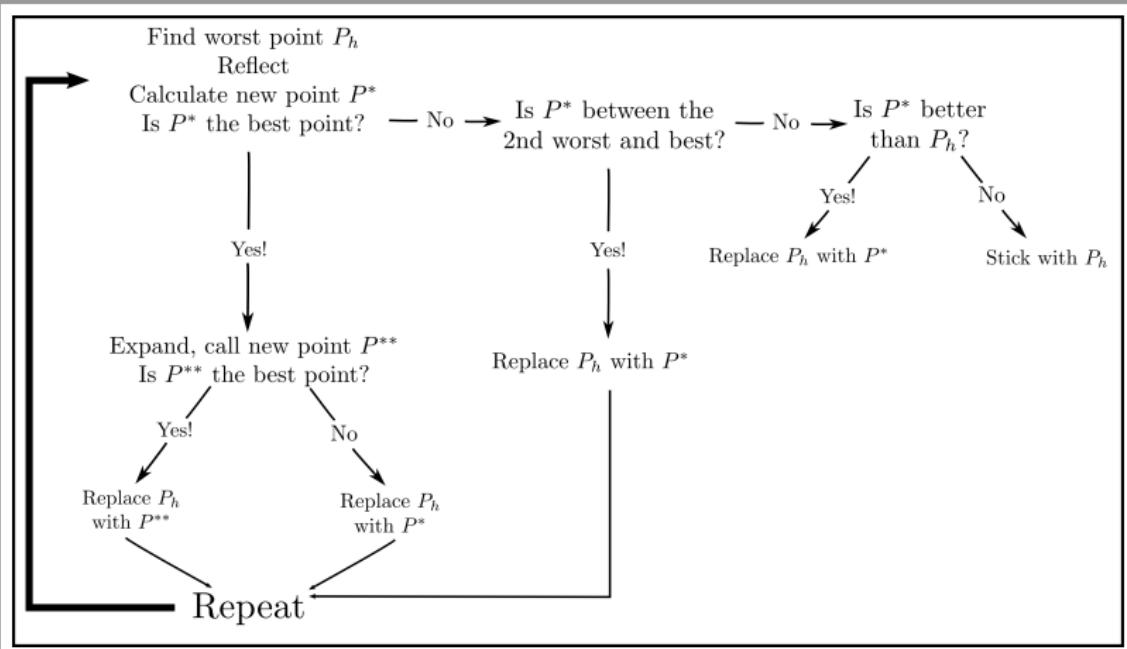
Nelder-Mead Algorithm



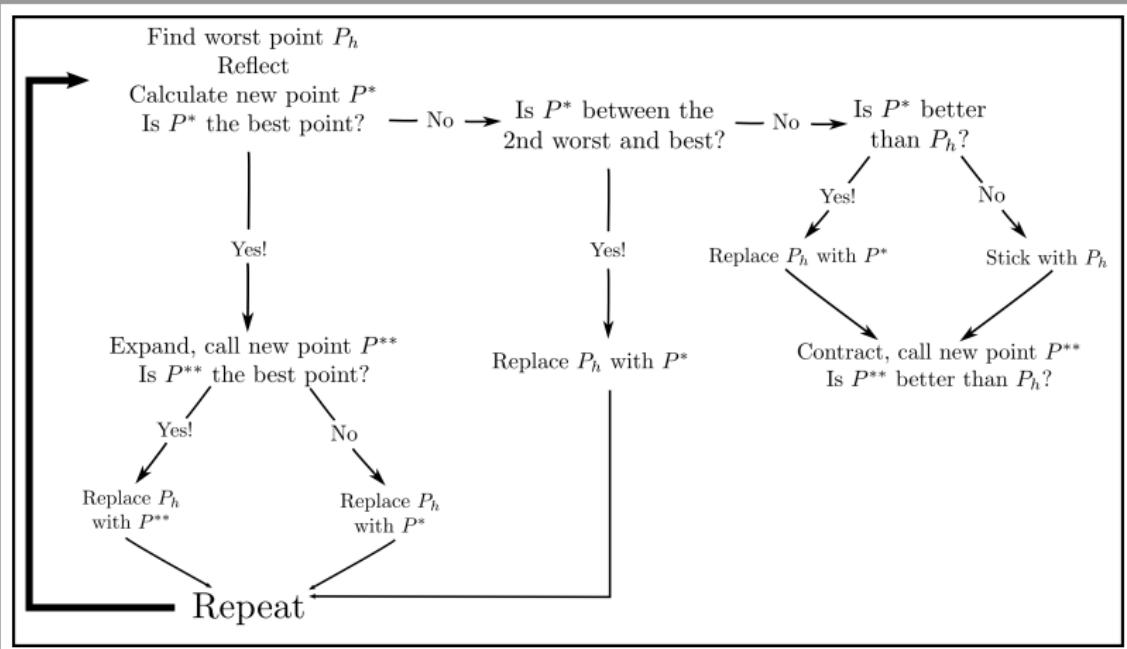
Nelder-Mead Algorithm



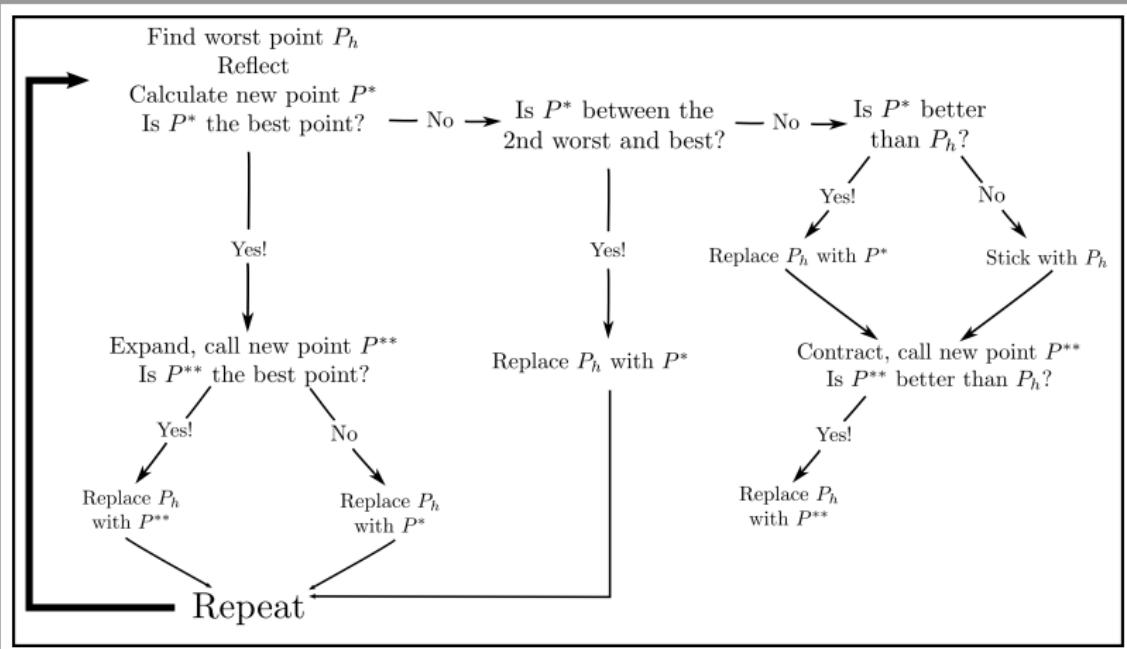
Nelder-Mead Algorithm



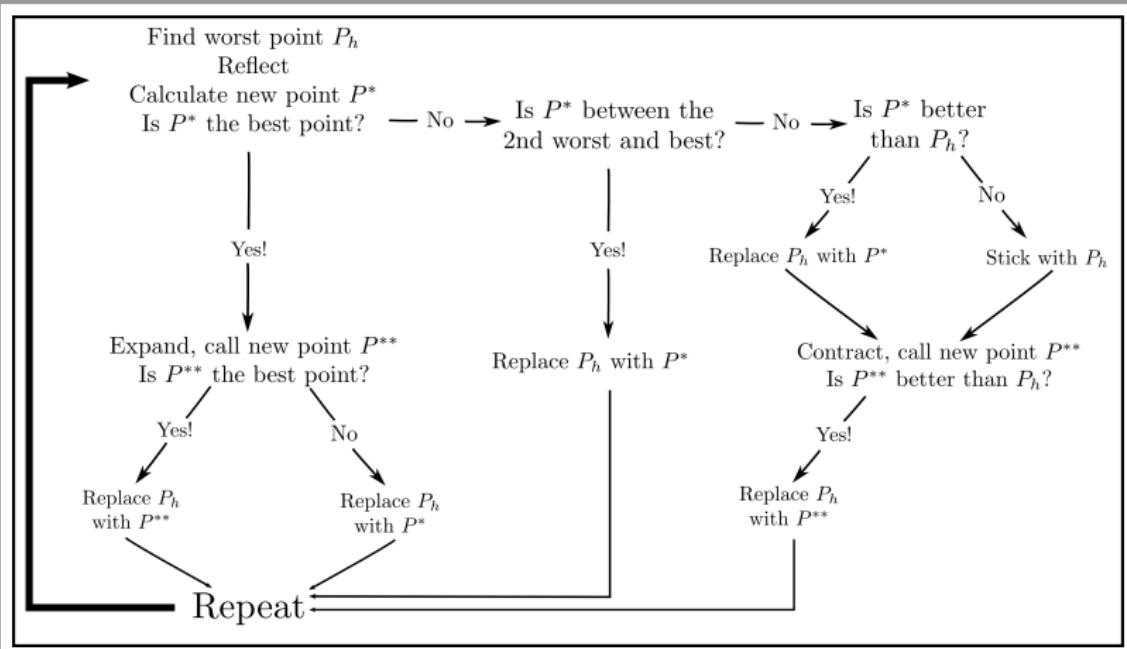
Nelder-Mead Algorithm



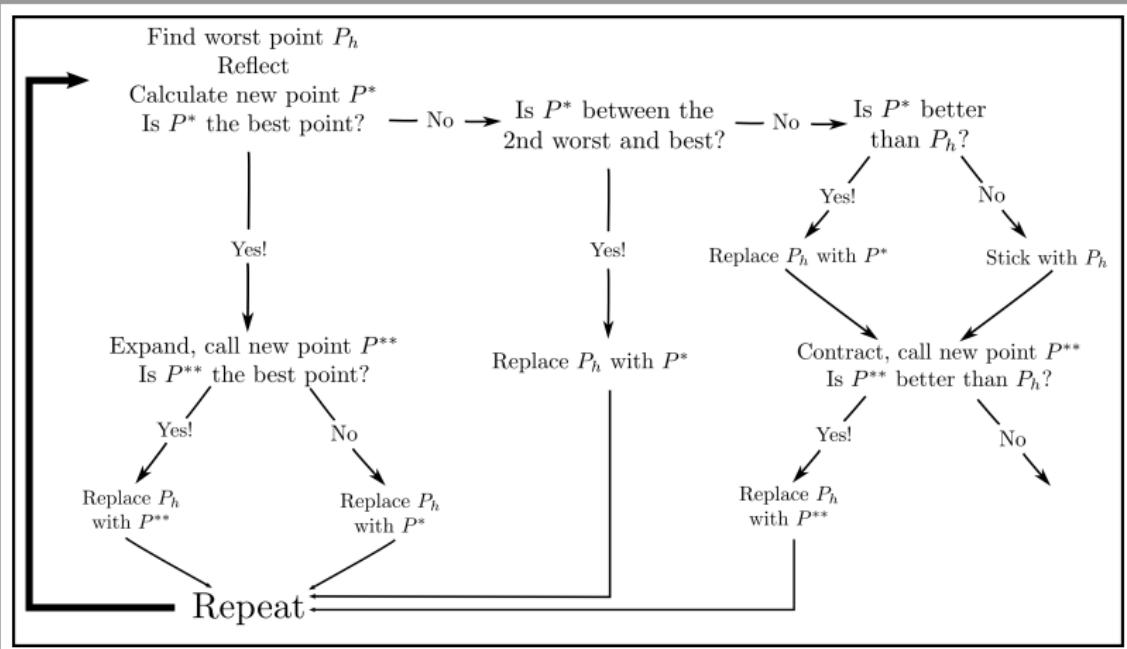
Nelder-Mead Algorithm



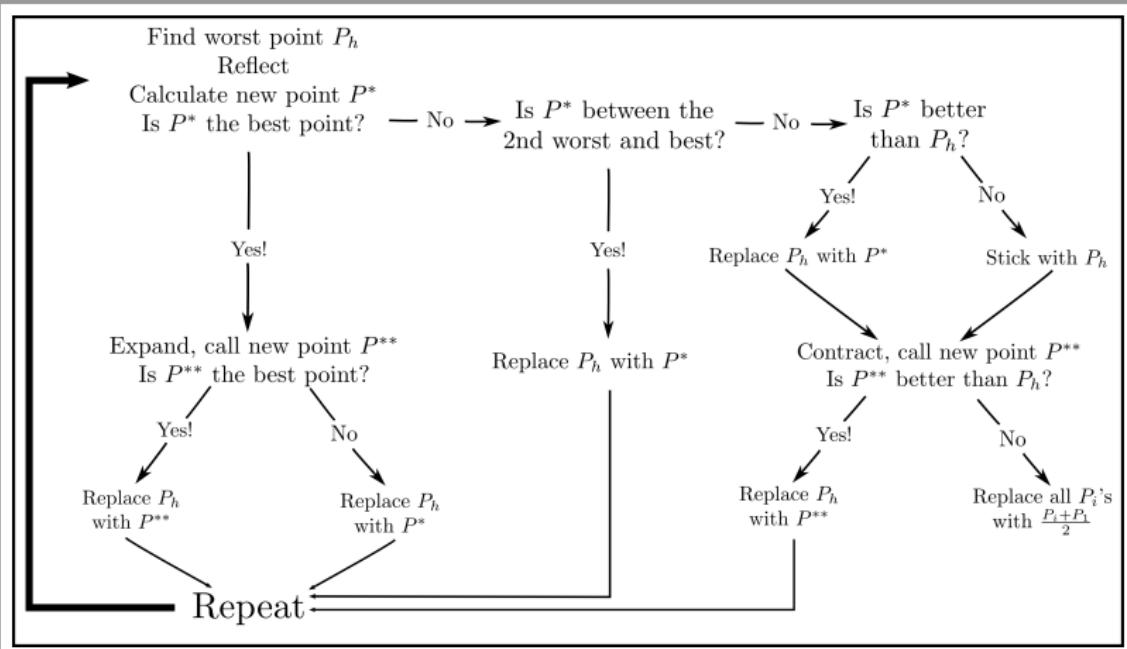
Nelder-Mead Algorithm



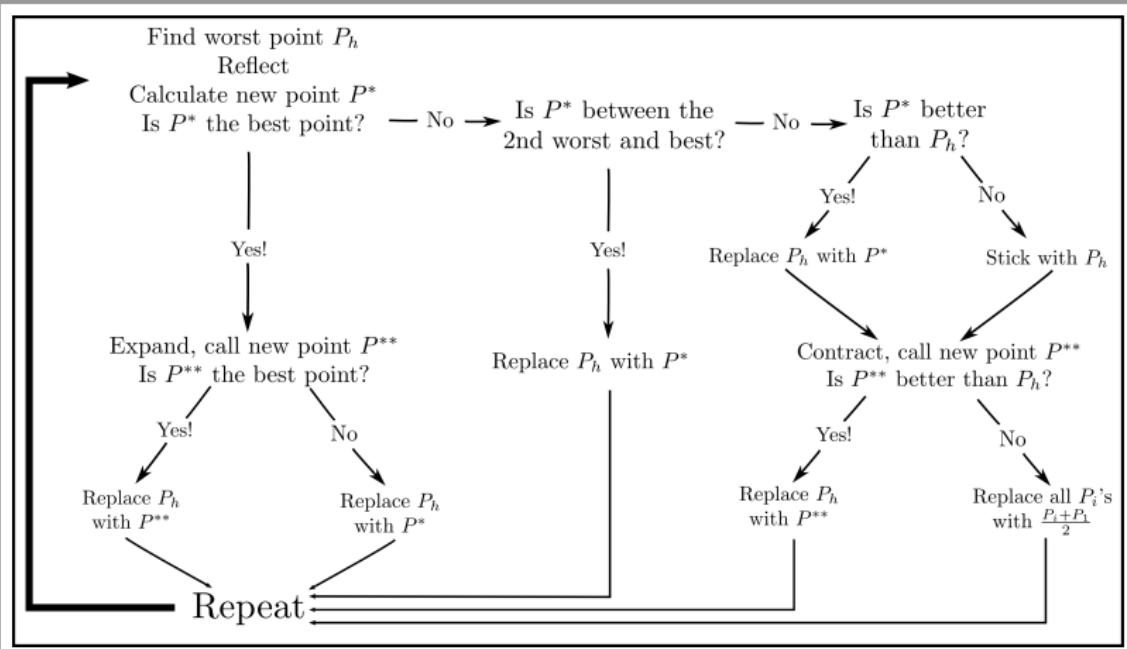
Nelder-Mead Algorithm



Nelder-Mead Algorithm



Nelder-Mead Algorithm



Examples

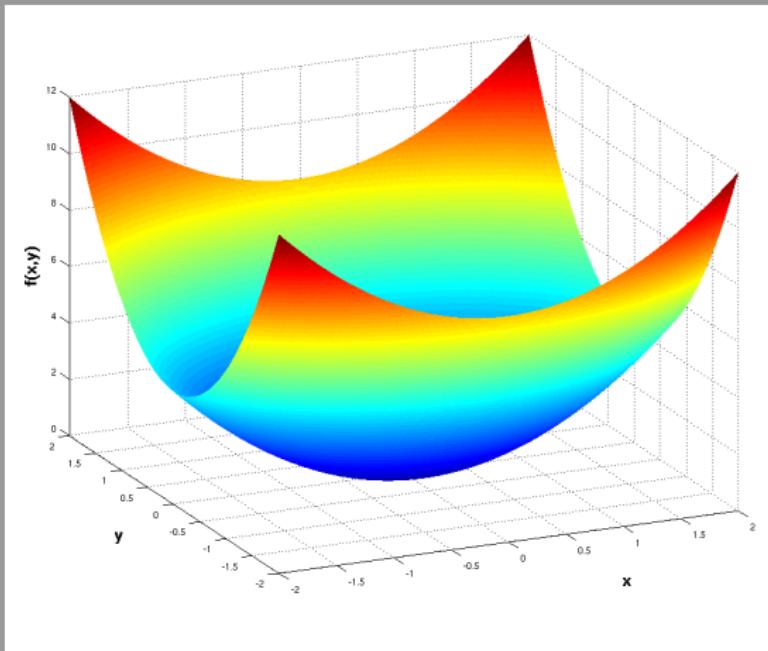
Examples

$$f(x) = x^4 + 2x^2 + 1$$

Examples

Examples

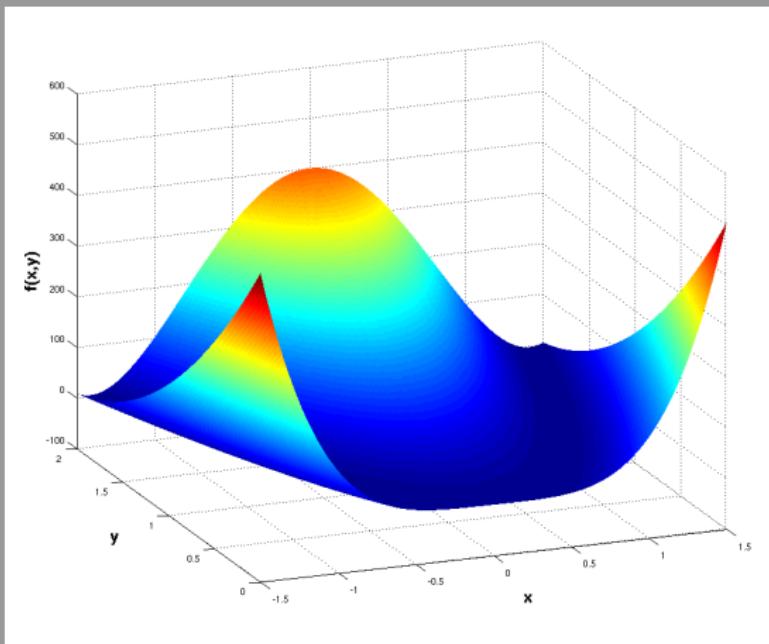
$$f(x, y) = x^2 + 2y^2$$



Examples

Examples

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$



Examples

Formal Convergence Theory

Formal Convergence Theory

Basically, there is none

Formal Convergence Theory

Basically, there is none

- There is some rigorous convergence results available for convex functions in low dimensions¹

¹J.C. Lagarias, 1998

Formal Convergence Theory

Basically, there is none

- There is some rigorous convergence results available for convex functions in low dimensions¹
- Example: For strictly convex functions f , shrink steps will never occur

¹J.C. Lagarias, 1998

General Advantages

General Advantages

- Only requires 1 or 2 function evaluations per iteration
(provided no shrink steps occur)

General Advantages

- Only requires 1 or 2 function evaluations per iteration
(provided no shrink steps occur)
- Converges quickly to an approximate minimum

General Advantages

- Only requires 1 or 2 function evaluations per iteration
(provided no shrink steps occur)
- Converges quickly to an approximate minimum
- Is extremely easy to implement

Disadvantages

Disadvantages

- No general convergence theory!

Disadvantages

- No general convergence theory!
- Highly sensitive to placement and size of initial simplex

Disadvantages

- No general convergence theory!
- Highly sensitive to placement and size of initial simplex
- Can take a large number of iterations with only very slight improvements

Conclusion

Conclusion

- Nelder-Mead algorithm directly searches the factor space for a minimum

Conclusion

- Nelder-Mead algorithm directly searches the factor space for a minimum
- Nelder-Mead is a good choice if you are optimizing a function that

Conclusion

- Nelder-Mead algorithm directly searches the factor space for a minimum
- Nelder-Mead is a good choice if you are optimizing a function that
 - has no derivatives

Conclusion

- Nelder-Mead algorithm directly searches the factor space for a minimum
- Nelder-Mead is a good choice if you are optimizing a function that
 - has no derivatives
 - is difficult to evaluate

Conclusion

- Nelder-Mead algorithm directly searches the factor space for a minimum
- Nelder-Mead is a good choice if you are optimizing a function that
 - has no derivatives
 - is difficult to evaluate
 - only needs an approximate solution (and quickly!)

Conclusion

- Nelder-Mead algorithm directly searches the factor space for a minimum
- Nelder-Mead is a good choice if you are optimizing a function that
 - has no derivatives
 - is difficult to evaluate
 - only needs an approximate solution (and quickly!)

Questions?