A hybrid direct search and model-based derivative-free optimization method with dynamic decision processing

Dominic (Zhongda) Huang

Supervisor: Dr. Warren Hare

The University of British Columbia - Okanagan Campus

June 17, 2022

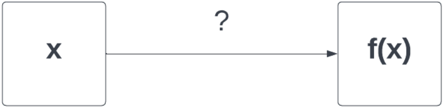
Outline

Purpose

- Develop the framework of a hybrid DFO method - the DQL method and its search strategies.
- Conduct the convergence analysis and numerical experiment for the DQL method.
- Design the SMART DQL method by the integration of decision processing.
- Apply the SMART DQL method on the solid tank design problem.

Derivative-Free and Black-Box Optimization

- Derivative-Free: No derivative information is used or available.
- Black-Box Function: The evaluation process is hidden.



Motivation

- We have a lot of well-developed methods for black-box problems.
- Due to the nature of black-box problems, we do not know how to choose the appropriate method.
- Inspired by the RQLIF method [Manno et al., 2020], we combine the strengths of three kinds of search strategies into one method.
- Allow the method to choose search strategies *dynamically* and *adaptively*.

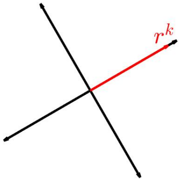
DQL Method Framework

1. Initialize
2. Direct Search Step
3. Quadratic Search Step
4. Linear Search Step
5. Update, Stop or Loop

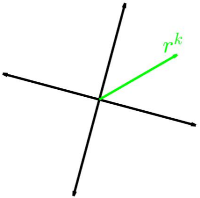
Framework of the Direct Step

Search on the directions of rotated positive and negative coordinate direction by a step length of δ^k.

- Desired Direction



- Undesired Direction



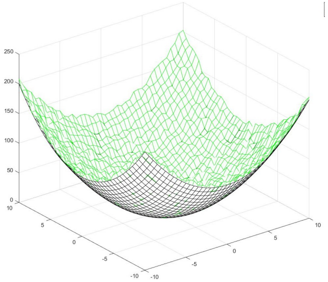
Direct Step Strategy 1: Random Rotation

The rotation directions alternates between two options:

- the coordinate directions.
- a random rotation.

Framework of the Quadratic Step

Extract the quadratic information from the previously evaluated candidates within the trust region.

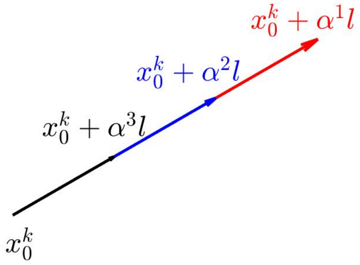


- Least-Squares Quadratic Model
- Approximate Newton's Method

Framework of the Linear Step

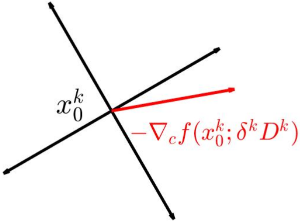
$$\mathbb{L} = \{x_0 + \alpha^j l\}$$

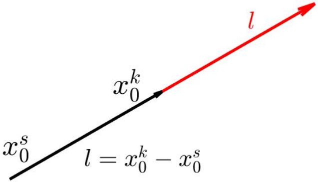
- Search direction $l \in \mathbb{R}^n$
- Linear search steps $\{\alpha^j \in \mathbb{R}\}$



Linear Step Strategies: Determine Search Direction

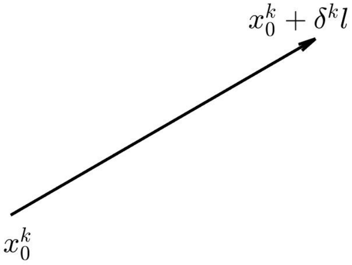
- Approximate Steepest Descent $l = -\nabla_c f(x_0^k; \delta^k D^k)$
- Last descent $l = x_0^k - x_0^s$



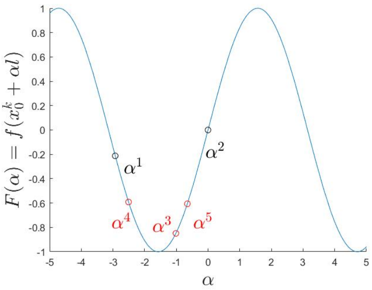


Linear Step Strategies: Determine Search Step Length

- Step Length δ^k



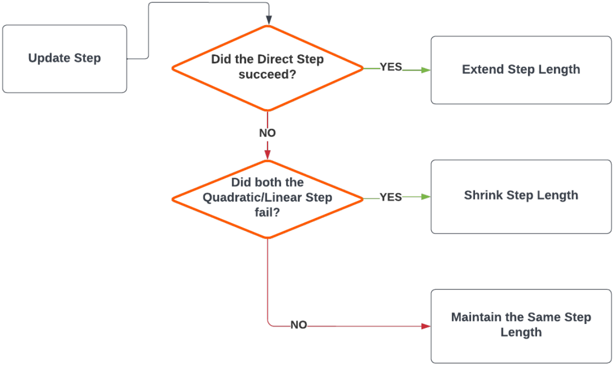
- Safeguarded Bracket Search [Mifflin and Strodiot, 1989]



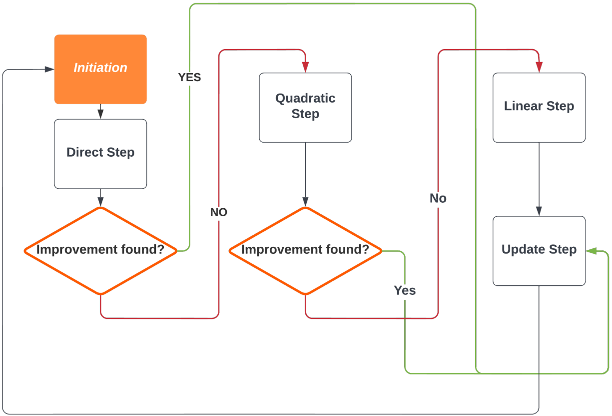
Linear Step Strategies

Label	Search Direction l	Search Step α
Strategy 1	Steepest Descent	One Step (δ^k)
Strategy 2	Steepest Descent	Bracket Search
Strategy 3	Last Descent	One Step (δ^k)
Strategy 4	Last Descent	Bracket Search

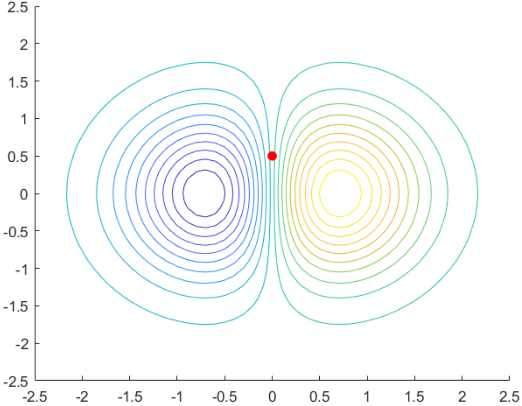
Framework of the Update Step



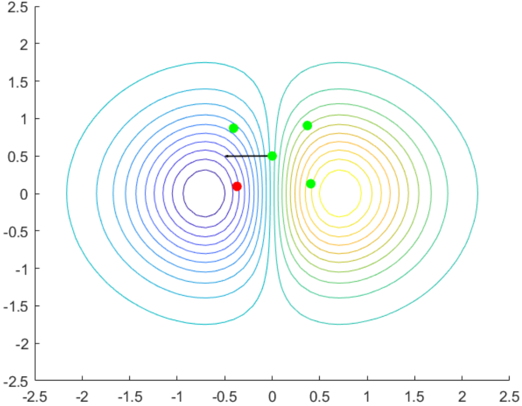
Flow Diagram of the DQL method



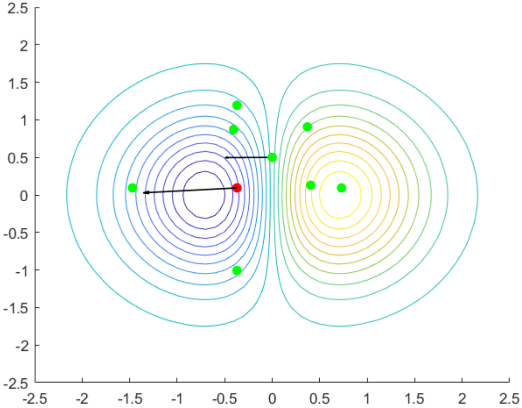
Demo of the DQL method



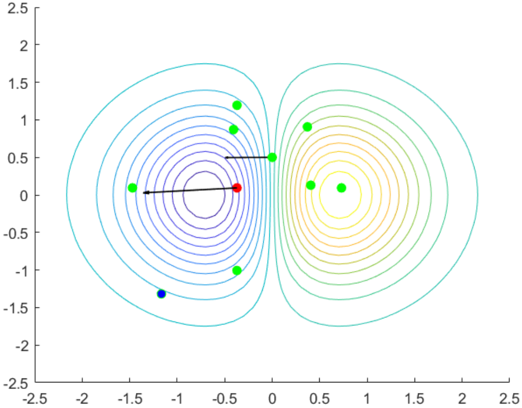
Demo of the DQL method



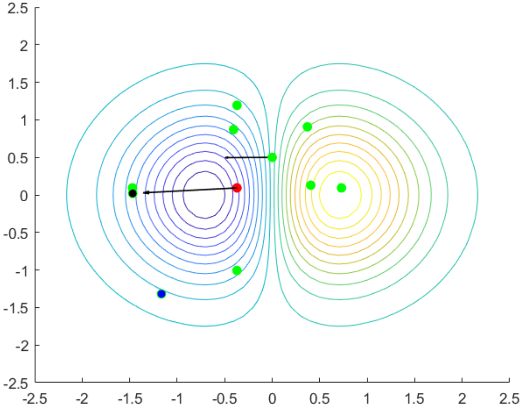
Demo of the DQL method



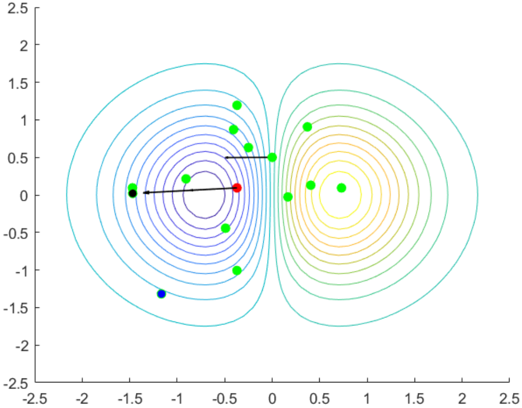
Demo of the DQL method



Demo of the DQL method



Demo of the DQL method



Convergence Analysis

Theorem 1

Let function $f : \mathbb{R}^n \to \mathbb{R}$ has compact level set $L(x^0)$. In addition, let ∇f be Lipschitz continuous in an open set containing $L(x^0)$. Then the DQL *method results in*

$$\liminf_{k \to +\infty} \left\| \nabla f(x^k) \right\| = 0,$$

and $\{x^k\}$ has a limit point $x^\star$ for which $\nabla f(x^\star) = 0$.

Proof.

The proof can be found in the thesis [Huang, 2022, Thm 3.5]. □

Performance Benchmark

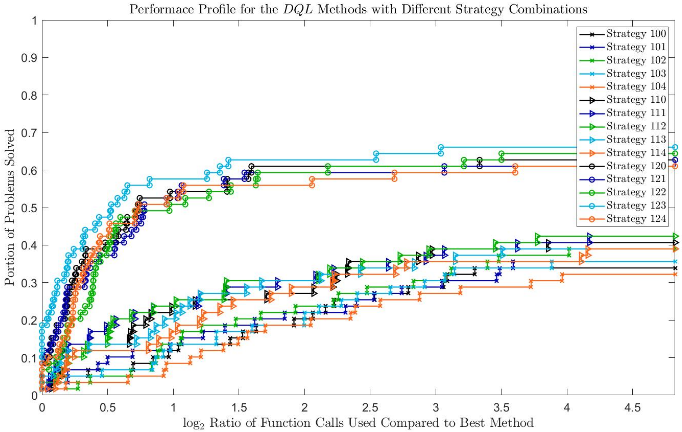
- Direct Step
  1 option: Strategy 1
- Quadratic Step
  3 options: Disable, Strategy 1-2
- Linear Step
  5 options: Disable, Strategy 1-4

Is there a winner among 15 combinations?

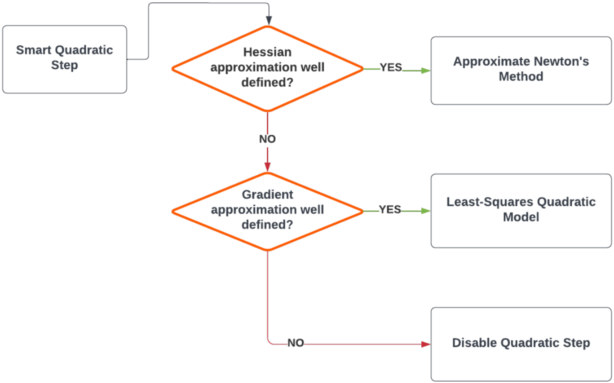
Performance Benchmark: Stopping Conditions

Parameter	Value
ϵ_{∇}	10^{-6}
$\epsilon_{\mathrm{MAX\_STEP}}$	10^{-3}
$\epsilon_{\mathrm{MIN\_STEP}}$	10^{-12}
$\mathrm{MAX\_SEARCH}$	10000

Performance Benchmark: Numerical Result



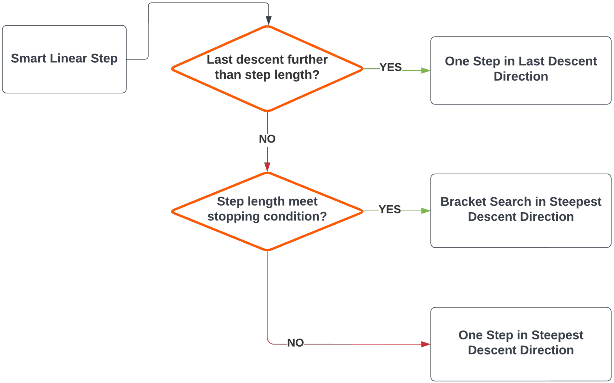
Smart Quadratic Step



Smart Linear Step

- One Step in Last Descent Direction
  - Best Exploration Ability
- Bracket Search in Steepest Descent Direction
  - Best Exploitation Ability
- One Step in Steepest Descent Direction
  - Simple and Efficient

Smart Linear Step

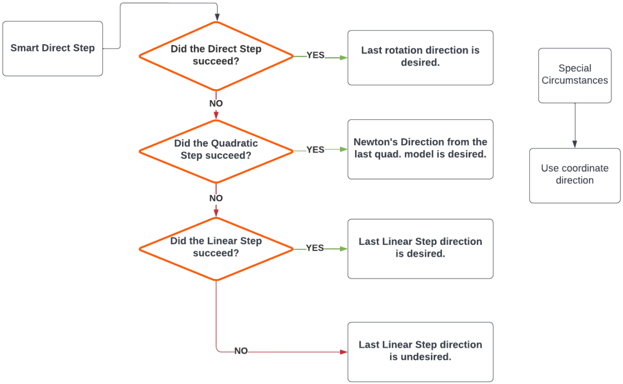


Smart Direct Step

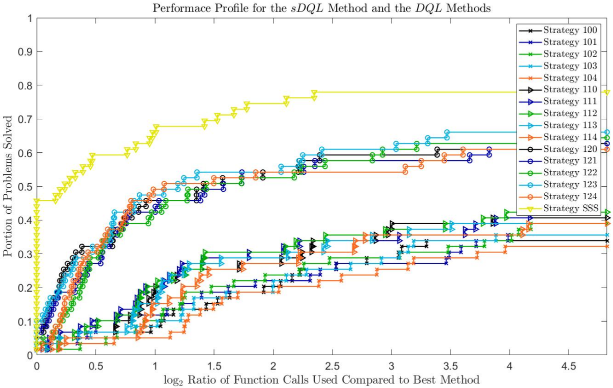
What information can we extract from the last iteration?

- Direct Step
  Is r^{k-1} a good rotation direction?
- Quadratic Step
  Is m^{k-1} a good quadratic model?
- Linear Step
  Is l^{k-1} a good linear search direction?

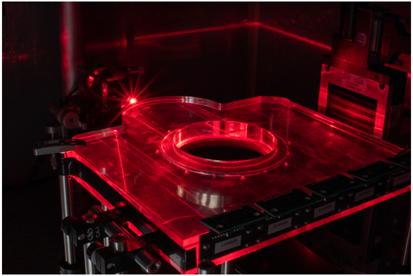
Smart Direct Step



Performance of SMART DQL Method



Background



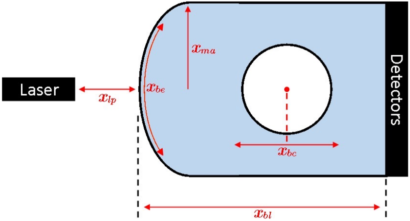


Figure: Solid Tank Design (Picture by Andy Oglivy).

Background

$$x = \begin{bmatrix} x_{bl} & x_{bc} & x_{lp} & x_{ma} & x_{be} \end{bmatrix}^\top \in \mathbb{R}^5$$

$$x_{bl} \in [200, 400]$$

$$x_{bc} \in [-30, 30]$$

$$x_{lp} \in [40, 100]$$

$$x_{ma} \in [40, 80]$$

$$x_{be} \in [0, 1]$$

$$\max\{F(x) | x \in C\}$$

Experiment Results

Table: Experimental Results for Solid Tank Design Problem

	Water	FlexDos3D	ClearView™
SMART DQL Method	2.7676	2.9360	2.9522
Grid Search Method	2.5611	2.9105	2.8694

Conclusion

DQL method

- is a local DFO method.
- is able to combine multiple search strategies.
- is converging to local optima for some functions.

SMART DQL method

- is built under the framework of DQL method.
- is able to choose search strategies dynamically and adaptively.
- is faster and more robust than any simple combinations from our DQL method study.
- is more reliable and efficient in real-world application as compared to the Grid Search Method

Future Development

- Develop the global search ability.
- Integrate more search strategies.
- Design a more sophisticated decision tree.
- Specialize the decision making mechanism for different real-world applications.

Reference

Thank you!

Huang, Z. (2022).
A hybrid direct search and model-based derivative-free optimization method with dynamic decision processing.

Manno, A., Amaldi, E., Casella, F., and Martelli, E. (2020).
A local search method for costly black-box problems and its application to CSP plant start-up optimization refinement.
Optimization and Engineering, 21(4):1563–1598.

Mifflin, R. and Strodiot, J.-J. (1989).
A bracketing technique to ensure desirable convergence in univariate minimization.
Mathematical programming, 43(1):117–130.