PSO Algorithm for Optimum Well Placement subject to Realistic Field Development Constraints

Mansoureh Jesmani, NTNU, Mathias C. Bellout, NTNU,
Remus Hanea, Statoil, and Bjarne Foss, NTNU

June 10, 2015

NTNU
Norwegian University of
Science and Technology

Well Placement Problem

- Common formulation of well placement problem:

$$\max_{\boldsymbol{\zeta},\mathbf{u}^n} [J = \sum_{n=0}^{N-1} L^n(\mathbf{x}^{n+1}, \boldsymbol{\zeta}, \mathbf{u}^n)],$$

subject to:

$$\boldsymbol{\zeta}^d \leq \boldsymbol{\zeta} \leq \boldsymbol{\zeta}^u,$$

$$\mathbf{u}^d \leq \mathbf{u}^n \leq \mathbf{u}^u,$$

$$\mathbf{x}^0 = \mathbf{x}_0,$$

$$g^n(\mathbf{x}^{n+1}, \mathbf{x}^n, \boldsymbol{\zeta}, \mathbf{u}^n) = 0, \; n = 0, 1, \cdots, N-1.$$

Well Placement Problem

- Common formulation of well placement problem:

$$\max_{\boldsymbol{\zeta},\mathbf{u}^n} \left[J = \sum_{n=0}^{N-1} L^n(\mathbf{x}^{n+1}, \boldsymbol{\zeta}, \mathbf{u}^n)\right],$$

subject to:

$$\boxed{\boldsymbol{\zeta}^d \leq \boldsymbol{\zeta} \leq \boldsymbol{\zeta}^u,}$$

$$\mathbf{u}^d \leq \mathbf{u}^n \leq \mathbf{u}^u,$$

$$\mathbf{x}^0 = \mathbf{x}_0,$$

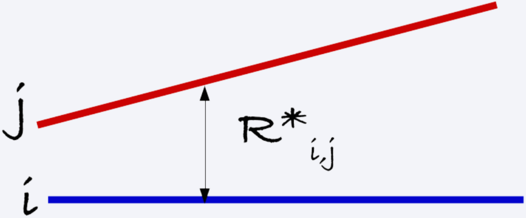
$$g^n(\mathbf{x}^{n+1}, \mathbf{x}^n, \boldsymbol{\zeta}, \mathbf{u}^n) = 0, \; n = 0, 1, \cdots, N-1.$$

Motivation

- Problem: Engineering experiences are not included.
- Valuable solution depends on
  - Identification of limitations,
  - Translation of them into constraints.
- The success of the optimization effort relies on
  - Efficient search algorithm,
  - Constraint-handling techniques.

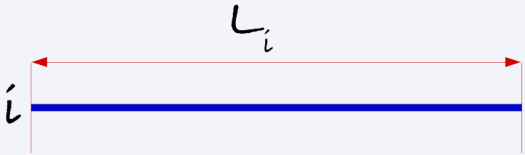
Well Placement Constraints

- Well distance $C_{wd} : R^*_{i,j} \geq d_{min}$
- Well length $C_{wl} : L_i = \|\zeta_i^h - \zeta_i^t\|_2,\ l^i_{min} \leq L_i \leq l^i_{max}$
- Reservoir bound $C_{rb} : \zeta_i^h \in R_i^h, \quad \zeta_i^t \in R_i^t$
- Well orientation
  $C_{wo} : \theta_{i,j} = \arccos\left|\frac{(\zeta_i^h-\zeta_i^t)\cdot(\zeta_j^h-\zeta_j^t)}{\|\zeta_i^h-\zeta_i^t\|_2\|\zeta_j^h-\zeta_j^t\|_2}\right| \leq \theta_{max}$



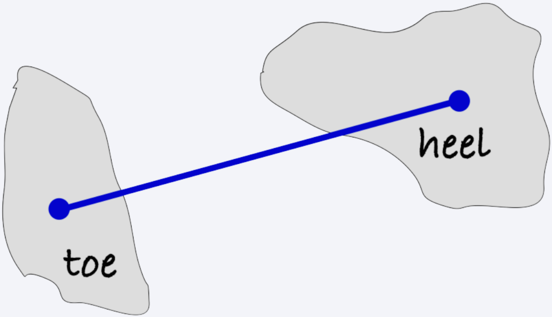
Well Placement Constraints

- Well distance $C_{wd} : R^*_{i,j} \geq d_{min}$
- Well length $C_{wl} : L_i = \|\zeta^h_i - \zeta^t_i\|_2,\ l^i_{min} \leq L_i \leq l^i_{max}$
- Reservoir bound $C_{rb} : \zeta^h_i \in R^h_i, \quad \zeta^t_i \in R^t_i$
- Well orientation
  $C_{wo} : \theta_{i,j} = \arccos \left| \frac{(\zeta^h_i - \zeta^t_i)\cdot(\zeta^h_j - \zeta^t_j)}{\|\zeta^h_i - \zeta^t_i\|_2 \|\zeta^h_j - \zeta^t_j\|_2} \right| \leq \theta_{max}$



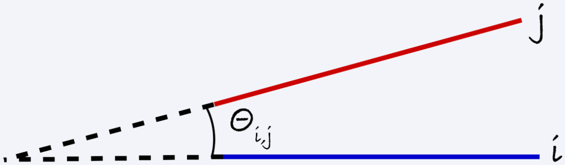
Well Placement Constraints

- Well distance C_{wd} : $R^*_{i,j} \geq d_{min}$
- Well length C_{wl} : $L_i = \|\zeta_i^h - \zeta_i^t\|_2,\ l^i_{min} \leq L_i \leq l^i_{max}$
- Reservoir bound C_{rb} : $\zeta_i^h \in R_i^h, \quad \zeta_i^t \in R_i^t$
- Well orientation
  $C_{wo} : \theta_{i,j} = \arccos \left| \frac{(\zeta_i^h - \zeta_i^t)\cdot(\zeta_j^h - \zeta_j^t)}{\|\zeta_i^h - \zeta_i^t\|_2 \|\zeta_j^h - \zeta_j^t\|_2} \right| \leq \theta_{max}$



Well Placement Constraints

- Well distance C_{wd} : $R^*_{i,j} \geq d_{min}$
- Well length C_{wl} : $L_i = \|\zeta_i^h - \zeta_i^t\|_2,\ l^i_{min} \leq L_i \leq l^i_{max}$
- Reservoir bound C_{rb} : $\zeta_i^h \in R_i^h, \quad \zeta_i^t \in R_i^t$
- Well orientation
  $C_{wo} : \theta_{i,j} = \arccos\left|\frac{(\zeta_i^h - \zeta_i^t)\cdot(\zeta_j^h - \zeta_j^t)}{\|\zeta_i^h - \zeta_i^t\|_2 \|\zeta_j^h - \zeta_j^t\|_2}\right| \leq \theta_{max}$



General Form of Well Placement Problem

$$\min -\text{NPV},$$

subject to:

$$C_i(\boldsymbol{\zeta}) \geq 0, \quad i \in \{wd, wl, rb, wo\},$$

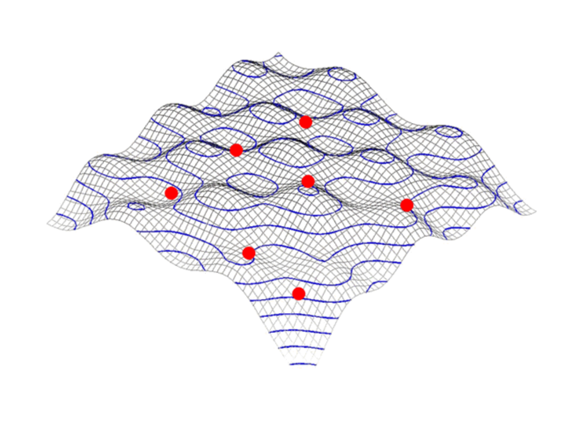
$$\mathbf{u}^d \leq \mathbf{u}^n \leq \mathbf{u}^u,$$

$$\mathbf{x}^0 = \mathbf{x}_0,$$

$$g^n(\mathbf{x}^{n+1}, \mathbf{x}^n, \boldsymbol{\zeta}, \mathbf{u}^n) = 0, \; n = 0, 1, \cdots, N-1.$$

Particle Swarm Optimization (PSO)

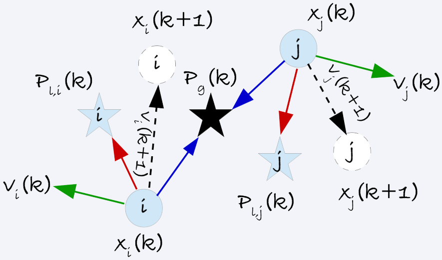
- PSO provides comparable or better results than binary GA (Onwunalu and Durlofsky, 2010).



Particle Swarm Optimization (PSO)

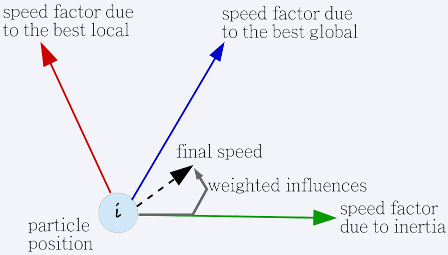
$$\boldsymbol{\nu}_i(k+1) = \boldsymbol{\nu}_i(k) + c_1\rho_1(k)(\mathbf{p}_{l,i}(k) - \mathbf{x}_i(k)) + c_2\rho_2(k)(\mathbf{p}_{g,i}(k) - \mathbf{x}_i(k)),$$

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \boldsymbol{\nu}_i(k+1).$$



Inertia Weight

$$\hat{\boldsymbol{\nu}}_i(k+1) = w(k)\boldsymbol{\nu}_i(k) + c_1\rho_1(k)(\mathbf{p}_{l,i}(k) - \mathbf{x}_i(k)),$$
$$+ c_2\rho_2(k)(\mathbf{p}_{g,i}(k) - \mathbf{x}_i(k)),$$
$$\nu_i^j(k+1) = \mathsf{sign}(\hat{\nu}_i^j(k+1)) \min\{|\hat{\nu}_i^j(k+1)|, \nu_{max}^j\},$$
$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \boldsymbol{\nu}_i(k+1),$$
$$\nu_{max}^j = \lambda(u^j - l^j), \quad w(k) = w_0 - \frac{k}{K}(w_0 - w_1).$$



Method 1: Penalty function

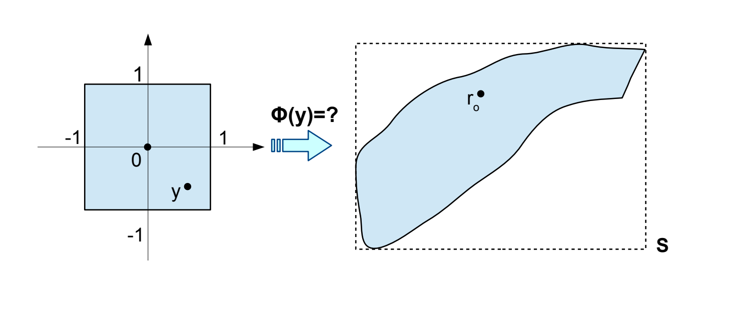
- Merit function

$$\phi_1(\zeta, \mu) = -(\mathsf{NPV})_{sc} + \mu \sum_i \max\{0, -(C_i)_{sc}\},$$

- Penalty parameter (μ) grows with iteration number.

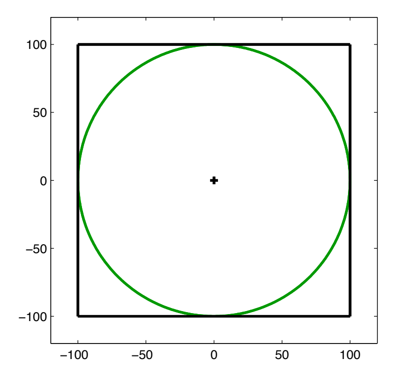
Method 2: Decoder

- A homomorphous mapping between an n-dimensional cube and a feasible search space (Koziel and Michalewicz, 1999).



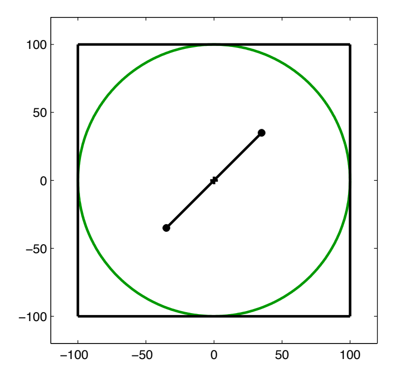
Introducing Decoder for Placing one Horizontal Well

- Constraints: Both toe and heel should stay in the circle (feasible region),
- Variables: Cartesian coordinate for both heel (x_h, y_h) and toe (x_t, y_t)



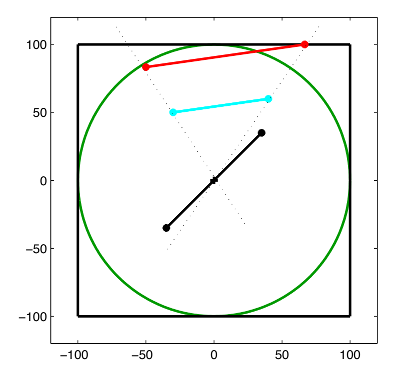
Introducing Decoder for Placing one Horizontal Well

- Step 1: Define reference $r_0 = \begin{bmatrix} 35 & 35 & -35 & -35 \end{bmatrix}$
- Step 2: The input of decoder should stay in the cube $[-1, 1]^4$ $y = \begin{bmatrix} 0.4 & 0.6 & -0.3 & 0.5 \end{bmatrix}$



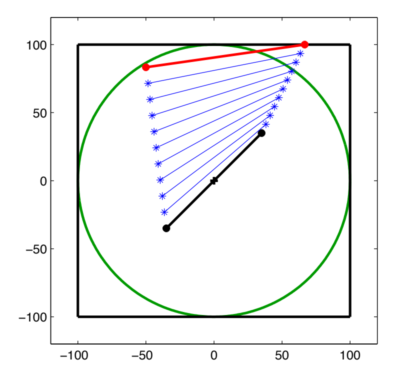
Introducing Decoder for Placing one Horizontal Well

- Step 3: Calculate
  $y/y_{max} = \frac{1}{0.6}\begin{bmatrix}0.4 & 0.6 & -0.3 & 0.5\end{bmatrix}$
- Step 4: Map $g(y)$ to s
  $s = g(y/y_{max}) = \begin{bmatrix}66.7 & 100 & -50 & 83.3\end{bmatrix}$
  $g(y) = (y - \frac{(u-l)}{2}) + \frac{u+l}{2}$



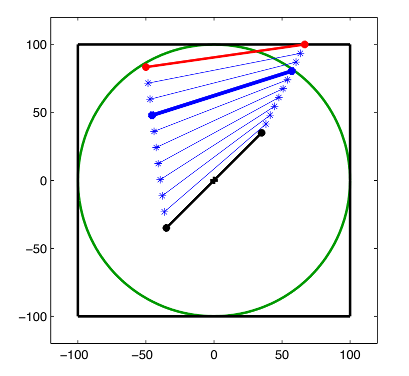
Introducing Decoder for Placing one Horizontal Well

- Step 5: Define line segment between s and $\mathbf{r_0}$: $L(r_0, s) = r_0 + t(s - r_0)$
- Step 6: Find t_0 where L intersects the boundary of circle: $t_0 = 0.72$



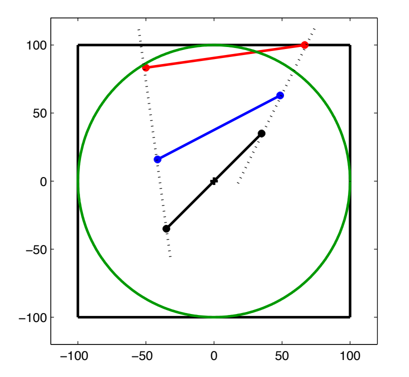
Introducing Decoder for Placing one Horizontal Well

- Step 5: Define line segment between s and $\mathbf{r_0}$:
  $L(r_0, s) = r_0 + t(s - r_0)$
- Step 6: Find t_0 where L intersects the boundary of circle: $t_0 = 0.72$



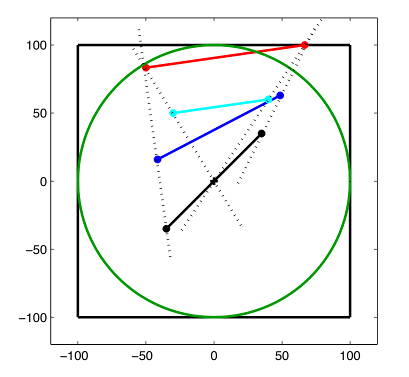
Introducing Decoder for Placing one Horizontal Well

- Step 7: Calculate $\phi(y)$:
  $\phi(y) = r_0 + y_{max} t_0 (s - r_0)$



Introducing Decoder for Placing one Horizontal Well

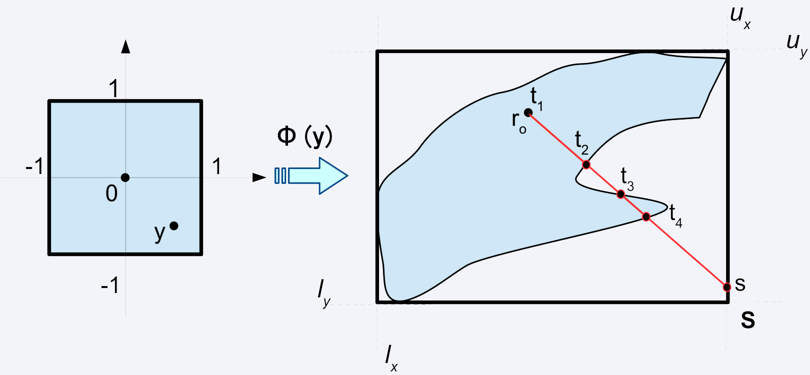
- $g(y)$
- $g(y/y_{max})$
- $r_0 + y_{max}t_0(s - r_0)$



Additional Constraints and Non-Convex Feasible Set

- Non-convex feasible set if:
  - Non-convex feasible region,
  - Include other constraints.
- In the case of non-convex feasible set:
  - All steps are same,
  - Several feasible interval: $[t_1, t_2], \cdots [t_{2k-1}, t_{2k}]$
  - Define new map: $\gamma : (0, 1] \to \cup_{i=1}^{k} (t_{2i-1}, t_{2i}]$

Non-Convex Feasible Space



$$\gamma : (0,1] \rightarrow \cup_{i=1}^{k} (t_{2i-1}, t_{2i}]$$

General Form of Decoder

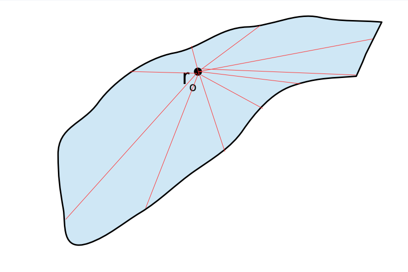
$$\phi(\mathbf{y}) = \begin{cases} \mathbf{r}_o + t_o \cdot (g(\mathbf{y}/y_{max} - \mathbf{r}_o)) & \text{if } \mathbf{y} \neq \mathbf{0} \\ \mathbf{r}_o & \text{if } \mathbf{y} = \mathbf{0} \end{cases}$$

$$y_{max} = \max_{i=1}^{n} \mid y_i \mid,$$

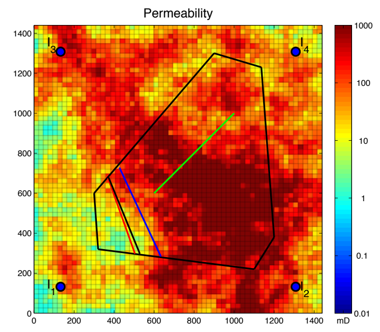
$$t_0 = \gamma(\mid y_{max} \mid).$$

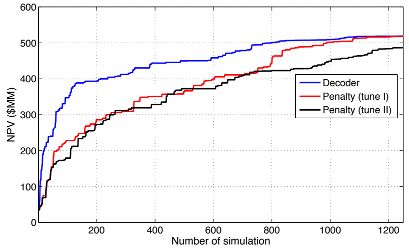
Decoder

- There is no need for any additional parameters,
- Always return a feasible solution,
- The map has locality feature, if any line segment, originates from the reference point, intersect the feasible search space just at one point.



Case Study I

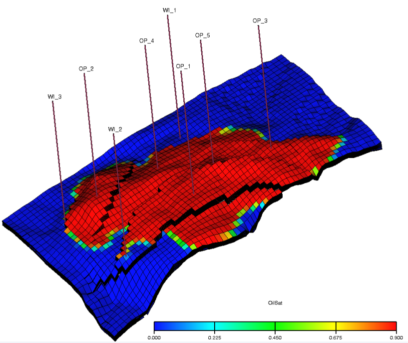


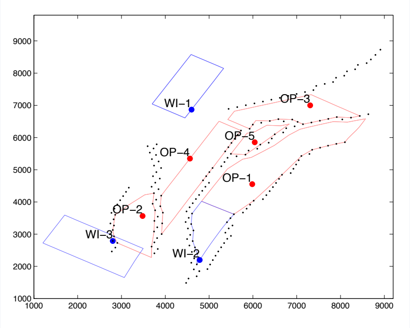


Algorithm	Best $(\times 10^8)$	Mean $(\times 10^8)$	Relative standard deviation (%)
Decoder	5.28	5.19	2.8
Penalty(tune I)	5.26	5.17	2.7
Penalty(tune II)	5.24	4.86	6.8

Case Study II: Regions Setting for Decoder

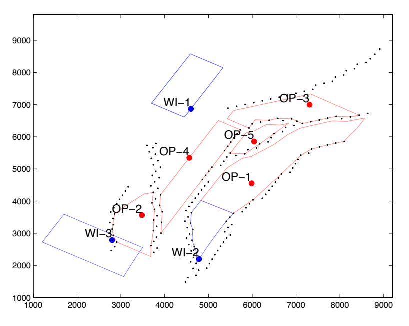
- 5 producers and 3 injectors,
- one realization,
- fixed production settings,
- $40 \times 64 \times 14 = 35,840$ grid cells.



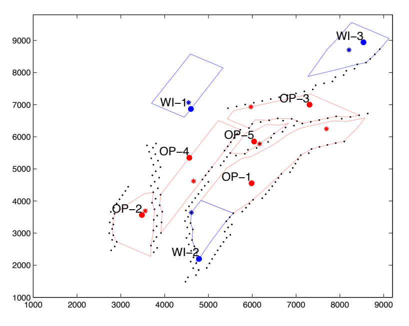


Case Study II: Regions Setting for Decoder

Initial search regions

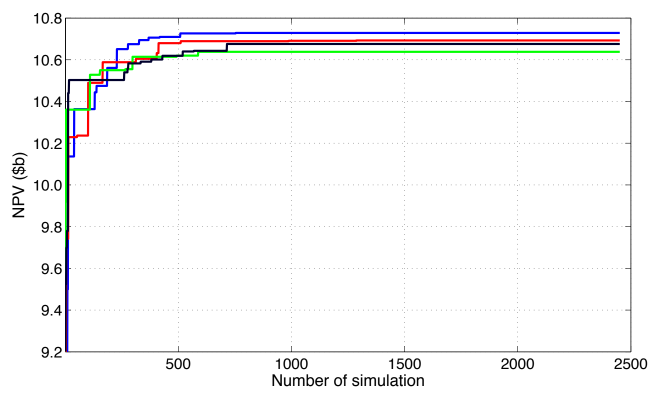


Improved search regions



Case Study II: results

$$n_p = 49, \; n_g = 50$$



Conclusion and Future Work

- Conclusion:
    - Improve the decision-making support by introducing realistic well placement constraints,
    - Couple decoder with the PSO algorithm,
    - Compare to the penalty method, the decoder can be used efficiently.
- Future work:
    - Applying this methodology to more complex cases,
    - Geological uncertainty,
    - Variable production strategy.

Thank You!

Koziel, S., Michalewicz, Z., Mar. 1999. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol. Comput. 7 (1), 19–44. URL `http://dx.doi.org/10.1162/evco.1999.7.1.19`

Onwunalu, J., Durlofsky, L., 2010. Application of a particle swarm optimization algorithm for determining optimum well location and type. Comput. Geosci. 14 (1), 183–198.