

**MULTIOBJECTIVE DESIGN AND OPTIMIZATION OF POLYMER FLOOD  
PERFORMANCE**

A Thesis

by

PEERAPONG EKKAWONG

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,  
Committee Members,  
Head of Department,

Akhil Datta-Gupta  
Michael King  
Mohanty Biyanak  
A. Daniel Hill

August 2013

Major Subject: Petroleum Engineering

Copyright 2013 Peerapong Ekkawong

## **ABSTRACT**

The multiobjective genetic algorithm can be used to optimize two conflicting objectives, oil production and polymer utility factor in polymer flood design. This approach provides a set of optimal solutions which can be considered as trade-off curve (Pareto front) to maximize oil production while preserving polymer performance. Then an optimal polymer flood design can be considered from post-optimization analysis. A 2D synthetic example, and a 3D field-scale application, accounting for geologic uncertainty, showed that beyond the optimal design, a relatively minor increase in oil production requires much more polymer injection and the polymer utility factor increases substantially.

## **DEDICATION**

To my parents for their love, care and support

## **ACKNOWLEDGEMENTS**

I would like to thank my academic advisor, Dr. Akhil Datta-Gupta for his valuable guidance throughout the course of this research. Also I want to thank my committee members, Dr. King and Dr. Mohanty for his valuable feedback and questions that have shaped the work in this research.

I would like to thanks Fulbright for financial support and this valuable experience to be part of the family, PTTEP for their support and working experience which is very useful for my study.

Special thanks to my colleagues at MCERI group; Satyajit for his inspiring ideas, Park for the foundation of the code, Shusei, Shingo, Yusuke, Feyi, Jichao, Changdong, Yanbin, Jiang Xie, Zheng, Xiaoyang, Jixiang, Nega, Aditya, Muhammed, Dongjae, Kim, Suksang for their assistance and for making my graduate years memorable.

## TABLE OF CONTENTS

	Page
CHAPTER I INTRODUCTION AND STUDY OBJECTIVES.....	1
1.1 Introduction .....	1
1.2 Objectives .....	5
1.3 Software Prototype .....	6
CHAPTER II MATHEMATICAL FORMULATION .....	7
2.1 Streamline-Based Rate Optimization .....	7
2.2 Genetic Algorithm .....	9
2.2.1 Initialization.....	10
2.2.2 Multiobjective Fitness Evaluation .....	11
2.2.3 Reproduction and Genetic Operator .....	18
2.2.4 Selection Process .....	21
2.2.5 Demonstration of the Workflow .....	22
CHAPTER III POLYMER FLOOD PERFORMANCE MEASURE OPTIMIZATION	29
3.1 Model and Objective Formulation.....	29
3.1.1 Polymer Properties .....	29
3.1.2 Performance Measures (Variable, Objective).....	33
3.1.3 Two-Stage Optimization Workflow .....	35
3.2 Application .....	36
3.2.1 Five-Spot Example .....	36
3.2.2 Field-Scale Polymer Flood Optimization .....	44
3.2.3 Geologic Uncertainty.....	52
CHAPTER IV CONCLUSIONS AND RECOMMENDATIONS .....	56
4.1 Conclusions .....	56
4.2 Recommendations .....	57

NOMENCLATURE.....	58
REFERENCES.....	60
APPENDIX A.....	63
APPENDIX B.....	65
APPENDIX C.....	72

## LIST OF FIGURES

	Page
Fig. 2.1—Overview of the GA workflow: initialization, population generation, and selection to keep the fittest populations. The optimal solution is obtained once the stopping criteria are reached.....	10
Fig. 2.2—Two-variable sampling with 10 initial points by (left) LHS design, (right) random sample. The result from LHS covers all grids of each variable while random samples left some regions unfilled. ....	11
Fig. 2.3—Optimal solutions from different weights indicated by stars: the weighting factor has strong influence on the optimal result. Different weighting factors result in a changing optimal solution. ....	13
Fig. 2.4—Pareto optimal solutions (Pareto front) with two objectives: points A and B are Pareto optimal solutions, but point C is not a Pareto optimal solution.....	14
Fig. 2.5—Example of domination concept: (left) “A” dominates “B” since both objective values of “A” are less than “B.” (right) “A” does not dominate “B” since objective $f_2$ of “A” is more than “B.” .....	16
Fig. 2.6—Example ranking result from NSGA-II algorithm: Rank 1 populations are nondominated solutions. Rank 2 populations become nondominated when all rank 1 populations are removed.....	17
Fig. 2.7—Tournament selection process: two populations are compared and only the best one is selected to the parent pool.....	18
Fig. 2.8—Summary of reproduction and selection steps: populations from the previous generation are used to produce offspring. The intermediate pool collects all previous populations and offspring and selects them based on their fitness for the next generation. ....	22

Fig. 2.9—Two variables sampling with LHS: variable space ( $x_1, x_2$ ) (left), objective space ( $f_1, f_2$ ) (right). Initial populations are widely spread throughout both variable and objective spaces.....	23
Fig. 2.10—Performance measures of NSGA-II algorithm. (Left) rank average reduces until all becomes 1. (Middle) Variance of crowding distance reduces indicate population becomes more uniform. (Right) Normalized hyper volume tracks movement of populations to Pareto front. ....	24
Fig. 2.11—Objective space of three generations: (Left) Initial generation has widely spread populations. (Middle) Generation#6, all populations become rank#1 and a clear trend of Pareto front is acquired. (Right) Generation#20, populations become more uniformly distributed with the same trend of Pareto front. ....	25
Fig. 2.12—(Left) Relation between square error and $f_1$ , the smallest error point is selected as a trade-off optimal point. (Right) Illustration of the trade-off optimal (red point) and single objective optimization (blue and green points). The trade-off optimal represents a compromised solution between the two objectives.....	26
Fig. 2.13—Histogram of optimal variables, $x_1$ and $x_2$ . The variables are limited in a range of 0 to 5.....	27
Fig. 2.14—Two dimensional distribution plots in variable space. (Left) Scatter plot of initial population (gray) and optimal populations (black) with three optimal point. (Right) Two dimensional histogram. Both of the plots show optimal populations are limited in a range of 0 to 5 with a clear trend. ....	28
Fig. 3.1—Viscosity multiplier, $M(C_p)$ at various $C_p$ : higher polymer concentration results in a more viscous fluid .....	31
Fig. 3.2—Polymer adsorption at various concentrations for HPAM polymer: rock absorbs more polymer at higher concentration.....	32
Fig. 3.3—Sensitivity analysis of $\Delta Np$ and UF with increasing polymer concentration: both objectives increase with increasing concentration.....	34



Fig. 3.4—Two-stage coupling optimization workflow: Streamline and NSGA-II.....	35
Fig. 3.5—2D five-spot example: permeability distribution and well location.....	36
Fig. 3.6—Production rate allocation to each producer with streamline-based method ...	37
Fig. 3.7—Oil saturation map of the two cases: I) Base case with equal production rates and II) Optimized case by the streamlines method: optimized case delays water breakthrough results in less oil saturation left in the reservoir...	38
Fig 3.8—NSGA-II algorithm generates a trade-off relation between UF and $\Delta Np$ (2D five-spot model): gray dots are nonoptimal solutions, black dots are Pareto optimal solutions.....	39
Fig. 3.9—Three optimal points with different focuses (2D five-spot model): (left) minimizing UF, (right) maximizing $\Delta Np$ , and (middle) the trade-off optimal point represents a compromised solution between efficiency and production. ....	40
Fig. 3.10—Cumulative oil production, cumulative polymer injection, and UF from the three optimal points (2D five-spot model) .....	41
Fig. 3.11—2D map from three optimal points (left), oil Saturation maps (right), polymer concentration maps. Increasing polymer beyond the optimal design results in much more trapped polymer with small improvements in remaining oil-in-place. ....	42
Fig. 3.12—Histogram of optimal variables, polymer concentration and slug size. Most of the populations have high concentrations with intermediate slug size. ....	43
Fig. 3.13—Two dimensional distribution plots in variable space. Higher polymer concentration tends to have higher slug size. The trade-off optimal solution is shown in the middle among all the solutions.....	43
Fig. 3.14—Goldsmith GSAU field: complexity of the field makes it appropriate for large scale field demonstration.....	44

Fig. 3.15—Field production/injection history. Maximum production/injection rate: 4000 RB/D (first 8 years), 8000 RB/D (next 13 years).....	45
Fig. 3.16—NSGA-II algorithm generates a trade-off relation between UF and $\Delta Np$ (field-scale model): gray dots are nonoptimal solutions, black dots are Pareto optimal solutions. ....	46
Fig. 3.17—Three optimal points with different focuses (field-scale model): (left) minimizing UF, (right) maximizing $\Delta Np$ , and (middle) the trade-off optimal point represents the compromised solution between efficiency and production. ....	47
Fig. 3.18—Cumulative oil production, cumulative polymer injection, and UF from three different optimal points (field-scale model).....	48
Fig. 3.19—2D map from three optimal points: (left) oil Saturation maps, (right) polymer concentration maps: injecting polymer beyond the optimal design gives small change in remaining oil saturation, but the amount of polymer trapped in the reservoir increases drastically. ....	49
Fig. 3.20—Histogram of optimal variables, polymer concentration and slug size. No population has polymer concentration exceed 1,300 ppm. ....	50
Fig. 3.21—Two dimensional distribution plots in variable space. Higher polymer concentration tends to have higher slug size. The trade-off optimal solution is in the middle among all the solutions. ....	50
Fig. 3.22—NPV result of the two scenarios, high and low oil price. The highest NPV point from the high oil price scenario is not an optimal solution in the low oil price scenario. The optimal trade-off point provides a good NPV result in both cases. ....	52
Fig. 3.23—30 porosity realizations from SGS with high porosity channel in NW-SE direction.....	53
Fig. 3.24—Workflow to compare optimization under geologic uncertainty with blind realizations. The expected Pareto front from five realizations is compared with multiple blinds realizations. ....	54

Fig. 3.25—Pareto fronts calculated from the average value and multiple individual blind realizations: The average Pareto front is a good representative to capture geologic uncertainty. ....55

## LIST OF TABLES

	Page
Table 3.1—Five-spot model production constraints .....	37
Table 3.2—Control variables for three optimal points (2D five-spot case).....	40
Table 3.3—Field-scale model production constraints.....	45
Table 3.4—Control variables for three optimal points (field-scale model) .....	47
Table 3.5—Economic assumptions.....	51

## CHAPTER I

### INTRODUCTION AND STUDY OBJECTIVES

#### 1.1 Introduction

Waterflooding is the most widely used oil recovery method after primary depletion. The popularity of the method hinges on the economic feasibility and ease of implementation. However, the presence of heterogeneities and adverse mobility contrasts leads to viscous fingering and poor sweep efficiency, resulting in reduced oil recovery. Mobility control by polymer flood is one of the attractive methods to overcome this problem to increase oil recovery (Lake, 1989).

The successful implementation of a polymer flood requires thorough understanding of influences of design parameters on polymer flood performance. The optimal slug size and concentration are controlled by economic limit, but are generally high. In terms of polymer flood initiation time, early implementation is an important criterion for success (Al-Sofi and Blunt, 2011)

Sensitivity analysis is the conventional approach to analyze the impact of control variables on objectives. Pope et al. (1979) used a compositional simulator to analyze the effect of chemical properties and control variables on oil recovery. Each parameter was analyzed separately by the sensitivity analysis framework. DeHekker et al. (1986) designed polymer control parameters in Byron and North Oregon basin field by using laboratory results from radial core flooding. The optimal polymer concentration and slug size were determined from sensitivity plot where oil recovery was found to increase with

more injected polymer. Barua et al. (1986) proposed an economic assessment framework for EOR projects using analytical models. Sensitivity analyses of chemical slug size and pattern area were analyzed by spider plot around the optimal design. Anderson et al. (2006) used a multiphase multicomponent chemical flood simulator (UTCHEM) to perform sensitivity studies of chemical control parameters. Apart from NPV, chemical efficiency was analyzed to be an indicator of chemical usage performance. Abedi and Algharaib (2012) performed polymer flood reservoir simulation study to analyze the effects of control parameters: polymer concentration, polymer slug size and salt concentration, on oil recovery. Even though, the sensitivity analysis is common for various studies, this method only provides a relation between single control variable to a single objective.

Over the past few years, many authors have proposed approaches to optimize profit and polymer efficiency. An adjoint method can maximize the net present value (NPV) of polymer flood simulations by computing gradients with respect to design parameters from a single simulation run (Doren et al., 2011). Streamline simulation has shown that the polymer utility factor (UF) gives a good indication of polymer efficiency (Clemens et al., 2011). The UF is a mass of polymer injected per volume of oil produced. Streamline simulation can also be used for rate optimization (Sharma et al., 2011), which focuses on arrival time equalization (Datta-Gupta and King, 2007). A major advantage of this approach is the analytical computation of sensitivities which makes it computationally efficient and suitable for large field cases.

For the viability of polymer flood projects, the two critical measures, production gain and polymer efficiency, should be considered simultaneously. To optimize multiple objectives, an optimization approach that has drawn much attention recently is multiobjective optimization (MOO) (Deb et al., 2002; Srinivas and Deb, 1994). This method evaluates fitness by the dominance relationship instead of fitness measures as in the ordinary single-objective optimization approaches. The algorithm generates a set of optimal solutions which represent the appropriate compromise between multiple objectives which is called the Pareto optimal solutions (or Pareto front).

Many authors have demonstrated benefits of multiobjective optimization over the conventional single objective in various applications. For history matching, the multiobjective approach provides improved overall match of the data than the single objective as it avoids arbitrary weighting factors between matching variables (Ferraro and Verga, 2009; Sayyafzadeh et al., 2012). The approach preserves variation among the population and provides more reliable uncertainty quantification than the single-objective approach (Hajizadeh et al., 2011; Han et al., 2010; Mohamed et al., 2011; Park et al., 2013; Schulze-Riegert et al., 2007). For example, the trapping efficiency of CO<sub>2</sub> storage in the saline aquifer can be optimized using the concept of Pareto optimality (Nghiem et al., 2009). The concept offers a detailed insight into the effect of operating conditions on residual gas and solubility trapping. The reduced-order model for reservoir simulation has also been used to generate Pareto front between maximizing cumulative oil production and minimizing cumulative water injection (Cardoso, 2009). The result shows the trade-off between cumulative oil production and cumulative water injection.

To address geologic uncertainty, optimization is performed with multiple realizations. An adjoint method can maximize the expected value of NPV from multiple realizations by optimizing production/injection rates (Van Essen et al., 2009). The genetic algorithm (GA) is also used to search the optimal location with fitness evaluated from multiple geologic models (Morales et al., 2011). The multiple realization approach has also been used for well-path design in gas-condensate field (Schulze-riegert et al., 2011). Alhuthali et al. (2008) used streamlines to optimize production/injection rate. The geological uncertainty is considered from the stochastic-optimization framework based on the combination of the expected value and variance of a performance measure from multiple realizations.

This study provides an approach to design and optimize the polymer flood considering both production improvement and polymer efficiency. We coupled the streamline-based rate optimization from previous work (Sharma et al., 2011) with one class of MOO, nondominated sorting genetic algorithm (NSGA), to optimize polymer concentration and slug size. This approach provides a comprehensive understanding between maximizing oil production and preserving polymer efficiency presented as a compromised trade-off curve. The trade-off curve generated from multiple realizations is used to account for geologic uncertainty. The post-optimization decision can be made from the trade-off curve to select the optimal design.



## 1.2 Objectives

Our goal here is to provide an approach to design and optimize polymer flood that considers both production improvement and polymer efficiency. We coupled the streamline-based rate optimization (Sharma et al., 2011) with the multiobjective genetic algorithm to optimize polymer concentration and slug size. This approach provides a comprehensive understanding between maximizing oil production and preserving polymer efficiency presenting as a compromised trade-off curve. Geologic uncertainty is accounted by generating average trade-off curve from multiple realizations. Post-optimization decision can be made from the trade-off curve to the select optimal design. In brief, the objectives of this study are:

- We will extend the previous study from Sharma et al. (2011) to incorporate more influencing parameters as polymer concentration and slug size to polymer flood optimization.
- We will use a multiobjective genetic algorithm to generate a set of optimal solutions representing a trade-off between production improvement and polymer performance.
- We will demonstrate applications of this approach through a five-spot synthetic case, and a 3D field-scale case accounting for geologic uncertainty.

### **1.3 Software Prototype**

The primary deliverable of this work will be an optimization software implementing a derivative free approach with various algorithms including design of experiment, self-adaptive proxy, and genetic algorithm. The software is designed to work for both single objective and multiobjective problems. Additionally, the software works in parallel environment which significantly increase calculation efficiency of the genetic algorithm.

The software is developed in MATLAB with parallel computing toolbox. It is designed to be user friendly and allows for easy enhancement. The applications presented in this study have been carried out using this software.

## CHAPTER II

### MATHEMATICAL FORMULATION

In this section, we discuss the mathematical formulation of two important approaches from the core of this work that are streamline-based rate optimization and multiobjective optimization including the genetic algorithm.

#### 2.1 Streamline-Based Rate Optimization

The goal of streamline-based rate optimization is to allocate production and injection rates for maximizing sweep efficiency. The formulation of this approach is based on equalizing the arrival times of injected fluid. The objective function consists of two terms (Taware et al., 2010).

$$p(q) = \sum_{m=1}^{N_{\text{group}}} \sum_{i=1}^{N_{\text{prod},m}} [t_{d,m}(\mathbf{q}) - t_{i,m}(\mathbf{q})]^2 + \eta \sum_{m=1}^{N_{\text{group}}} \sum_{i=1}^{N_{\text{prod},m}} [t_{i,m}(\mathbf{q})]^2 \dots\dots\dots(2.1)$$

The sweep efficiency optimization can be achieved by minimizing the first term in **Eq. 2.1** which denotes the misfit between the calculated waterfront arrival time at each producer and the average arrival time of producers in a specified group. Also,  $t_{i,m}$  represents the calculated arrival time of well  $i$ , belonging to group  $m$ , and  $t_{d,m}$  represents average arrival time for the wells in group  $m$ . Minimizing the second term denotes reducing the magnitude of arrival times, leading to production acceleration. The trade-off between equalizing arrival time and production acceleration can be controlled by the

weight  $\eta$ . The arrival time is a function of vector  $\mathbf{q}$ , which contains production/injection rates and has a dimension equal to the number of well rates to be optimized.

The term  $t$  is defined as travel time of the waterfront from the injector to each producer. For calculation purposes, we compute the arrival time to a producer as the average of the time of flight of 20% of the fastest streamlines denoted by  $N_{fsl,i}$ . Time of flight of streamline  $l$  of producer  $i$  is represented as  $\tau_{l,i}$ . Thus, the arrival time to producer  $i$  will be computed as,

$$t_{i,m}(\mathbf{q}) = \frac{1}{N_{fsl,i}} \sum_{l=1}^{N_{fsl,i}} \tau_{l,i}(\mathbf{q}) \quad \dots\dots\dots(2.2)$$

The variable  $\tau_{l,i}$  is defined as (Datta-Gupta and King, 2007)

$$\tau_{l,i} = \int_{\Sigma} s(x) dx \quad \dots\dots\dots(2.3)$$

where the integral is along the streamline trajectory,  $\int_{\Sigma} s(x) dx$  is the “slowness” defined as the reciprocal of the total interstitial velocity.

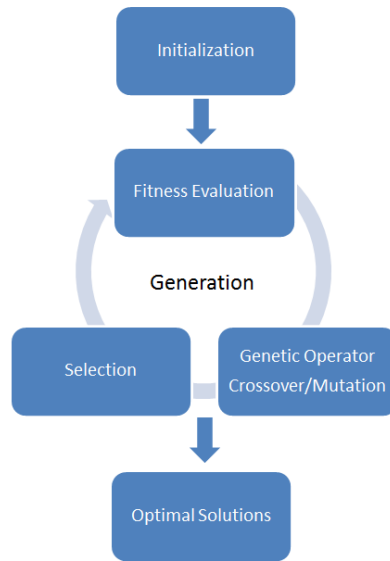
$$s(x) = \frac{1}{|v(x)|} = \frac{A(x)\phi(x)}{q_{sl}} \quad \dots\dots\dots(2.4)$$

The average arrival time for group  $m$  ( $t_{d,m}$ ) is an arithmetic average of the arrival times of the wells in the group

$$t_{d,m}(\mathbf{q}) = \frac{1}{N_{prod,m}} \sum_{i=1}^{N_{prod,m}} t_{i,m}(\mathbf{q}) \quad \dots\dots\dots(2.5)$$

## 2.2 Genetic Algorithm

The genetic algorithm (GA) is an adaptive heuristic optimization method that has been applied for many petroleum engineering problems. The idea of the GA comes from self-adapting behaviors of living organisms. In other words, GA tries to replicate the concepts of natural evolution by mathematical concepts. The algorithm does not require estimation of gradients, but typically requires multiple forward simulations to evaluate the fitness of populations. The control variables are codified to form chromosomes as representative of the characteristics of populations. The fittest populations will be selected to transfer part of their chromosomes to reproduce offspring by the genetic operator: crossover and mutation. This process will be done iteratively and the population fitness will be improved over generations. The advantage of this method is the ability to search for the global optimal while the gradient-based optimization usually converges to a local solution. The following details of this section are a brief review of each step of the GA and the concept of the multiobjective GA. An overview of the GA workflow is summarized in **Fig. 2.1**.



**Fig. 2.1—Overview of the GA workflow: initialization, population generation, and selection to keep the fittest populations. The optimal solution is obtained once the stopping criteria are reached.**

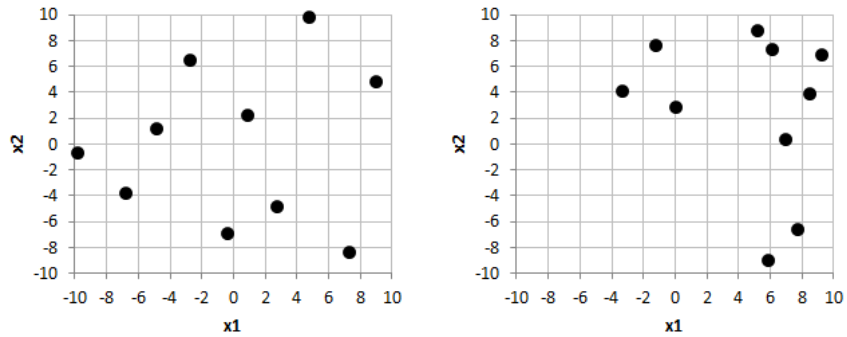
### **2.2.1 Initialization**

Initialization is the first step of the GA. Initial chromosomes are generated from a given range of control variables. Latin hypercube sampling (LHS) was implemented in this work to provide an efficient variable-sampling method from the given range. Unlike simple random sampling, this method ensures a full coverage of the range of each variable by maximally stratifying each marginal distribution (Yin et al., 2010). The steps for LHS sampling are as follows:

- 1) Divide search in each variable into “N” intervals where “N” is the number of sampling points.
- 2) Uniformly sample from each interval.

3) Randomly pair sampled variable values.

To illustrate the process, 10 sample points from LHS design and the random process of two-variable space with are compared in **Fig. 2.2**. LHS design ensures the coverage of the initial population in the given range, but random samples might bias toward the local region and leave some regions unfilled. The locally distributed initial population will limit the GA search in later generation leading to a suboptimal solution.



**Fig. 2.2—Two-variable sampling with 10 initial points by (left) LHS design, (right) random sample. The result from LHS covers all grids of each variable while random samples left some regions unfilled.**

### 2.2.2 Multiobjective Fitness Evaluation

After the initial population is generated, the iterative evolution process begins. The objective function is used to evaluate each proposed population by how well it provides a good solution to the problem. For reservoir study, most problems require more than one objective to evaluate fitness: history matching comprises multiple production misfits, such as, water cut, bottomhole pressure, and GOR; optimization has to consider

production improvement and operational efficiency simultaneously. Two approaches can be used to define fitness for the multiobjective problems: scalarization and Pareto-based (Park et al., 2013).

*Scalarization (Weighted-Sum) Method*

The scalarization method simplifies multiple objectives to the single objective by summing them using the weighting factor ( $w_i$ ).

$$f = w_1f_1 + w_2f_2 + \dots + w_nf_n \dots\dots\dots(2.7)$$

However, appropriate weights for each objective are difficult to be determined, especially when they are in different scales, such as pressure misfit (psi) or total production (STB) (Yin et al., 2010). We do not know which weights are the most appropriate to retrieve a satisfactory solution (Hajizadeh et al., 2011). To demonstrate the importance of the weighting factor, **Eq. 2.8** shows the multiobjective problem with two objectives,  $f_1, f_2$ , and two variables,  $x_1, x_2$ .

$$\begin{aligned} \min \underline{f}(\underline{x}) \\ f_1(\underline{x}) = x_1^2 + x_2^2 \dots\dots\dots(2.8) \\ f_2(\underline{x}) = (x_1 - 5)^2 + (x_2 - 5)^2 \end{aligned}$$

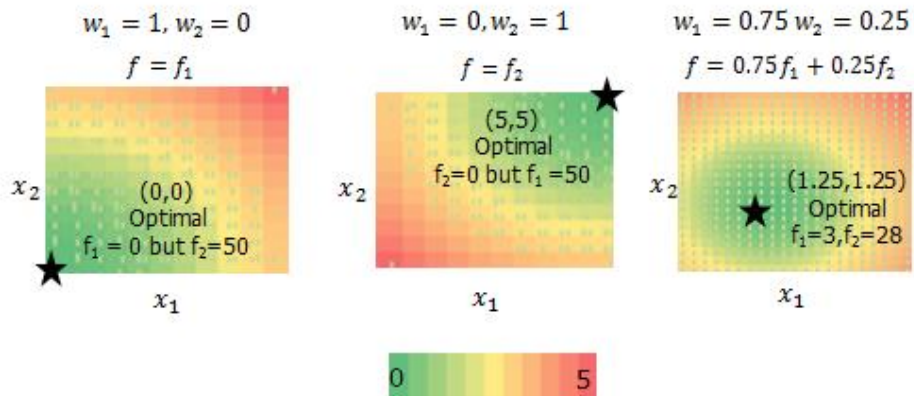
with the weighted sum approach, the minimization objective of this problem is written as

$$\min f(\underline{x}) = w_1f_1(\underline{x}) + w_2f_2(\underline{x}) \dots\dots\dots(2.9)$$

The importance of weighting factors is shown in **Fig. 2.3**. With different weights, the optimal (minimal) solutions change. This issue becomes more critical in petroleum



engineering problems as they always have more dimensions of objectives and variables with different units.

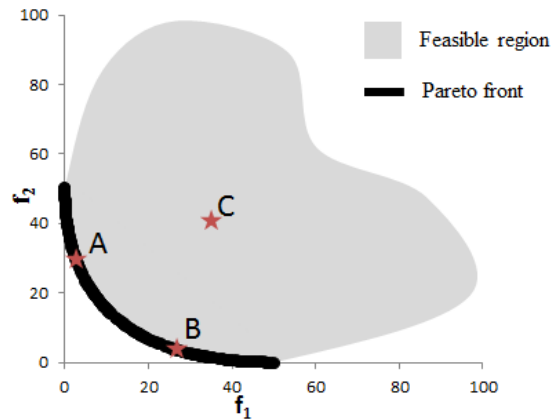


**Fig. 2.3—Optimal solutions from different weights indicated by stars: the weighting factor has strong influence on the optimal result. Different weighting factors result in a changing optimal solution.**

### *Multiobjective (Pareto-Based) Method*

#### Pareto Optimality

Instead of searching for the single optimal solution as in the conventional weighted sum method, the purpose of the multiobjective approach is to explore a set of optimal solutions that represents the compromised trade-off between multiple conflicting objectives. Within the optimal solutions, no objectives in the solution can be made better without making another objective worse. This concept is called Pareto optimality (or a Pareto front).



**Fig. 2.4—Pareto optimal solutions (Pareto front) with two objectives: points A and B are Pareto optimal solutions, but point C is not a Pareto optimal solution.**

The example of Pareto optimality is illustrated in **Fig. 2.4**. In this example, we define two objectives,  $f_1$  and  $f_2$ , for the minimization problem from the previous example (**Eq. 2.8**). The gray area represents the feasible solutions space for this problem. Let us consider point C. Now, we can find some solutions in the space that are superior to it in both objectives (for example, point A has both  $f_1$  and  $f_2$  lower than point C). In this situation, point C is not a Pareto optimal solution. At point A, we cannot find any solutions in the feasible region that are superior to it in both objectives. If we want to improve  $f_2$  from point A (minimize  $f_2$ ), we may consider moving it to point B. However, this move deteriorates  $f_1$  since  $f_1$  gets higher value. This behavior is the same for all points in the black line. We call the solutions in the black line Pareto optimal solutions (or Pareto a front) for this problem.

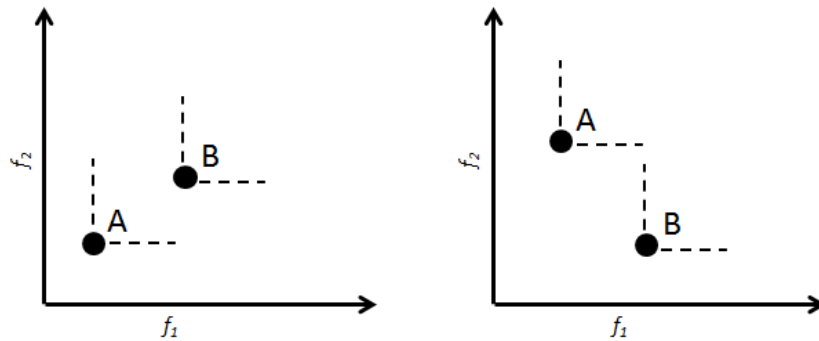
## NSGA-II Algorithm

The Pareto front represents a set of optimal solutions for the multiobjective problem. However, practically, we cannot search the whole solution space and define the true Pareto front. Deb et al. (2002) introduced the NSGA-II algorithm, which uses the genetic algorithm framework to improve the optimality over multiple generations. The algorithm evaluates fitness by the dominant relationship instead of fitness measures as in the ordinary single-objective genetic algorithm. The domination concept defines levels of optimality by assigning several ranks to each population. Over generations, the NSGA-II algorithm minimizes ranks until all the members of the population become rank 1, which implies the Pareto optimality condition is reached.

Mathematically, to illustrate the dominance concept, “A” is said to dominate “B” if the two points satisfy the following condition:

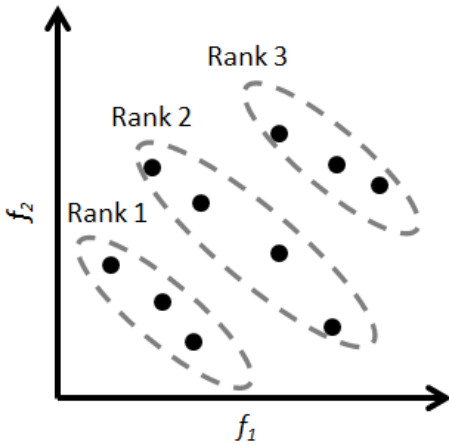
$$\forall i \in \{1, 2, \dots, n\}: f_i(a) \leq f_i(b) \wedge \exists j \in \{1, 2, \dots, n\}: f_j(a) < f_j(b) \quad \dots\dots\dots(2.10)$$

This means “A” dominates “B” when all objectives from “A” are not more than “B” and at least one objective from “A” is less than “B.” The example of the dominance concept is shown in **Fig. 2.5**. The figure on the left illustrates a situation when “A” dominates “B” since both objective values of “A” are less than “B.” The figure on the right illustrates a situation when “A” does not dominate “B” since one of the objectives,  $f_2$ , of “A” is more than “B.”



**Fig. 2.5**—Example of domination concept: (left) “A” dominates “B” since both objective values of “A” are less than “B.” (right) “A” does not dominate “B” since objective  $f_2$  of “A” is more than “B.”

The NSGA-II algorithm defines levels of optimality from the nondominated solution as rank “1”. Next, the algorithm removes all the rank “1” points out of the space and defines the nondominated solution from the remaining points as rank “2.” This process is repeated and rank “N+1” is defined until the last population. **Fig. 2.6** shows the ranking example of 10 populations from the NSGA-II algorithm.



**Fig. 2.6—Example ranking result from NSGA-II algorithm: Rank 1 populations are nondominated solutions. Rank 2 populations become nondominated when all rank 1 populations are removed.**

The purpose of the ranking is to obtain a set of solutions as close as possible to the true Pareto front. In addition to this main purpose, the NSGA-II algorithm keeps the diversity of the solutions simultaneously by imposing a secondary objective called *crowding distance* (Srinivas and Deb, 1994). The crowding distance is a measure of how close an individual is to its neighbors (**Eq. 2.11**). The large crowding distance results in better diversity.

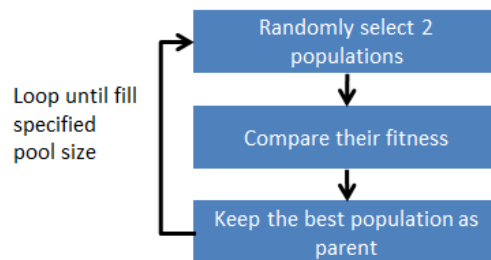
$$\text{Crowding Distance} = \sum_{i=1}^{N_{\text{objective}}} \left( \frac{f_{i_{\text{next}}} - f_{i_{\text{previous}}}}{f_{i_{\text{max}}} - f_{i_{\text{min}}}} \right) \dots\dots\dots(2.11)$$

At this point, all populations will be sorted by rank and crowding distance. First, they are sorted by rank. The populations with the same rank will be sorted by the crowding distance. The best population has low rank and high crowding distance.

### 2.2.3 Reproduction and Genetic Operator

#### *Parent Selection by Tournament Pool*

Once the populations are sorted, some of them are selected to produce offspring. This selection is done by tournament pool **Fig. 2.7**. First, two members are selected with uniform probability distribution. Next, their fitnesses are compared and only the best one is selected to be kept in the parent pool. The operation is repeated until the parent pool is filled with the specified number. In this work, the size of the parent pool is half of the size of the population pool.



**Fig. 2.7—Tournament selection process: two populations are compared and only the best one is selected to the parent pool.**

#### *Offspring Reproduction*

The selected parents will transfer part of their chromosomes (genomes) to reproduce offspring. Conventionally, the chromosomes are represented by binary codes (Nasrabadi et al., 2012). However, the coding of our variables is in continuous space which has difficulty with the conventional binary coding concept. In this work we used simulated

binary operator which has similar behavior to binary-coded GA , but works with continuous space (Deb and Agrawal, 1995; Deb et al., 2002).

*Simulated Binary Crossover*

Crossover is the key process of the genetic algorithm. It replicates the behavior of the living organism to recombine genomes from two parents to produce offspring. The assumption behind this operation is that recombination of fit parents will produce better offspring. The operation starts from a random selection of a number  $u_k$  between 0 and 1; then calculation of the weighting factor,  $\beta_k$ (**Eq. 2.12** and **Eq. 2.13**).

$$\beta_k(u_k) = (2u_k)^{\frac{1}{\eta+1}}, \text{ if } 0 < u_k \leq 0.5 \quad \dots\dots\dots(2.12)$$

$$\beta_k(u_k) = \frac{1}{[2(1 - u_k)]^{\frac{1}{\eta+1}}}, \text{ if } 0.5 < u_k < 1 \quad \dots\dots\dots(2.13)$$

where  $\eta$  is a distribution index for crossover.  $\beta_k$  is a weighting factor for  $k^{\text{th}}$  genome. Then this weighting factor,  $\beta_k$ , will be used to produce two offspring genomes (**Eq. 2.14** and **Eq. 2.15**).

$$c_{1,k} = \frac{1}{2}[(1 - \beta_k)p_{1,k} + (1 + \beta_k)p_{2,k}] \quad \dots\dots\dots(2.14)$$

$$c_{2,k} = \frac{1}{2}[(1 + \beta_k)p_{1,k} + (1 - \beta_k)p_{2,k}] \quad \dots\dots\dots(2.15)$$

where  $c_{i,k}$  is the  $i^{\text{th}}$  child with  $k^{\text{th}}$  genome .  $p_{1,k}$  and  $p_{2,k}$  are selected parents with  $k^{\text{th}}$  genome.

*Polynomial Mutation*

The mutation operator is used to randomly alter values of some genomes from a single parent. This process is similar to “asexual” influence of a single population. Mutation is a vital function to search for global optimal solutions since it increases diversity of the population pool. With the crossover only, the solution will converge to a suboptimal solution close to the fittest population found in each generation. The rate of crossover/mutation can be controlled by crossover probability. In most of the cases, the crossover probability is around 90%. For highly nonlinear problems, the crossover operator becomes less efficient, and the probability should be altered to give more focus on the mutation process.

Similar to the crossover operator, this polynomial mutation works for the continuous variable space. The operation starts from a random selection of a number  $r_k$  between 0 and 1; then calculation of a weighting factor,  $\delta_k$  (**Eq. 2.16** and **Eq. 2.17**).

$$\delta_k = (2r_k)^{\frac{1}{\eta_m+1}} - 1, \text{ if } 0 < r_k < 0.5 \quad \dots\dots\dots(2.16)$$

$$\delta_k = 1 - [2(1 - r_k)]^{\frac{1}{\eta_m+1}}, \text{ if } 0.5 \leq r_k < 1.0 \quad \dots\dots\dots(2.17)$$

where  $\eta_m$  is a distribution index for mutation.  $\delta_k$  the a weighting factor for  $k^{\text{th}}$  genome.

Next, this weighting factor,  $\delta_k$ , will be used to update a genome in a selected parent in **Eq.2.18**.

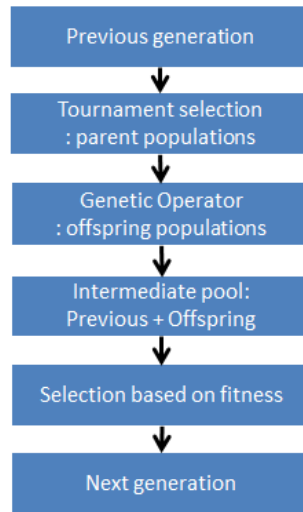
$$c_k = p_k + (p_k^u - p_k^l)\delta_k \quad \dots\dots\dots(2.18)$$



where  $c_k$  is the child at  $k^{\text{th}}$  genome and  $p_k$  is the selected parent.  $p_k^u$  and  $p_k^l$  are upper and lower bounds for  $k^{\text{th}}$  genome.

#### **2.2.4 Selection Process**

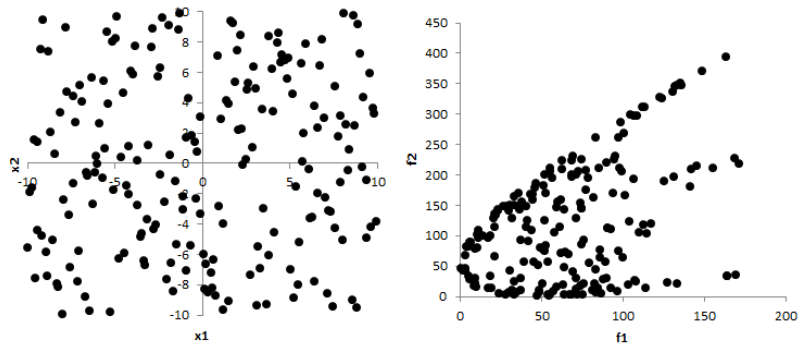
After offspring population is generated, their fitness will be evaluated. Next, they will be combined with all populations from the previous generation in the intermediate pool. For example, if we have  $N$  populations from the previous generation and the genetic operator generates  $N$  offspring, now the intermediate pool has  $2N$  population size. Then, all populations in this pool will be sorted (single objective: by weighted objective, multiobjective: by rank and crowding distance). The only fittest  $N$  populations will be kept for the next generation. The following flowchart (**Fig.2.8**) summarizes the workflow of the reproduction and selection process.



**Fig. 2.8—Summary of reproduction and selection steps: populations from the previous generation are used to produce offspring. The intermediate pool collects all previous populations and offspring and selects them based on their fitness for the next generation.**

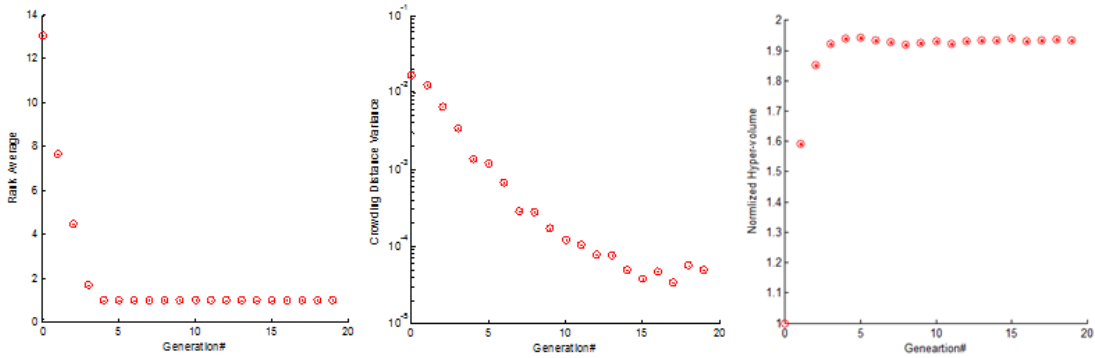
### **2.2.5 Demonstration of the Workflow**

The multiobjective problem from **Eq.2.8** was re-analyzed using the multiobjective genetic algorithm framework. Initial populations were generated using LHS Parameters in a range of  $[-10,10]$  with 200 samples (**Fig. 2.9**). The LHS ensures diversity throughout the variable space resulting in wide spread in the objective space.



**Fig. 2.9—Two variables sampling with LHS: variable space ( $x_1, x_2$ ) (left), objective space ( $f_1, f_2$ ) (right). Initial populations are widely spread throughout both variable and objective spaces.**

The NSGA-II algorithm was applied to the initial population. The primary objective of the algorithm is to move the population close to the Pareto front by the domination concept. Average rank of all populations is reduced over generations until all populations become rank one (**Fig. 2.10-left**). In addition to rank, crowding distance is considered as a secondary objective in the process to increase diversity among the populations. Variance of crowding distance is considered to monitor improvement of the population distribution (**Fig. 2.10-middle**). The later generations provides a reduction in the variance of crowding distance which implies populations become more uniformly distributed.



**Fig. 2.10—Performance measures of NSGA-II algorithm. (Left) rank average reduces until all becomes 1. (Middle) Variance of crowding distance reduces indicate population becomes more uniform. (Right) Normalized hyper volume tracks movement of populations to Pareto front.**

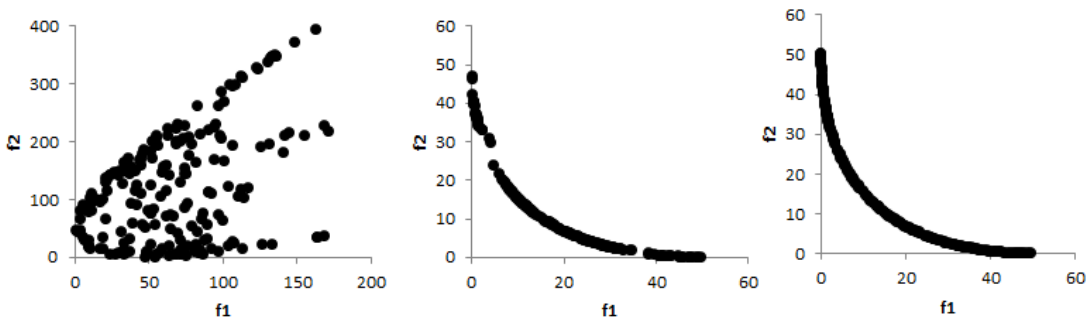
To track the movement of the population to the Pareto front, we applied the hyper volume concept (Li, 2012) to measure the overall moved distance from the initial generation. The hyper volume is a summation of volume of each population to a reference point in objective space (Eq. 2.19). The reference point is assumed to be the worst population (high rank and low crowding distance) from the initial generation.

$$\text{Hyper volume} = \sum_{i=1}^{\text{N\#Population}} (\Delta f_1 * \Delta f_2 * \Delta f_3 \dots \Delta f_{\text{N\#Objective}})_i \dots\dots\dots(2.19)$$

The populations in the later generation are moved away from the reference point to be closer to the Pareto front resulting in higher hyper volume. The hyper volume can be used as a qualitative measure to track the movement by normalizing the value with hyper volume from the initial generation (Eq. 2.20).

$$(\text{Normalized hyper volume})_i = \frac{(\text{Hyper volume})_i}{(\text{Hyper volume})_1} \dots\dots\dots(2.20)$$

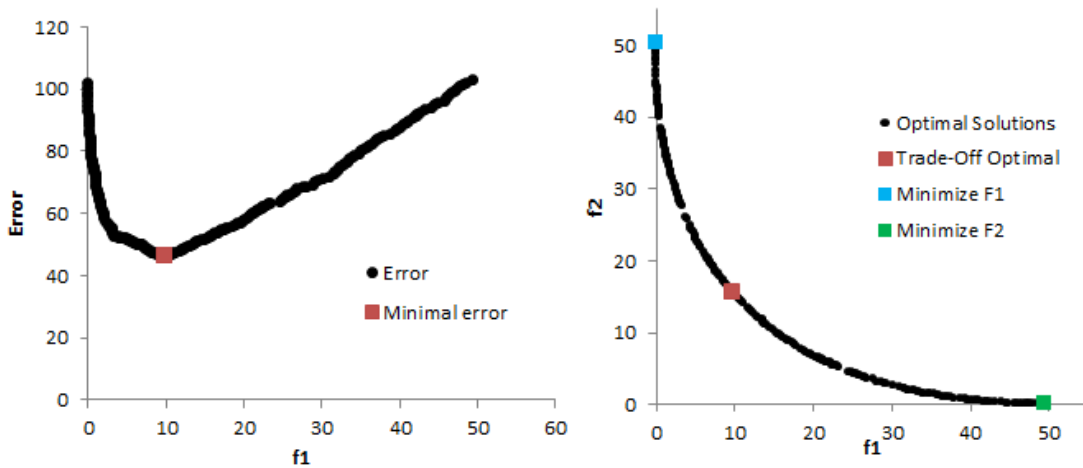
In this analytical example, the NSGA-II algorithm was run for 20 generations to clearly demonstrate the effect of these three performance measures. The improvement is relatively small after all populations have become rank one in generation#6 since the normalized hyper volume (**Fig. 2.10-right**) becomes stable. After this generation, only the variance of crowding distance is improved (**Fig. 2.10-left**). The result from initial generation, generation#6, and generation#20 are shown in **figure 2.11**. A clear trend of Pareto front can be acquired from generation#6 where all populations become rank one. Beyond this point, the populations become more uniform, but the trend of the Pareto front is still the same.



**Fig. 2.11—Objective space of three generations: (Left) Initial generation has widely spread populations. (Middle) Generation#6, all populations become rank#1 and a clear trend of Pareto front is acquired. (Right) Generation#20, populations become more uniformly distributed with the same trend of Pareto front.**

After the Pareto front is reached, post-optimization decision will be performed to select the optimal trade-off solution. The MSE algorithm (Ma, 2008) is used to select the

trade-off point. The idea behind this algorithm is to search the solution with the smallest square error from the other solutions around it which result in a point around the ramp-up of a curve. The trade-off optimal point is selected from the MSE algorithm (**Fig. 2.12-left**). This optimal trade-off represents a compromised solution between the two objectives in the objective space (**Fig. 2.12-right**).

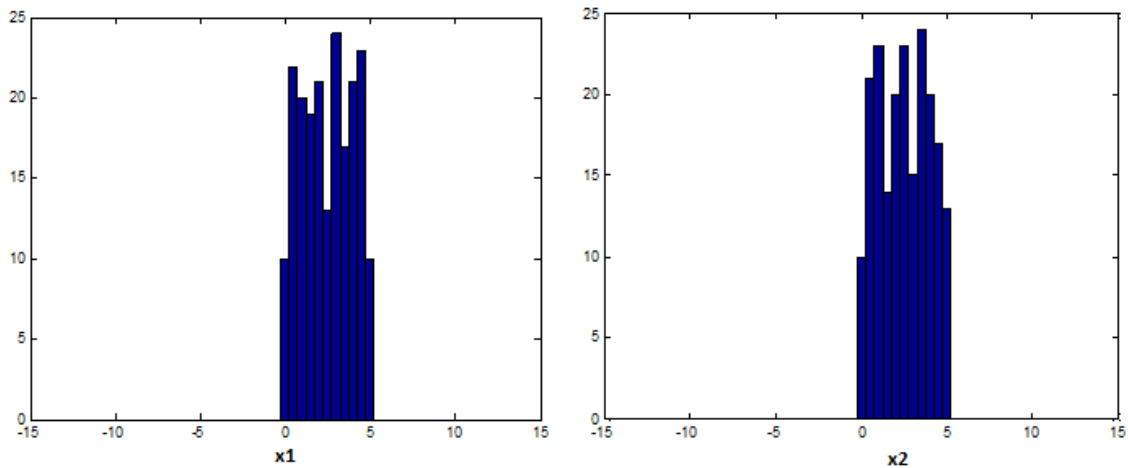


**Fig. 2.12—(Left) Relation between square error and  $f_1$ , the smallest error point is selected as a trade-off optimal point. (Right) Illustration of the trade-off optimal (red point) and single objective optimization (blue and green points). The trade-off optimal represents a compromised solution between the two objectives.**

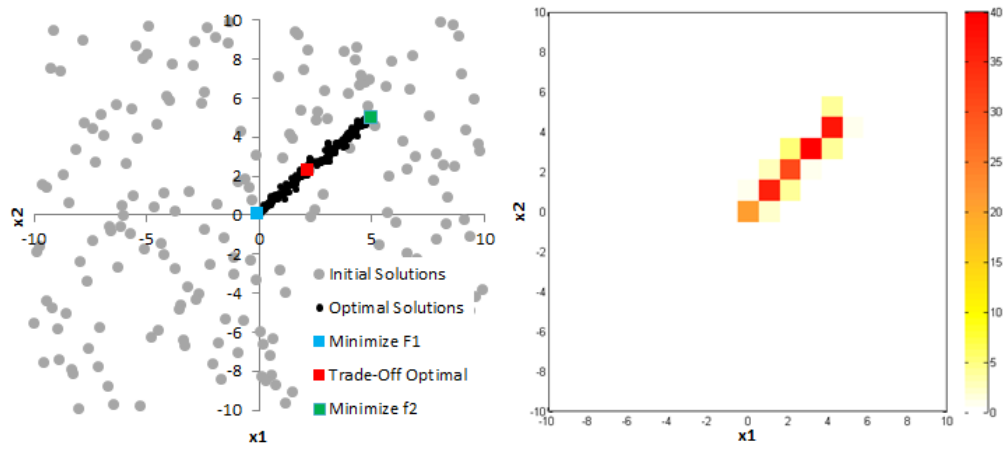
The three optimal solutions from (Fig. 2.12-right) illustrate optimization with different focuses. The blue and the green points represent an optimization with focus to minimize  $f_1$  and  $f_2$  respectively. These solutions only satisfy small value of the focused objective, but provide the unfavorable objective for the other. The point in the middle

represents the trade-off optimal solution. The solution does not provide the smallest value for either  $f_1$  and  $f_2$ , but it represents a compromised solution between them.

In the variable space, the distribution of the variable after optimization is limited in a range of 0 to 5 (**Fig. 2.13**). The two dimensional distribution plots show a clear trend of the optimal population density (**Fig. 2.14**). The optimal solution to minimize  $f_1$  and  $f_2$  are close to the edge of optimal solution (0, 0) and (5, 5) respectively while the trade-off optimal solution provides variable in between them. The combination of the variables in the optimal space provides population in the Pareto front.



**Fig. 2.13—Histogram of optimal variables,  $x_1$  and  $x_2$ . The variables are limited in a range of 0 to 5.**



**Fig. 2.14—Two dimensional distribution plots in variable space. (Left) Scatter plot of initial population (gray) and optimal populations (black) with three optimal point. (Right) Two dimensional histogram. Both of the plots show optimal populations are limited in a range of 0 to 5 with a clear trend.**



## **CHAPTER III**

### **POLYMER FLOOD PERFORMANCE MEASURE OPTIMIZATION**

In this chapter, we propose a new approach for polymer flood optimization that considers both production improvement and polymer performance. This work expands the scope of rate optimization study (Sharma et al., 2011) to consider more control variables for polymer flood: concentration and slug size. The two algorithms from the previous chapter are coupled. First, the streamline-based rate optimization allocates production/Injection rate among all wells which capture effects of reservoir heterogeneity. Next, the multiobjective genetic algorithm (NSGA-II) optimizes polymer concentration and slug size and presents the result as a compromised trade-off curve between production gain and utility factor. The application of this approach will be demonstrated with three cases: a five-spot synthetic model, a 3D field-scale polymer flood, and optimization under geologic uncertainty. Advantages of the analysis under multiobjective framework over the single-objective approach will be discussed at the end of this chapter.

#### **3.1 Model and Objective Formulation**

##### **3.1.1 Polymer Properties**

Recovery improvement by waterflooding often results in viscous fingering from adverse mobility ratio. Mobility control by polymer injection decrease mobility of injected water

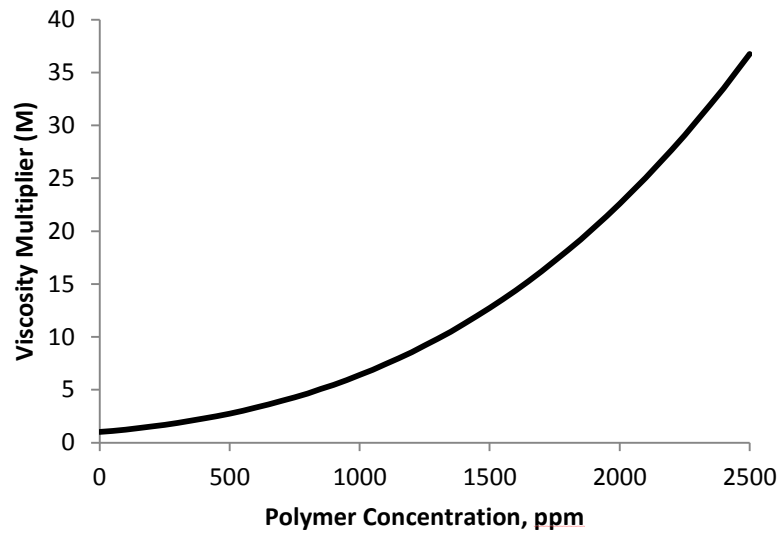
by increase of water viscosity and decrease in rock permeability to water (Lake, 1989). However, the oil permeability remains unaffected. These effects result in more favorable fraction flow curve and more efficient sweep efficiency. Ultimately, oil recovery is increased and water production is decreased.

Properties of the polymers in this work are representative of partially hydrolyzed polyacrylamides (HPAM) that's frequently used in field applications. To simplify the multicomponent behavior, we simulated ECLIPSE black oil simulation with tabulated polymer properties as follows.

*Water Viscosity*

The viscosity of water is modified when a polymer is present in the solution. In this work, we assume the polymer solution and water are fully mixed in each grid block. The viscosity of a polymer solution is inputted as a relation between viscosity multiplier and polymer concentration ( $C_p$ ) (**Fig. 3.1**). This term is used to multiply pure water viscosity (**Eq. 3.1**) to make it more viscous resulting in a favorable mobility ratio.

$$\mu_{mix} = M(C_p) * \mu_w \dots\dots\dots(3.1)$$

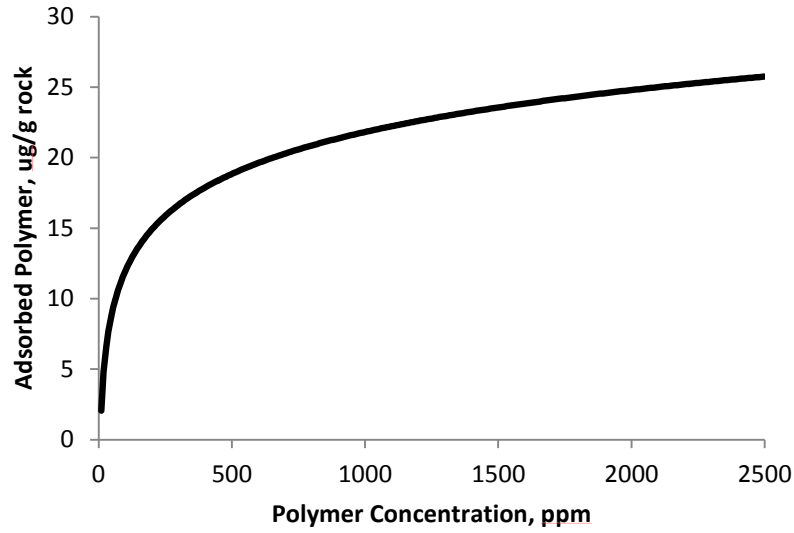


**Fig. 3.1—Viscosity multiplier,  $M(C_p)$  at various  $C_p$ : higher polymer concentration results in a more viscous fluid**

### *Polymer Adsorption*

Adsorption is treated as an instantaneous effect in the model. The effect of polymer adsorption is to create a stripped water bank at the leading edge of the slug. This water bank has mobility lower than the injected polymer solution and reduces the efficiency of polymer flood.

In this study, the adsorption effect is treated as a relation between the saturated concentration of polymer adsorbed by rock formation and the polymer concentration (**Fig. 3.2**). This effect is considered a reversible process; thus, the polymer adsorption is retraced whenever the local polymer concentration in the solution decreases.



**Fig. 3.2—Polymer adsorption at various concentrations for HPAM polymer: rock absorbs more polymer at higher concentration**

*Permeability Reduction*

A further effect of adsorption is a reduction in the permeability of the rock to the passage of polymer mixed water. This relation is directly related to the amount of adsorbed polymer and can be defined as a permeability reduction factor ( $R_k$ ) which can be calculated by **Eq. 3.2**

$$R_k = 1.0 + (RRF - 1.0) \frac{C_p^a}{C_p^{a_{max}}} \dots\dots\dots(3.2)$$

RRF is the residual resistance factor which equal to 3.0 for this study.  $C_p^a$  denotes adsorbed polymer concentration in the rock, and  $C_p^{a_{max}}$  is maximum adsorbed concentration that is defined as adsorption at 2500 ppm in this work.

### *Dead Pore Volume*

The dead pore volume represents the fraction of the total pore volume that is inaccessible to the polymer solution. This causes the polymer solution to travel faster than inactive tracer. In this study, a constant value of 0.2 is used as dead pore volume.

### **3.1.2 Performance Measures (Variable, Objective)**

#### *Objective Function*

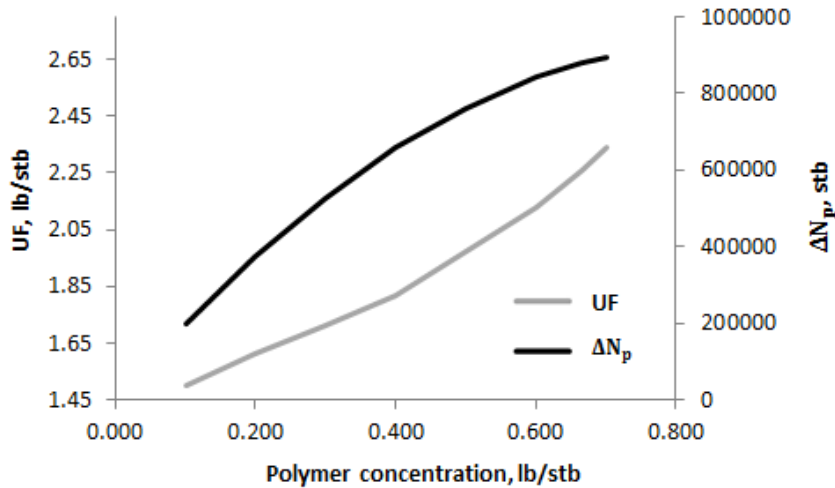
The conventional way to measure success of a polymer flood project is by production improvement, which is the difference between cumulative oil production in case polymer is injected and cumulative oil production in case of no polymer injection (**Eq. 3.3**). This objective is a standard objective in many petroleum engineering problems. The more production gain indicates success of the project

$$\Delta N_p = N_p(\text{with polymer injection}) - N_p(\text{without polymer injection}) \quad \dots\dots(3.3)$$

For polymer flood project, polymer utility factor (UF) is an indirect measurement (Clemens et al., 2011; Sharma et al., 2011). The polymer utility factor is a ratio between injected polymer and production improvement (**Eq. 3.4**). The UF implies efficiency of polymer usage. Large UF means the project requires large amounts of polymer for 1 STB of oil gain. In contrast, small UF indicates the project is efficient since it only requires small amounts of polymer for 1 STB of oil gain.

$$UF = \frac{\text{Total injected polymer}}{\Delta N_p} \dots\dots\dots(3.4)$$

This study considers these two objectives simultaneously. A successful project should have high  $\Delta N_p$  and small UF at the same time. However, in practice, these two requirements cannot be satisfied in chorus since high  $\Delta N_p$  always comes with the cost of poor efficiency (high UF). Sensitivity analysis of these objectives was run with varied polymer concentration (**Fig. 3.3**). The increasing polymer concentration provides more  $\Delta N_p$  and increases UF (poor performance). These two objectives are called *conflicting objectives* because both of them cannot be optimized simultaneously.



**Fig. 3.3—Sensitivity analysis of  $\Delta N_p$  and UF with increasing polymer concentration: both objectives increase with increasing concentration.**

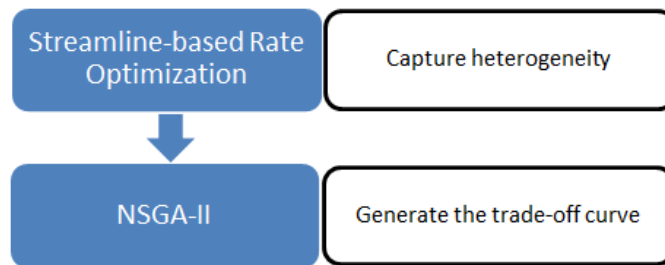
*Polymer Flood Control Variable*

The control variables in the polymer flood project are polymer concentration ( $C_p$ ), slug size, and initiation time. Al-Sofi and Blunt (2011) studied the effect of these parameters

and concluded that optimal slug size and concentration are controlled by economic limit, but the project always gains benefits from early initiation. This study implemented this idea by initializing polymer flood at the beginning of the simulation and leaving the other two variables,  $C_p$  and slug size, as control variables for the optimization process.

### 3.1.3 Two-Stage Optimization Workflow

The optimization workflow comprised of two stages: rate optimization and polymer flood optimization (**Fig. 10**). First, the heterogeneity of the reservoirs is accounted using streamline-based rate optimization by allocating production and injection rates. Next, the NSGA-II is coupled with the rate-optimized model to focus on polymer control variables. The result is represented as a compromised trade-off curve between production improvement ( $\Delta N_p$ ) and polymer utility factor (UF). The validation of this coupling is analyzed in **Appendix A**.



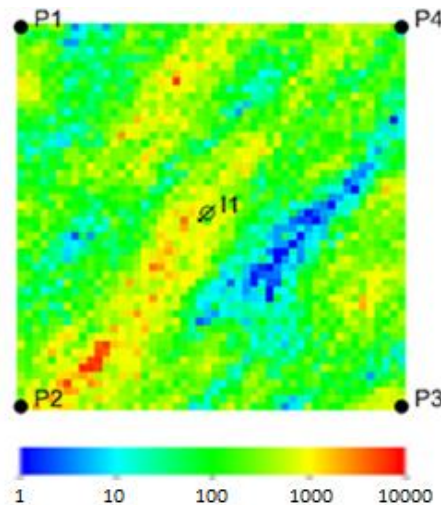
**Fig. 3.4—Two-stage coupling optimization workflow: Streamline and NSGA-II**

## 3.2 Application

In this section, we demonstrate the robustness of the approach by three cases: a 2D five-spot, a 3D field-scale polymer flood model, and optimization under geologic uncertainty.

### 3.2.1 Five-Spot Example

The 2D synthetic reservoir model (50x50x1 grid) has 4 producers on each corner and one injector at the center representing a five-spot pattern. The model has a fixed porosity of 0.225, initial water saturation of 0.18, initial pressure of 4,000 psia and spatially heterogeneous permeability with high-permeability streak in NE-SW direction (**Fig. 3.5**). The reservoir is produced under injection/production rates constraints (**Table 3.1**).



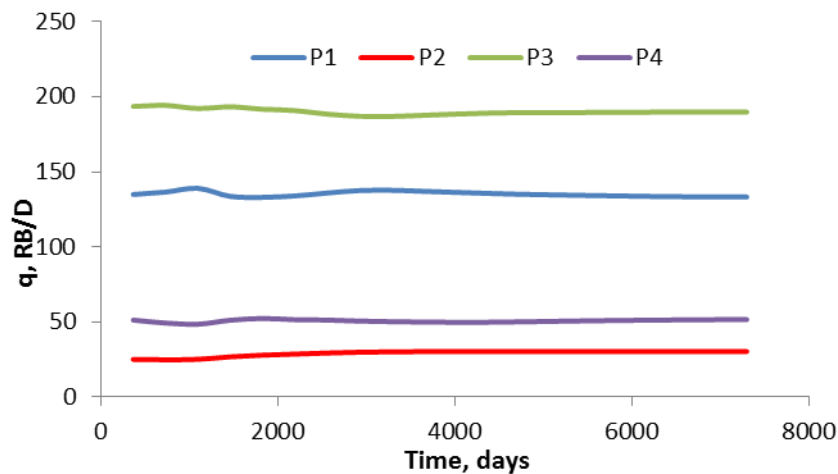
**Fig. 3.5—2D five-spot example: permeability distribution and well location**



Table 3.1—Five-spot model production constraints	
Constraints	Limit
Maximum field production rate	405 RB/D
Maximum field injection rate	405 RB/D
Maximum well production rate	250 RB/D
Minimum bottom hole pressure (producer)	1,000 psia
Maximum bottom hole pressure (injector)	7,000 psia

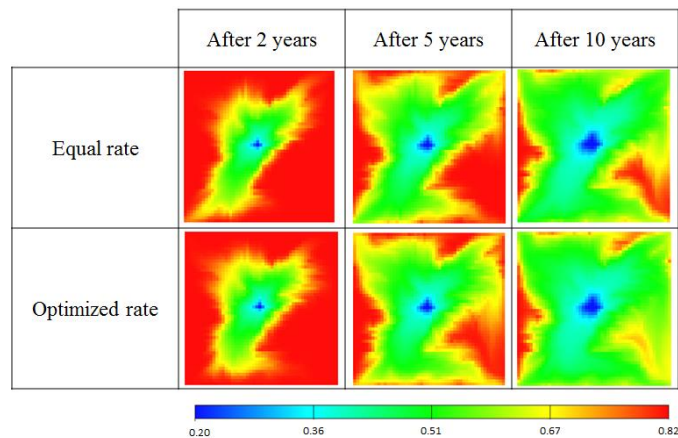
### Rate Optimization

A high permeability streak connects the injector to producer P2 and P4 (**Fig. 3.5**) leading to preferential fluid movement and early water breakthrough from these wells. The streamline-based rate optimization method overcomes this problem by reallocating production rate to equalize arrival time from all producers. After optimization, production rates from the wells in the preferential path (P2, P4) are reduced and the rates are relocated to increase production from other wells (P1, P3) (**Fig. 3.6**).



**Fig. 3.6—Production rate allocation to each producer with streamline-based method**

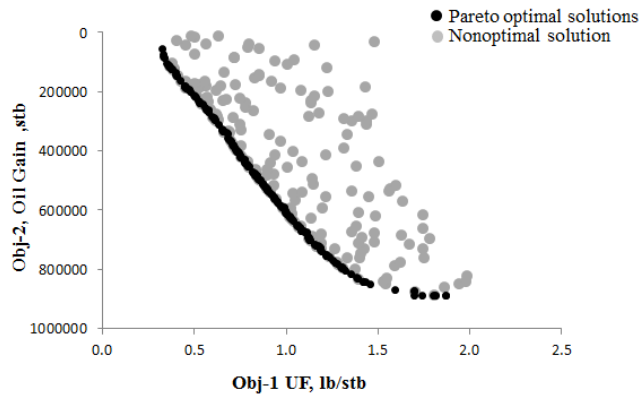
The benefits of this method are illustrated by comparing oil saturation map over time from two cases: (I) base case with equal production rates from all wells and (II) the streamline-based rate optimization (**Fig. 3.7**). With the optimization, the water front shows more uniform movement. Therefore, water breakthrough is delayed, oil production increased, and field sweep efficiency improved.



**Fig. 3.7—Oil saturation map of the two cases: I) Base case with equal production rates and II) Optimized case by the streamlines method: optimized case delays water breakthrough results in less oil saturation left in the reservoir.**

### *Result in Objective Space*

Once the production and injection rates are optimized, the next step is designing optimal polymer-flood control parameters. Polymer concentration is varied in a range of 1 to 2000 ppm and slug size is considered in terms of injection duration from 1 to 20 years. The NSGA-II algorithm is performed to search optimal combination of these parameters to generate a trade-off curve between  $\Delta N_p$  and UF.

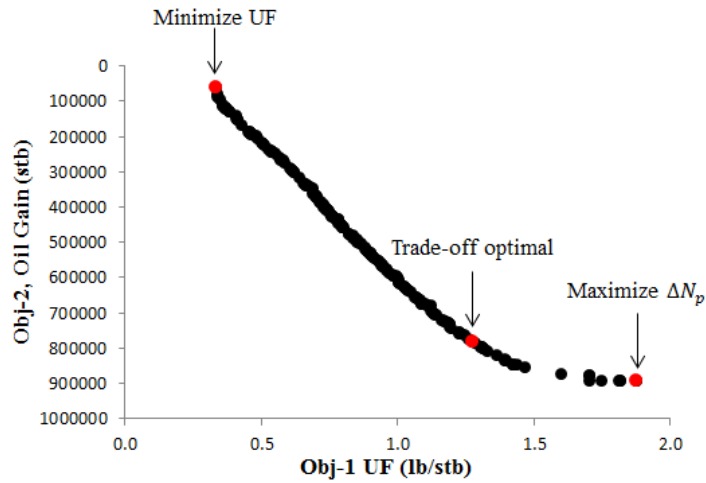


**Fig 3.8—NSGA-II algorithm generates a trade-off relation between UF and  $\Delta N_p$  (2D five-spot model): gray dots are nonoptimal solutions, black dots are Pareto optimal solutions.**

Each population in the objective space represents result from a combination of  $C_p$  and slug size. During the initial generation, the populations spread throughout the objective space as in the gray dots (**Fig.3.8**). The dominance relationship in the NSGA-II algorithm selects populations with low domination ranking and moves them close to the Pareto front (black dots). The stopping criterion is reached when all the populations become rank one, which implies all populations are close to Pareto front.

The Pareto front represents a compromised trade-off between the two objectives. The result will be used as a decision tool to determine an optimal trade-off between production gain and polymer efficiency. To illustrate the benefits of this trade-off curve on post-optimization decision, three different optimal points are selected (**Fig 3.9**). The leftmost point with the lowest value of UF is representative of optimization with the main focus on preserving polymer efficiency. The rightmost point with the largest  $\Delta N_p$

is representative of optimization with the main focus on maximizing production. The \ point in between represents a trade-off optimal solution was picked from the MSE algorithm. The trade-off optimal point is considered as a compromised solution between production improvement and polymer efficiency. The control variables for these three points are summarized in **Table 3.2**.

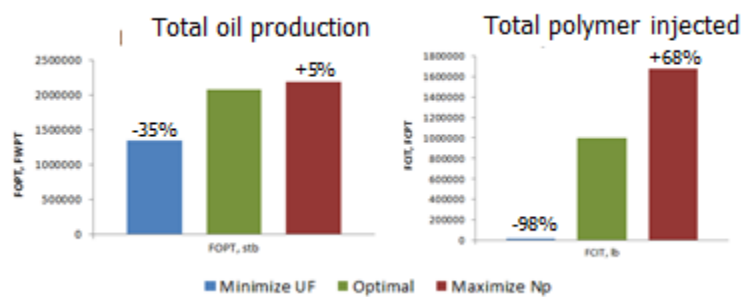


**Fig. 3.9—Three optimal points with different focuses (2D five-spot model): (left) minimizing UF, (right) maximizing  $\Delta N_p$ , and (middle) the trade-off optimal point represents a compromised solution between efficiency and production.**

<b>Table 3.2—Control variables for three optimal points (2D five-spot case)</b>		
Optimal point	$C_p$ , ppm	Slug size, year
Minimize UF (focus on efficiency)	778	1
Trade-off optimal	1,812	10
Maximize $\Delta N_p$ (focus on production)	2,000	16

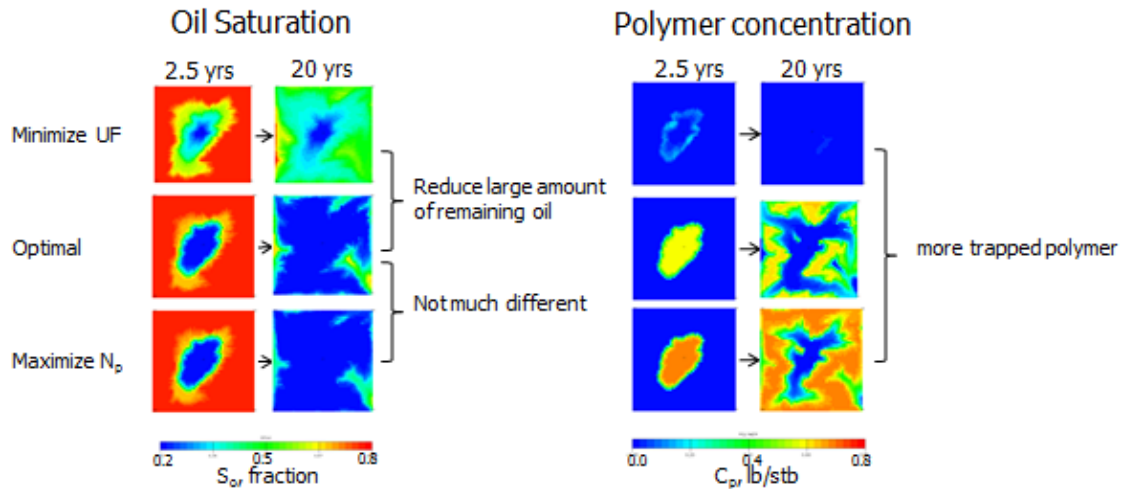
In a comparison of cumulative oil production, cumulative polymer injected, and polymer utility factor from the three optimal points (**Fig 3.10**), the polymer efficiency

optimization gives the smallest UF and requires very small amounts of polymer injection. However, the cumulative oil production from this case is 35% less than the trade-off optimal point. In contrast, maximizing production gives the largest amount of cumulative oil production, but with the highest polymer UF. In this case, the increment is only 5% and it requires 68% more polymer injection from the trade-off optimal point.



**Fig. 3.10—Cumulative oil production, cumulative polymer injection, and UF from the three optimal points (2D five-spot model)**

The 2D maps of oil saturation and polymer concentration at 2.5 years and at the end of the simulation (20 years) demonstrate differences between the three optimal points (**Fig. 3.11**). The increasing amount of injected polymer from the minimized UF case to trade-off optimal reduces large amount of remaining oil. However, increasing polymer beyond this point to focus only on maximized production case does not give a big difference in the remaining oil. In contrast, there is increased amounts of trapped polymer in the reservoir. These results clearly demonstrate that polymer injection beyond the optimal design is inefficient since much more polymer is required to bring about a relatively small increase oil production.



**Fig. 3.11—2D map from three optimal points (left), oil Saturation maps (right), polymer concentration maps. Increasing polymer beyond the optimal design results in much more trapped polymer with small improvements in remaining oil-in-place.**

### *Result in Variable Space*

For the variable space, most of the Pareto optimal solutions have concentration higher than 1000 ppm with slug size ranging between 0 to 10 years (**Fig 3.12**). The two dimensional distribution plot shows most of the optimal populations have high concentrations and the higher concentration tends to have the larger slug size (**Fig 3.13**). This behavior is consistent with the small production/injection rates for this synthetic case. The wells are still capable of handling increasing pressure drop due to more viscous fluid. With the larger rate, the optimal polymer concentration is expected to be affected by high polymer concentration.

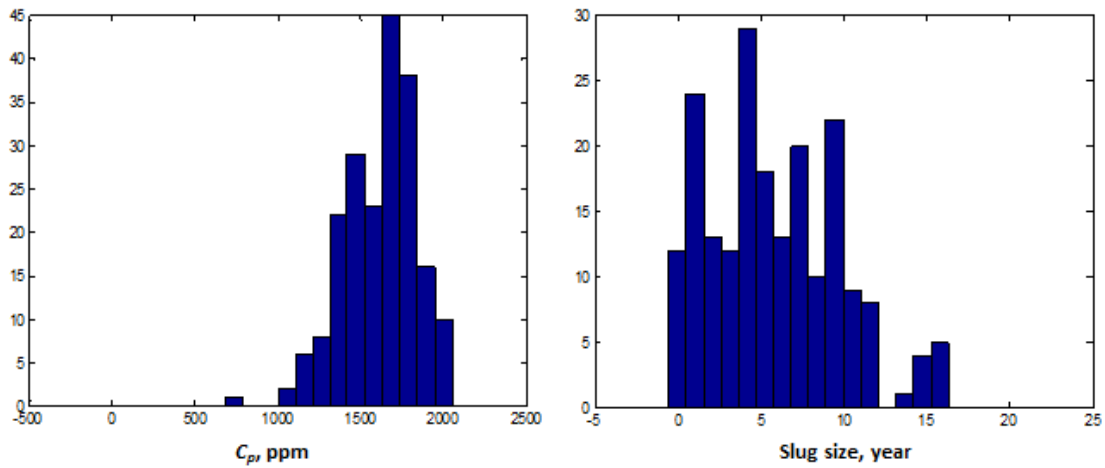


Fig. 3.12—Histogram of optimal variables, polymer concentration and slug size. Most of the populations have high concentrations with intermediate slug size.

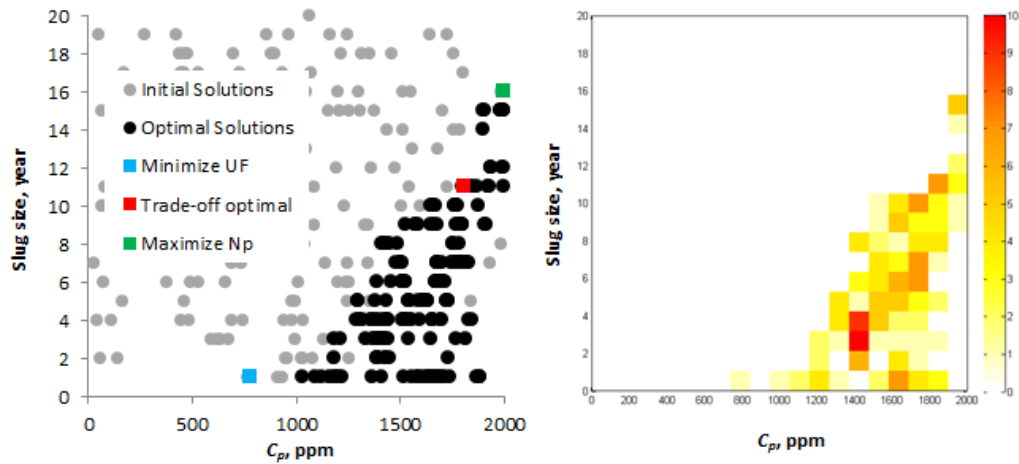
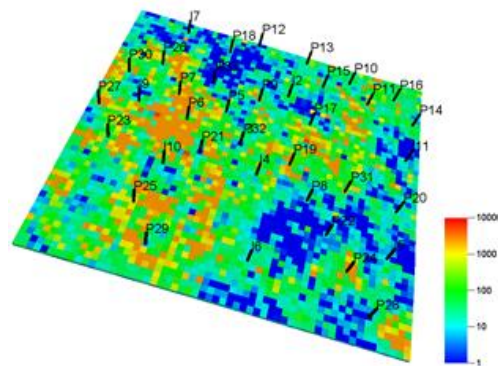


Fig. 3.13—Two dimensional distribution plots in variable space. Higher polymer concentration tends to have higher slug size. The trade-off optimal solution is shown in the middle among all the solutions.

### 3.2.2 Field-Scale Polymer Flood Optimization

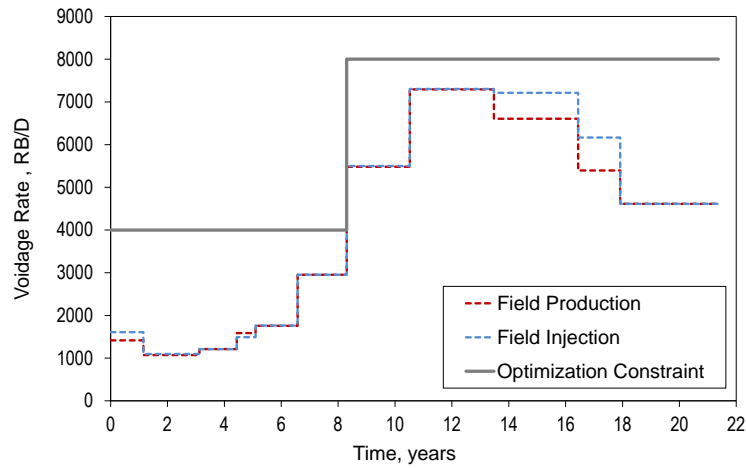
The reservoir model in this example corresponds to the Goldsmith San Andres Unit (GSAU) in the Goldsmith field (Jasek et al., 1998). To our knowledge, no polymer flooding was carried out in this field. Nevertheless, the highly heterogeneous permeability and complex well system makes it a good candidate for illustrating our approach. The 3D model (58x53x10) contains 31 producers and 11 injectors with 21 years of production history (**Fig. 3.14**)



**Fig. 3.14—Goldsmith GSAU field: complexity of the field makes it appropriate for large scale field demonstration.**

First, the arrival time was optimized by streamlines to counteract the heterogeneity effects. Sharma et al. (2011) optimized this field's control with the base case production/injection rates from the the actual field history during the first 21 years of water flooding (**Fig. 3.15**). Wells production and pressure constraints are imposed during optimization (**Table 3.3**).





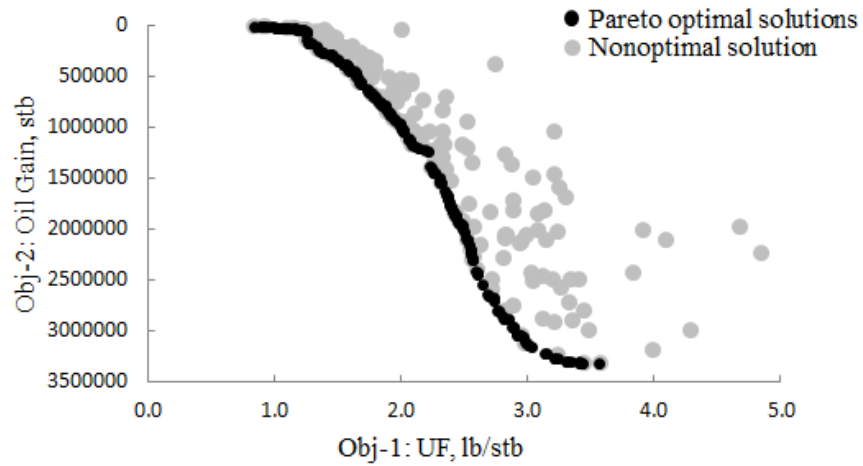
**Fig. 3.15—Field production/injection history. Maximum production/injection rate: 4000 RB/D (first 8 years), 8000 RB/D (next 13 years).**

Table 3.3—Field-scale model production constraints	
Constraints	Limit
Maximum field production rate (first 8 year)	4,000 RB/D
Maximum field production rate (next 13 year)	8,000 RB/D
Maximum well production rate	1,200 RB/D
Maximum well injection rate	900 RB/D
Minimum bottom hole pressure (producer)	1,000 psia
Maximum bottom hole pressure (injector)	4,500 psia

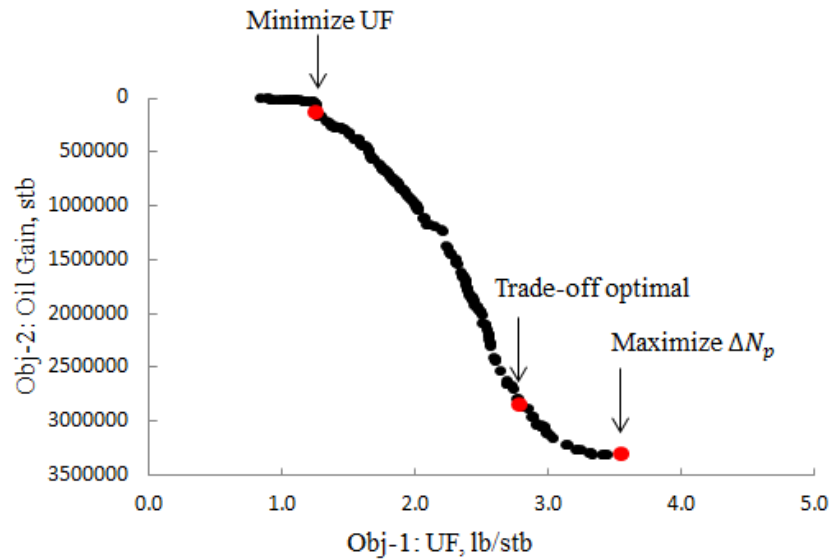
### *Result in Objective Space*

The rate-optimized model is coupled with the NSGA-II for polymer flood optimization (**Fig. 3.16**) with the identical polymer properties and ranges from the previous case. As before, the Pareto front represents a compromised trade-off between the two objectives. The post-optimization decision was made from the Pareto front. Three different points

with different focuses: (I) minimizing UF, (II) maximizing  $\Delta N_p$ , and (III) a trade-off optimal point, were picked from the Pareto front (**Fig. 3.17 and Table 3.4**).



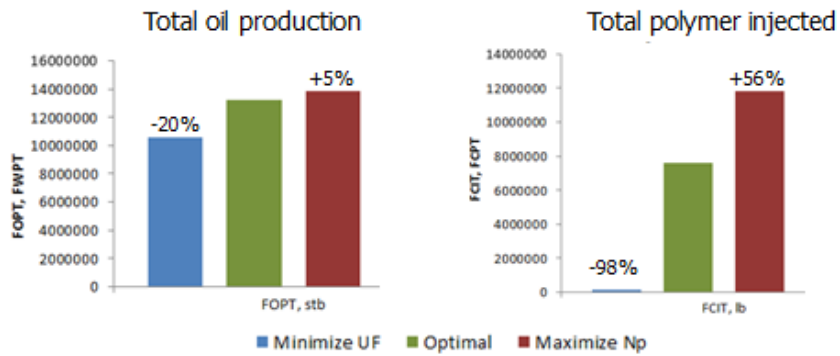
**Fig. 3.16**—NSGA-II algorithm generates a trade-off relation between UF and  $\Delta N_p$  (field-scale model): gray dots are nonoptimal solutions, black dots are Pareto optimal solutions.



**Fig. 3.17**—Three optimal points with different focuses (field-scale model): (left) minimizing UF, (right) maximizing  $\Delta N_p$ , and (middle) the trade-off optimal point represents the compromised solution between efficiency and production.

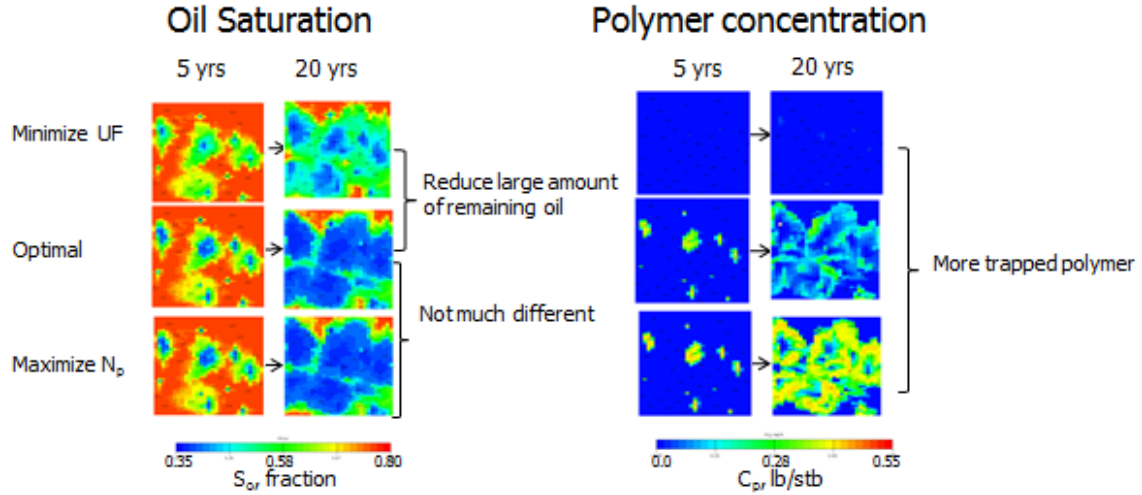
Optimal point	$C_p$ , ppm	Slug size, year
Minimize UF (focus on efficiency)	114	7
Trade-off optimal	1115	15
Maximize $\Delta N_p$ (focus on production)	1256	20

Injecting polymer beyond the trade-off optimal point would be inefficient, and resulting in relatively marginal increase in oil recovery (**Fig. 3.18**). Focusing only on maximizing oil production results in poor polymer efficiency, which requires 56% more total injected polymer for only 5% improvement from the trade-off optimal solution.



**Fig. 3.18—Cumulative oil production, cumulative polymer injection, and UF from three different optimal points (field-scale model)**

Beyond the optimal design, much more injected polymers are trapped in the reservoir (**Fig. 3.19-right**) without significant change in remaining oil in place (**Fig. 3.19-left**). The same conclusions can be drawn from both the 2D five-spot model and the field-scale model, which prove the benefits and robustness of this approach over the conventional single-objective approach.



**Fig. 3.19—2D map from three optimal points: (left) oil Saturation maps, (right) polymer concentration maps: injecting polymer beyond the optimal design gives small change in remaining oil saturation, but the amount of polymer trapped in the reservoir increases drastically.**

### *Result in Variable Space*

In the variable space, most of the optimal solutions have polymer concentration around 600 to 1,300 ppm and none of them has concentration exceeding 1,300 ppm (**Fig. 3.20-left**). The majority of the optimal solutions have slug size around 5 to 15 years (**Fig. 3.20-right**). Higher concentration tends to have higher slug size (**Fig. 3.21**). Since this field-scale example has higher field production rate and restricted bottomhole pressure control, injection with very high polymer concentration becomes unfavorable since the more viscous fluid introduce higher pressure drop.

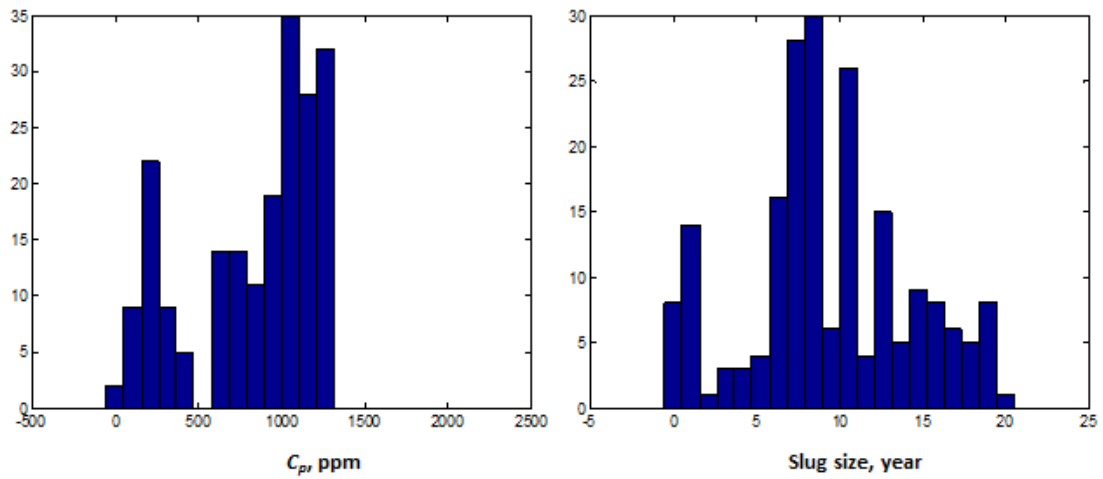


Fig. 3.20—Histogram of optimal variables, polymer concentration and slug size. No population has polymer concentration exceed 1,300 ppm.

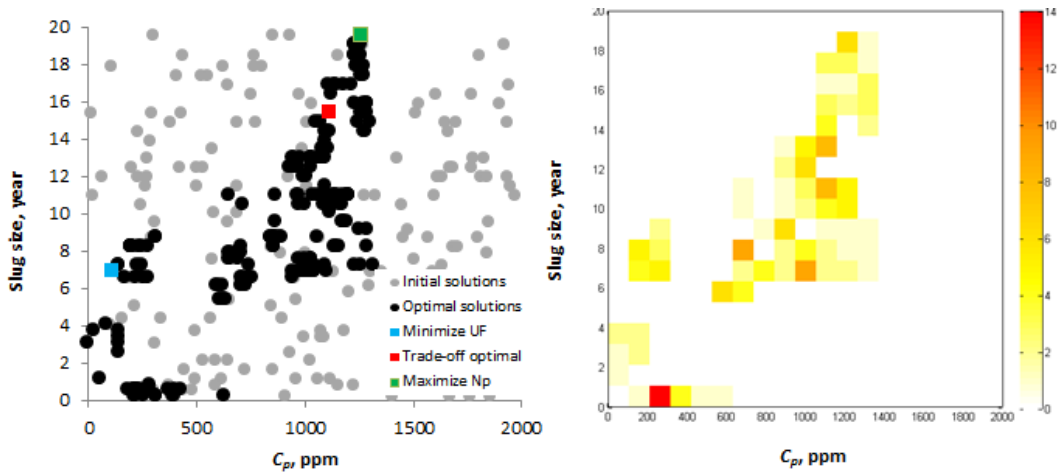


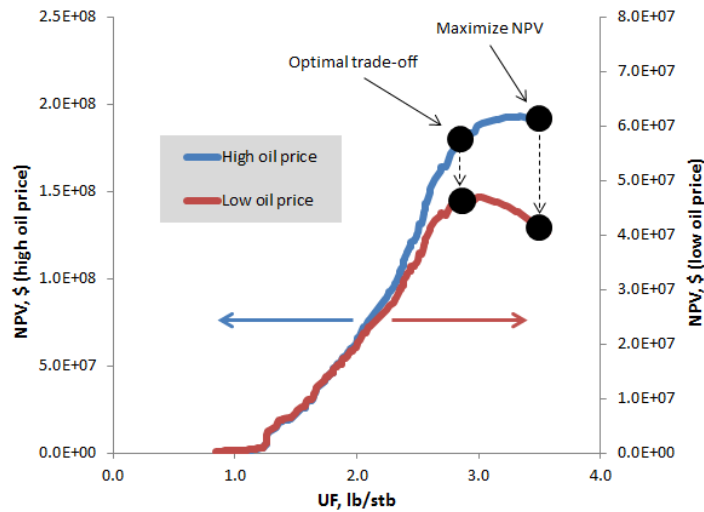
Fig. 3.21—Two dimensional distribution plots in variable space. Higher polymer concentration tends to have higher slug size. The trade-off optimal solution is in the middle among all the solutions.

### *Advantages to Engineering Decision*

The result from multiobjective approach is compared to the conventional single objective optimization approach. Conventionally, the optimization is done based on the single objective, NPV without considering the production efficiency. However, the NPV depends on some economic assumption such as oil price. The change in economic assumption affects an optimal solution. In the favorable economic condition, oil price is very high and NPV increases regardless of low operational efficiency. In contrast, in an unfavorable economic condition, oil price is not high and the efficiency gains more attention.

In this case, we simply assume oil price and polymer operating cost in two scenarios, high oil price and low oil price (**Table 3.5**). The Pareto front from Goldsmith field (**Fig. 3.17**) is analyzed under these two scenarios. For the high oil price scenario, the highest NPV can be acquired at the highest UF solution (**Fig. 3.22**). The trade-off optimal point gives slightly lower than the highest NPV and it is located in the ramp-down slope of the plot. However, under the low oil price scenario, the highest NPV solution under favorable economic condition becomes deteriorated. In contrast, the trade-off optimal solution provides better NPV.

Scenario	Oil price \$/STB	Polymer operating cost, \$/lb
High oil price	75	5
Low oil price	30	5



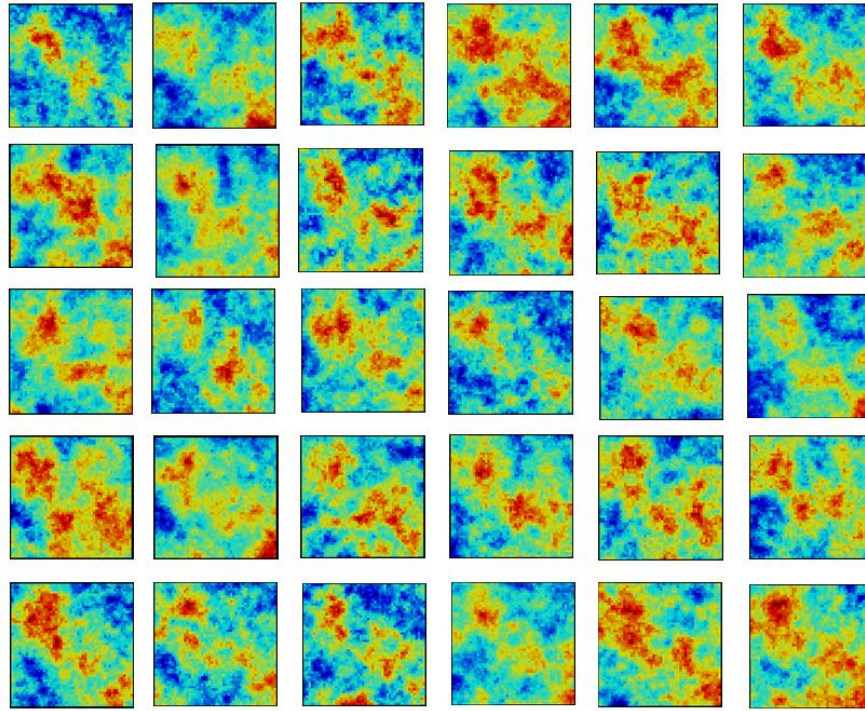
**Fig. 3.22—NPV result of the two scenarios, high and low oil price. The highest NPV point from the high oil price scenario is not an optimal solution in the low oil price scenario. The optimal trade-off point provides a good NPV result in both cases.**

Generally, the economic factor such as NPV is a good objective of optimization. The multiobjective optimization provides another interpretation to the problem considering economic, and performance simultaneously. The decision considering these two factors together can be made with more confidence.

### 3.2.3 Geologic Uncertainty

We demonstrated application of our approach using multiple realizations to address geologic uncertainty. Multiple porosity distributions were generated using sequential gaussian simulation (SGS). Then, the permeability field was generated via logarithmic permeability-porosity correlation (**Fig. 3.23**).





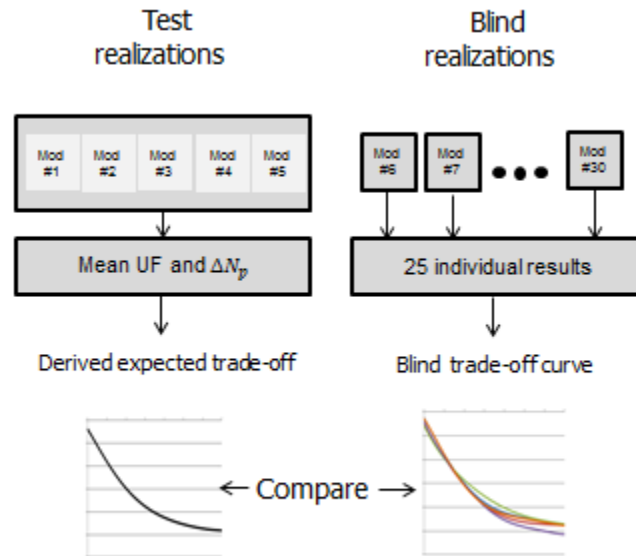
**Fig. 3.23—30 porosity realizations from SGS with high porosity channel in NW-SE direction**

Five realizations were selected randomly to be optimized with our approach. The streamline-based rate optimization was carried out using the expected value under the risk neutral condition (Alhuthali et al., 2008). The domination concept of NSGA-II is still the same as single realization optimization except that the objective values were calculated from the expected value (**Eq. 3.5** and **Eq. 3.6**).

$$\Delta N_{p_{avg}} = \sum_{i=1}^{N_{realization}} \Delta N_{p_i} \dots\dots\dots(3.5)$$

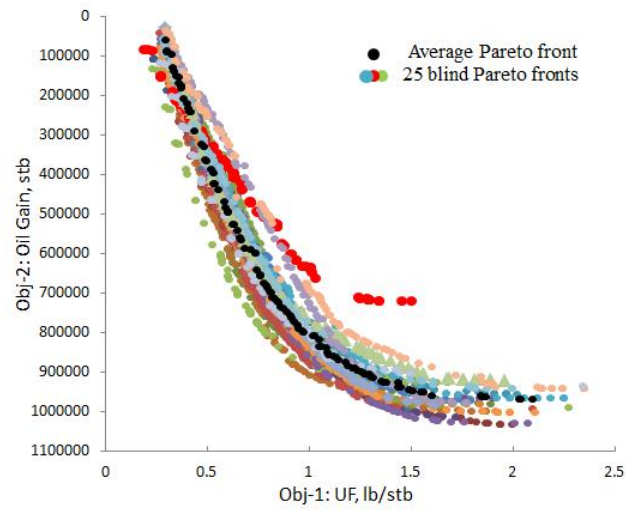
$$UF_{avg} = \sum_{i=1}^{N_{realization}} UF_i \dots\dots\dots(3.6)$$

The other 25 realizations were considered as blind models. These models were optimized with our approach individually. The Pareto front results from multiple realizations was compared to the Pareto fronts from the blind models (**Fig. 3.24**).



**Fig. 3.24—Workflow to compare optimization under geologic uncertainty with blind realizations. The expected Pareto front from five realizations is compared with multiple blinds realizations.**

The Pareto fronts generated from the blind realizations can be biased toward extreme pessimistic or optimistic results. An inappropriate design might be picked from the post-optimization decision from the biased Pareto front. The average Pareto front from multiple realizations is a good representative to capture geologic uncertainty from multiple outcomes (**Fig. 3.25**). More robust decision can be made from this Pareto front.



**Fig. 3.25—Pareto fronts calculated from the average value and multiple individual blind realizations: The average Pareto front is a good representative to capture geologic uncertainty.**

## CHAPTER IV

### CONCLUSIONS AND RECOMMENDATIONS

#### 4.1 Conclusions

In this paper, we have proposed a technique to explore the trade-off relationship between cumulative oil production and polymer efficiency in polymer flood projects. The multiobjective-framework approach can be used as an effective tool for production optimization. The major findings in the proposed approach are as follows:

- The multiobjective genetic algorithm can generate a set of optimal solutions which represent an appropriate compromise for maximizing oil production while maintaining low polymer utility factor.
- Instead of relying on conventional single-objective production optimization, the multiobjective approach provides more comprehensive understanding between production improvement and operational efficiency. The optimal polymer flood design can be selected from the trade-off curve. Beyond the optimal design, much more polymer injection is required to produce a relatively minor increase in oil production, and the polymer utility factor increases substantially.
- The robustness and practical feasibility of our proposed approach have been demonstrated using the synthetic model and the field-scale reservoir model. Stochastic optimization was also implemented to take geologic uncertainty into consideration. The trade-off curve from the stochastic approach is representative

of multiple realizations since it can avoid extreme optimistic and pessimistic results.

## **4.2 Recommendations**

Several recommendations that could improve the performance of the multiobjective performance measure approach are listed below:

- The current work coupled two algorithms separately under the assumption that the result from the first stage (rate optimization via streamlines method) would not have strong influence on the second stage (multiobjective polymer flood optimization). However, the change in polymer control parameters affects mobility field and alter the results from the rate optimization. An improvement can be made to seamlessly couple these two steps.
- In this work, we have made an assumption that all wells in the model share same polymer control variables (polymer concentration and slug size) for simplicity. Practically, different controls can be assigned to each individual well. The objectives can be improved by imposing more control variables. However, the trend of Pareto front is expected to be the same.
- The idea of multiobjective optimization is applicable for other types of EOR project. For example, it can be used to optimize ASP or CO<sub>2</sub> flooding. The appropriate performance measure objectives are required for new EOR types.

## NOMENCLATURE

### Chapter II

$p(\mathbf{q})$	Scalar objective function, $t^2$ , sq day
$\mathbf{q}$	Total fluid rate vector, $L^3/t$ , B/D [ $m^3/d$ ]
$\mathbf{t}$	Arrival time vector, t, day(s)
$t_{i,m}$	Arrival time at producer $i$ which belongs to group $m$ , t, day(s)
$t_{d,m}$	Desired arrival time for group $m$ , t, day(s)
$\eta$	Norm weight trade-off between equalizing arrival time and production acceleration
$\tau_{l,i}$	Time of flight of streamlines $l$ belongs to producer $i$
$N_{prod,m}$	Number of production well(s) in group $m$
$N_{group}$	Number of group(s)
LHS	Latin Hypercube Sampling
GA	Genetic Algorithm
NSGA	Nondominated sorting genetic algorithm
$w_i$	Weighting factor for scalarization method
$f_i$	Objective function $i$
$u_k$	Random number at $k^{th}$ genome for crossover operation
$\beta_k$	Weighting factor at $k^{th}$ genome for crossover operation

$\eta_c$	Distribution index for crossover operation
$c_{i,k}$	Child i at $k^{th}$ genome
$p_k$	Selected parent at $k^{th}$ genome
$p_k^u, p_k^l$	Upper and lower limit of $k^{th}$ genome
$\delta_k$	Weighting factor at $k^{th}$ genome for mutation operation

### Chapter III

$\mu_{mix}$	Viscosity of water solution with polymer, cp
$\mu_w$	Viscosity of pure water, cp
$C_p$	Polymer concentration, ppm
$M(C_p)$	Viscosity multiplier
RRF	Residual resistance factor
$R_k$	Permeability reduction factor
$C_p^a$	Absorbed polymer concentration in rock, ppm
$C_p^{a \max}$	Maximum absorbed polymer concentration in rock, ppm
$\Delta N_p$	Cumulative oil gain from polymer injection, stb
UF	Polymer Utility factor, lb/stb
$\Delta N_{p \text{ avg}}$	Average cumulative oil gain from multiple realizations
$UF_{avg}$	Average polymer utility factor from multiple realizations
$N_{realization}$	Number of realizations

## REFERENCES

- Abedi, M. and Algharaib, M. 2012. Optimization of Polymer Flood Performance by Preflush Injection - Numerical Investigation. Presented at the 2012 SPE Kuwait International Petroleum Conference and Exhibition. SPE-163280-MS.
- Al-Sofi, A.M. and Blunt, M.J. 2011. The Design and Optimization of Polymer Flooding under Uncertainty. Presented at the SPE Enhanced Oil Recovery Conference. SPE-145110-MS.
- Alhuthali, A.H.H., Datta-Gupta, A., Yuen, B.B.W. et al. 2008. Optimal Rate Control under Geologic Uncertainty. Presented at the SPE/DOE Symposium on Improved Oil Recovery. SPE-113628-MS.
- Anderson, G.A., Delshad, M., King, C.L.B. et al. 2006. Optimization of Chemical Flooding in a Mixed-Wet Dolomite Reservoir. Presented at the SPE/DOE Symposium on Improved Oil Recovery. SPE-100082-MS.
- Barua, J., Prescott, T., and Haldorsen, H.H. 1986. Financial and Technical Decision Making for Surfactant Flooding. Presented at the SPE California Regional Meeting. SPE-15074-MS.
- Cardoso, M.A. 2009. Reduced-Order Models for Reservoir Simulation. Presented at the SPE International Student Paper Contest. SPE-129636.
- Clemens, T., Abdev, J., and Thiele, M. 2011. Improved Polymer-Flood Management Using Streamlines. *SPE Reservoir Evaluation & Engineering* **14** (2): 171-181. SPE-132774-PA.
- Datta-Gupta, A. and King, M.J. 2007. *Streamline Simulation : Theory and Practice*. Spe Textbook Series. Richardson, Texas: Society of Petroleum Engineers.
- Deb, K. and Agrawal, R. 1995. Simulated Binary Crossover for Continuous Search Space. *Complex Systems* **9**: 115-148.
- Deb, K., Pratap, A., Agarwal, S. et al. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-II. *Evolutionary Computation, IEEE Transactions* **6** (2): 182-197.
- DeHekker, T.G., Bowzer, J.L., Coleman, R.V. et al. 1986. A Progress Report on Polymer-Augmented Waterflooding in Wyoming's North Oregon Basin and Byron Fields. Presented at the SPE Enhanced Oil Recovery Symposium. SPE-14953-MS.



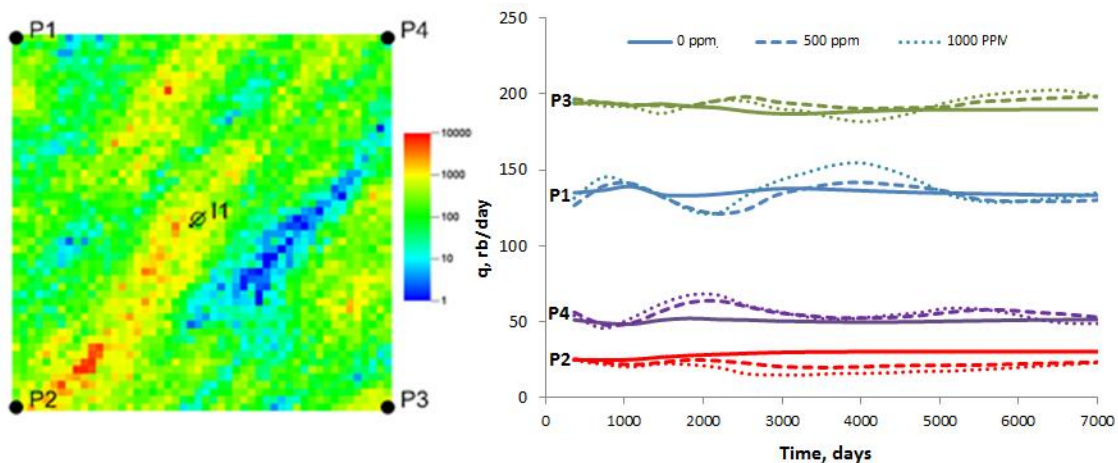
- Doren, J.V., Douma, S.G., Wassing, L.B.M. et al. 2011. Adjoint-Based Optimization of Polymer Flooding. Presented at the SPE Enhanced Oil Recovery Conference. SPE-144024-MS.
- Ferraro, P. and Verga, F. 2009. Use of Evolutionary Algorithms in Single and Multi-Objective Optimization Techniques for Assisted History Matching. Presented at the Offshore Mediterranean Conference and Exhibition. OMC-2009-079.
- Hajizadeh, Y., Christie, M.A., and Demyanov, V. 2011. Towards Multiobjective History Matching: Faster Convergence and Uncertainty Quantification. Presented at the SPE Reservoir Simulation Symposium. SPE-141111-MS.
- Han, Y., Park, C., and Kang, J.M. 2010. Estimation of Future Production Performance Based on Multi-Objective History Matching in a Waterflooding Project. Presented at the SPE EUROPEC/EAGE Annual Conference and Exhibition. SPE-130500-MS.
- Jasek, D.E., Frank, J.R., Mathis, L.S. et al. 1998. Goldsmith San Andres Unit Co2 Pilot - Design, Implementation, and Early Performance. Presented at the SPE Annual Technical Conference and Exhibition. SPE-48945-MS.
- Lake, L.W. 1989. *Enhanced Oil Recovery*. Englewood Cliffs, New Jersey: Prentice Hall.
- Li, Y. 2012. Momcmc: An Efficient Monte Carlo Method for Multi-Objective Sampling over Real Parameter Space. *Computers & Mathematics with Applications* **64** (11): 3542-3556.
- Ma, X. 2008. History Matching and Uncertainty Quantification Using Sampling Method. PhD dissertation, Texas A&M University. College Station, Texas.
- Mohamed, L., Christie, M.A., and Demyanov, V. 2011. History Matching and Uncertainty Quantification: Multiobjective Particle Swarm Optimisation Approach. Presented at the SPE EUROPEC/EAGE Annual Conference and Exhibition. SPE-143067-MS.
- Morales, A.N., Nasrabadi, H., and Zhu, D. 2011. A New Modified Genetic Algorithm for Well Placement Optimization under Geological Uncertainties. Presented at the SPE EUROPEC/EAGE Annual Conference and Exhibition. SPE-143617-MS.
- Nasrabadi, H., Morales, A., and Zhu, D. 2012. Well Placement Optimization: A Survey with Special Focus on Application for Gas/Gas-Condensate Reservoirs. *Journal of Natural Gas Science and Engineering* **5** (0): 6-16.

- Nghiem, L.X., Yang, C., Shrivastava, V.K. et al. 2009. Optimization of Residual Gas and Solubility Trapping for CO<sub>2</sub> Sequestration in Saline Aquifers. Presented at the SPE Reservoir Simulation Symposium. SPE-119080-MS.
- Park, H.-Y., King, M., and Datta-Gupta, A. 2013. Handling Conflicting Multiple Objectives Using Pareto-Based Evolutionary Algorithm for History Matching of Reservoir Performance. Presented at the 2013 SPE Reservoir Simulation Symposium. SPE-163623-MS.
- Pope, G.A., Wang, B., and Tsaur, K. 1979. A Sensitivity Study of Micellar/Polymer Flooding. *Society of Petroleum Engineers Journal* **19** (6): 357-368.
- Sayyafzadeh, M., Haghghi, M., and Carter, J.N. 2012. Regularization in History Matching Using Multi-Objective Genetic Algorithm and Bayesian Framework. Presented at the SPE Europec/EAGE Annual Conference. SPE-154544-MS.
- Schulze-riegert, R., Bagheri, M., Krosche, M. et al. 2011. Multiple-Objective Optimization Applied to Well Path Design under Geological Uncertainty. Presented at the SPE Reservoir Simulation Symposium. SPE-141712-MS.
- Schulze-Riegert, R.W., Krosche, M., Fahimuddin, A. et al. 2007. Multiobjective Optimization with Application to Model Validation and Uncertainty Quantification. Presented at the SPE Middle East Oil and Gas Show and Conference. SPE-105313-MS.
- Sharma, M., Taware, S.V., and Datta-gupta, A. 2011. Optimizing Polymerflood Via Rate Control. Presented at the SPE Enhanced Oil Recovery Conference. SPE-144833-MS.
- Srinivas, N. and Deb, K. 1994. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*. **2** (3): 221-248.
- Taware, S.V., Sharma, M., Alhuthali, A. et al. 2010. Optimal Water Flood Management under Geological Uncertainty Using Accelerated Production Strategy. Presented at the SPE Annual Technical Conference and Exhibition. SPE-133882-MS.
- Van Essen, G.M., Zandvliet, M., Hof, P.V.d. et al. 2009. Robust Waterflooding Optimization of Multiple Geological Scenarios. *SPE Journal* **14** (1): 202-210. SPE-102913-PA.
- Yin, J., Park, H., Datta-Gupta, A. et al. 2010. A Hierarchical Streamline-Assisted History Matching Approach with Global and Local Parameter Updates. Presented at the SPE Western Regional Meeting. SPE-132642-MS.

## APPENDIX A

### VALIDATION OF THE TWO-STAGE COUPLING OPTIMIZATION

The coupling between the two methods preserves the efficiency of each algorithm. The streamline-based rate optimization works efficiently with analytical sensitivity calculation. The population-based method such as NSGA-II fits with a small number of control variables which is only two ( $C_p$  and slug size). However, optimizing polymer control variables may change the mobility field, which changes the rate optimization result. To observe this effect, we performed the rate optimization on the synthetic five-spot model with different polymer concentrations (**Fig. A.1**).



**Fig. A.1—Rate optimization result with different polymer concentrations: changing concentration only slightly affects rate optimization results.**

The streamline-based rate optimization was run with three different polymer concentrations: 0 ppm, 500 ppm, and 1,000 ppm. The production rate from the wells in high-permeability streak (P2, P4) are reduced to equalize arrival time with wells with low-permeability path (P1, P3). Different polymer concentrations change field mobility

and results in small changes in rate optimization result (**Fig.A.1**). The results from different concentrations are very close and fluctuate around the same range since the main control factor of mobility is permeability heterogeneity. The optimized rates from different concentration still follow the permeability streak trend. This analysis confirms the concept of two-stage optimization and allows the genetic algorithm to rely on the single result from streamline-based rate optimization.

## APPENDIX B

### DETAILS ON STREAMLINE-BASED RATE OPTIMIZATION

#### B.1 Optimization after Breakthrough

The optimization is carried out after water breakthrough at a well by incorporating penalized term into the calculated arrival time in order to prevent allocating high production rates to the high water-cut wells. The arrival time is modified to incorporate water-cut as follows:

$$t'_{i,m}(\mathbf{q}) = t_{i,m}(\mathbf{q}) * (1 - f_{w_{i,m}})^{\alpha} \dots\dots\dots(B.1)$$

If the water cut is zero, the modified arrival time is the same as the original arrival time. When the water cut at the well is greater than zero, the original arrival time will be reduced based on the level of water-cut. The extent of reduction can be controlled by the exponent term,  $\alpha$ . As a consequence, the rate allocation to wells with high water-cut will be lowered.

A similar modification is done to the arrival time after polymer breakthrough based on producing polymer concentration at the well as follows.

$$t''_{i,m}(\mathbf{q}) = t'_{i,m}(\mathbf{q}) * (1 - f_{pol_{i,m}})^{\alpha} \dots\dots\dots(B.2)$$

where

$$f_{pol_{i,m}} = \frac{\text{polymer concentration at well}}{\text{Maximum allowable polymer concentration}} \dots\dots\dots(B.3)$$

For gas breakthrough, the well GOR is incorporated to the arrival time as follows

$$t'_{i,m}(q) = t_{i,m}(q) * (1 - f_{gas_{i,m}})^\alpha \dots\dots\dots(B.4)$$

where

$$f_{gas_{i,m}} = \frac{\text{Well GOR}}{\text{Maximum allowable GOR}} \dots\dots\dots(B.5)$$

## **B.2 Accounting for Geologic Uncertainty**

Geologic uncertainty is accounted by considering the objective function in **Eq. B.6** as expected value from multiple realizations. The standard deviation is incorporated as a penalized term.

$$f(q) = E[p(\mathbf{q})] + r\sigma[p(\mathbf{q})] \dots\dots\dots(B.6)$$

The variable  $r$  is the risk coefficient that weights the trade-off between the expected value and the standard deviation. A positive  $r$  means that the decision-maker is risk adverse; thus, the optimization will not only minimize the expected value but also minimize variance of arrival time from all realization i.e. all realizations perform equally well. However, this will lower that level of optimality that could be reached. A negative

$r$  means decision-maker is risk prone and has inverse effect from positive  $r$ , while a zero risk coefficient indicates that the decision is risk neutral.

### B.3 Objective Function Minimization

#### *Analytical Sensitivity Calculation*

To minimize the objective function, the sensitivity matrix  $S_{ij}$  is needed to be computed as follows:

$$S_{ij} = \frac{\delta t_{i,m}(\mathbf{q})}{\delta q_j} \dots\dots\dots (B.7)$$

Combining **Eq. 2.2** and **Eq. B.7**  $S_{ij}$  can be written as

$$S_{ij} = \frac{1}{N_{fsl,i}} \sum_{l=1}^{N_{fsl,i}} \frac{\delta \tau_{l,i}(\mathbf{q})}{\delta q_j} \dots\dots\dots (B.8)$$

Using the chain rule, the partial differential term can be written as

$$\frac{\delta \tau_{l,i}(\mathbf{q})}{\delta q_j} = \frac{\delta \tau_{l,i}}{\delta q_{sl,i}} \frac{\delta q_{sl,i}}{\delta q_j} \dots\dots\dots (B.9)$$

The first term  $\frac{\delta \tau_{l,i}}{\delta q_{sl,i}}$  represents the change in time of flight along individual streamlines connected to the producer  $i$  because of changes in the total flowrate along streamlines. **If**

we assume that the streamlines do not shift because of small perturbations in the well rate, then this term can be computed analytically using **Eq. 2.3** and **Eq.2.4** as

$$\frac{\delta \tau_{l,i}}{\delta q_{sl,i}} = \int_{\Sigma} \frac{\delta S(x)}{\delta q_{sl,i}} dx = \int_{\Sigma} \frac{\delta}{\delta q_{sl,i}} \frac{A(x)\phi(x)}{q_{sl,i}} dx = -\frac{\tau_{l,i}}{q_{sl,i}} \dots\dots\dots(\text{B.10})$$

The second term in **Eq. B.9**,  $\frac{\delta q_{sl,i}}{\delta q_j}$ , represents the change in the total production rate along a streamline connected to producer  $i$  because of a change in the total rate of well  $j$ . Recall that well  $j$  can be either a producer or an injector.

*Case 1: j is a Producer*

This term will be vanished for  $i \neq j$  because of the assumption that streamlines do not shift for small perturbation in other producers' well rate. Then we have

$$q_i = N_{sl} q_{sl,i} \dots\dots\dots(\text{B.11})$$

$$\frac{\delta q_{sl,i}}{\delta q_i} = \frac{1}{N_{sl,i}} \dots\dots\dots(\text{B.12})$$

Then the sensitivity from **Eq. B.7** in case  $i = j$  will be defined as

$$S_{ij} = -\frac{t_{i,m}}{q_j}, i = j \dots\dots\dots(\text{B.13})$$

$$S_{ij} = 0, i \neq j \dots\dots\dots(\text{B.14})$$



Case 2:  $j$  is an Injector

With similar assumptions, the sensitivity with respect to injector rate can be written as

$$S_{ij} = -\frac{\sum_{l=1}^{N_{fsl,i,j}} \tau_{l,i,j}}{q_j N_{fsl,i}}, \text{ if } N_{fsl,i,j} = 0 \text{ (has a connection between } i \text{ and } j) \quad \dots\dots\dots(\text{B.15})$$

$$S_{ij} = 0, \text{ if } N_{fsl,i,j} = 0 \text{ (no connection between } i \text{ and } j) \quad \dots\dots\dots(\text{B.16})$$

$N_{fsl,i,j}$  is the number of the fastest streamlines connecting a producer  $i$  to an injector  $j$ .

Similarly,  $\tau_{l,i,j}$  represents Time of Flight of the connected streamlines.

*Jacobian, Gradient and Hessian matrix*

Jacobian matrix

The Jacobian matrix is given by the following expression

$$\mathbf{J} = \nabla_{\mathbf{q}} \mathbf{e} \quad \dots\dots\dots(\text{B.17})$$

where  $\mathbf{e}$  is the vector of arrival time residuals at all producing wells in all groups. Recall from objective function **Eq. 2.1**, we have two terms. For the first term, equalizing arrival time, a single element of  $\mathbf{e}$  corresponding to producer  $i$  in group  $m$  is given by,

$$e_{i,m} = t_{d,m}(\mathbf{q}) - t_{i,m}(\mathbf{q}) \quad \dots\dots\dots(\text{B.18})$$

From **Eq. 2.5**, substitute  $t_{d,m}(\mathbf{q})$

$$e_{i,m} = \frac{1}{N_{prod,m}} \sum_{i=1}^{N_{prod,m}} t_{i,m}(\mathbf{q}) - t_{i,m}(\mathbf{q}) \dots\dots\dots(B.19)$$

The Jacobian matrix becomes

$$J_{ij} = \nabla_{\mathbf{q}} \mathbf{e} = \left[ \frac{1}{N_{Group}} \sum_{i=1}^{N_{Group}} S_{kj} \right] - S_{ij} \dots\dots\dots(B.20)$$

Where  $N_{group}$  is number of wells in a group. Every element in sensitivity matrix S is given by **Eq. B.13** and **Eq. B.14**

For the second term, the production acceleration can be written as

$$e_{i,m} = t_{i,m}(\mathbf{q}) \dots\dots\dots(B.21)$$

The Jacobian matrix becomes

$$J_{ij} = \nabla_{\mathbf{q}} \mathbf{e} = S_{ij} \dots\dots\dots(B.22)$$

Gradient and Hessian

The Gradient and Hessian matrix can be calculated from Jacobian matrix (**J**) and residual vector (**e**) as follows:

$$\mathbf{G} = \mathbf{J}'\mathbf{e} \dots\dots\dots(B.23)$$

$$\mathbf{H} = \mathbf{J}'\mathbf{J} \dots\dots\dots(\text{B.24})$$

*Minimization Algorithm*

To minimize the objective function, we use the sequential quadratic programming (SQP) algorithm which is one of the widely used algorithms for nonlinear constrained optimization. The main concept behind it is to formulate the problem into a series of quadratic programming (QP) sub-problems and iteratively update Gradient and Hessian matrix. In this work, the QP sub-problems are solved with a MATLAB optimization toolbox.

$$\begin{aligned} &\min_{\delta q} \mathbf{G}'\delta q + \frac{1}{2}\delta q'\mathbf{H}\delta q \\ &\text{Subject to} \dots\dots\dots(\text{B.25}) \\ &\mathbf{A}_{eq}\delta q = \mathbf{b}_{eq} \\ &\mathbf{A}\delta q \leq \mathbf{b} \end{aligned}$$

where  $\mathbf{A}_{eq}$ ,  $\mathbf{b}_{eq}$ ,  $\mathbf{A}$  and  $\mathbf{b}$  are matrixes to restrict production rate constraints.

## APPENDIX C

### IMPROVING CALCULATION EFFICIENCY OF GENETIC ALGORITHM

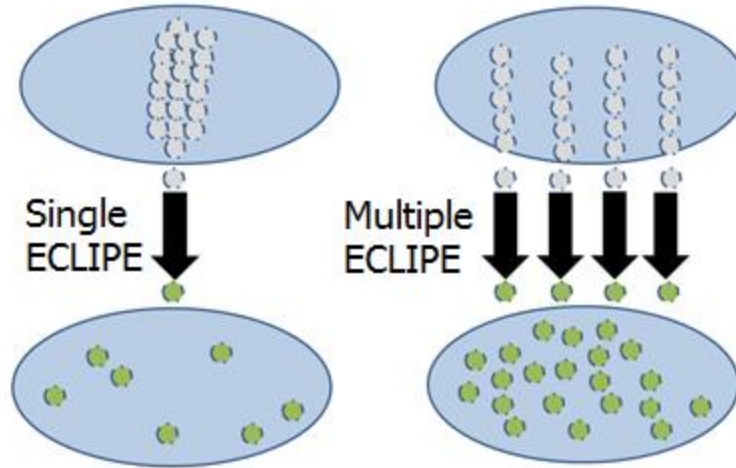
The genetic algorithm is a robust approach to search the global optimal solution. However, one of the disadvantages of GA is computational efficiency. GA requires a large number of simulation runs to update the whole populations. This problem becomes more critical when GA is coupled with reservoir simulation, which requires an expensive calculation, especially for field-scale models. In this work, we implement approaches to speed-up GA calculation, parallel computing and dynamic response surface. The following details briefly discuss about workflow of these approaches.

#### C.1 Parallel Computing

The calculation bottleneck of GA is the step to evaluate objective function. Each population has to queue up for the simulation run sequentially. However, the simulation input and output of each simulation does not relate to each other. In other word, this process can be run separately by parallel computing concept.

##### *Parallel Computing Workflow*

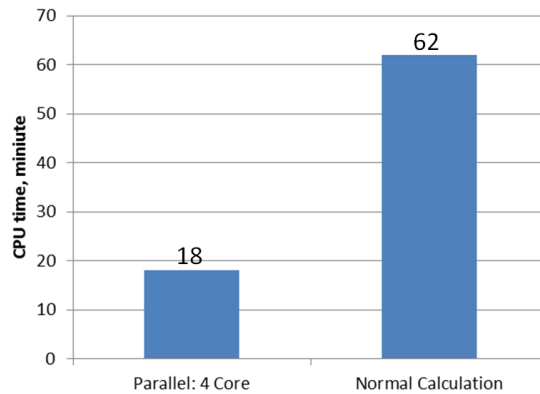
Instead of using a single core to run simulations, the parallel concept uses multiple cores to perform the work simultaneously and combine the results after all populations are evaluated (**Fig. C.1**). In this work, the MATLAB parallel computing toolbox is used with maximum number of cores as four.



**Fig. C.1—Separated worker from parallel computing: multiple simulation can be run simultaneously resulting in more efficient calculation.**

*Illustration of Efficiency*

To demonstrate the benefits of the parallel run with the conventional sequential run, required time to complete 150 simulations were compared (**Fig. C.2**). The parallel run required 18 minutes while the single core used 62 minutes to finish the work which means the parallel can complete the job with 3.4 times faster.



**Fig. C.2—Time required to complete 150 simulations: the parallel computing is 3.4 times faster than the sequential run.**

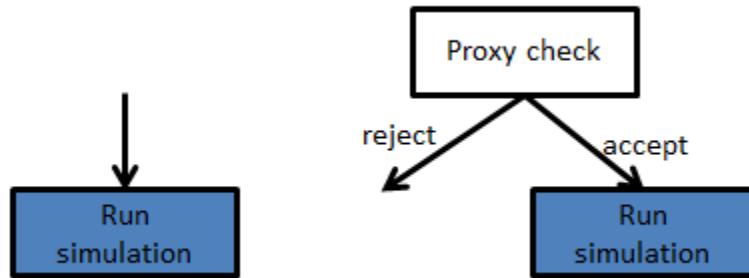
The parallel computing workflow is implemented in this research’s GA code. This implementation helps many following researches to speed-up calculation and allows more number of populations to be evaluated.

### **C.2 Dynamic Response Surface Construction by Kriging Estimation**

Every generation, the GA produces new populations and all of the newly generated population have to be evaluated their fitness.

Generally, all populations in GA require reservoir simulation for objective evaluation. This process can be improved by using the results from early-generation populations to estimate fitness without running simulation. The approximated fitness will be used as pre-conditioner proxy to check whether the new populations require simulation or not. If the proxy response indicates poor fitness and the populations are

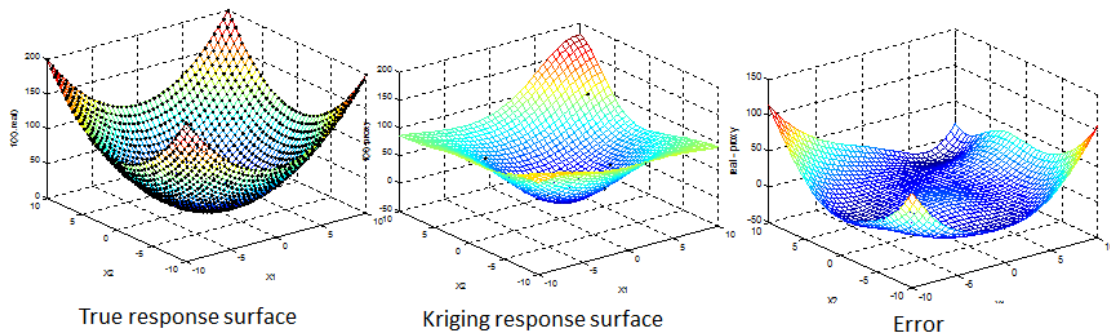
expected to be removed, the populations are rejected without running simulations (**Fig. C.3**).



**Fig. C.3**—Workflow of proxy checks: (left) the conventional GA need to run simulation for all populations, (right) proxy gives feedback whether the population requires simulation run or not.

#### *Response Surface by Kriging Estimation*

In this work, we use response surface constructed by kriging estimation. The kriging estimation is done by the MATLAB toolbox, DACE (Design and Analysis of Computer Experiments). DACE constructs a kriging approximation model based on populations that have been evaluated and use the model to calculate response at unknown points. More number of points near kriging location results in smaller estimation error (**Fig. C.4**).



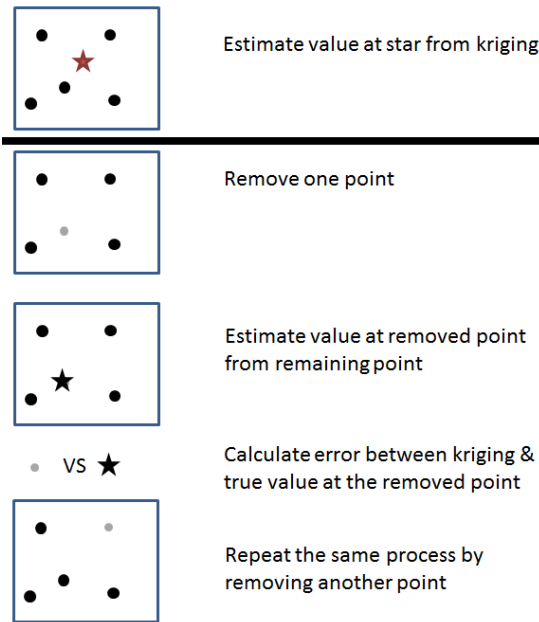
**Fig. C.4—(left) true response, (middle) kriging response and (right) error from kriging estimation: More number of data in central area results in smaller estimation error.**

#### *Tolerance Estimation by Cross Validation*

The results from kriging estimation can only be used as an approximated value since it has error associate with the estimation. Using the result directly is risky as it might lead to wrong rejection. In this study, we use the cross validation concept to estimate error tolerances. The tolerances will be subtracted from the approximated results to ensure the populations are rejected properly.

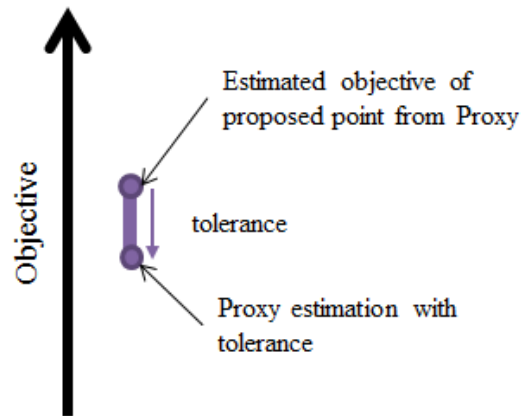
To illustrate this process, **Fig. C.5** shows example of 5 known data points (black dots) to estimate one unknown point (star). First, kriging is used to estimate value of the unknown point. Next, one of the five known points is removed and kriging surface is constructed from the remaining four points. The estimated and the true values of the removed point are compared and the estimated error is stored as tolerance. This process is repeated for every known point in the kriging site.





**Fig. C.5—Workflow of the cross validation: the kriging results and true results from known points are compared and the errors are stored as tolerances.**

The tolerance will be subtracted from fitness to make “the best possible estimation” (**Fig. C.6**). The large tolerance drastically reduces the estimated fitness. Conversely, the small tolerance slightly reduces the estimated fitness. Users can select the level of tolerance based on their certainty on the response surface.

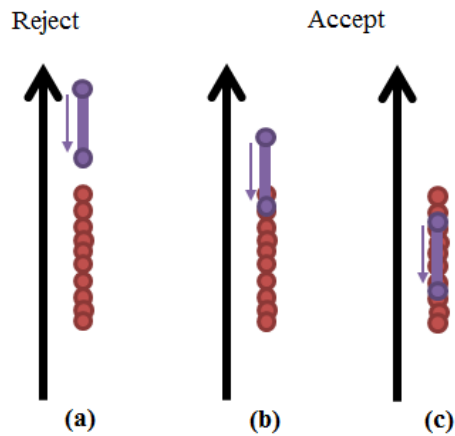


**Fig. C.6—Proxy estimated value will be subtracted by tolerance: larger tolerance drastically reduces the estimated objective value.**

The results cross validation become more accurate with more data. For GA, the amount of known data point increase in later generation which improve the efficiency of this proxy model.

*Acceptance Criteria: Single Objective*

The subtracted value will be compared with the poorest population from the previous generation (the one with the highest value). If it has a larger value than the poorest population, it will be rejected (**Fig. C.7a**). Accepted points for simulation run are required to have smaller value than the poorest population (**Fig. C.7b&c**).



**Fig. C.7—Acceptance criteria to compare with the poorest value from previous generation: (a) the proposed point is rejected if it has the larger value (b & c) the proposed point is accepted if it has the smaller value.**

#### *Acceptance Criteria: Multiple Objectives*

This approach can expand to the multiobjective problem. First, multiple proxies are constructed to estimate each objective and tolerance individually (**Fig C.8**). The proxy results with tolerances will be compared with results from the previous generation by the domination concept instead of the conventional fitness value. The proposed point will be rejected if it has the lowest rank (**Fig C.9a**). If it has a chance to dominate some populations, it will be accepted for the simulation run (**Fig C.9b&c**).

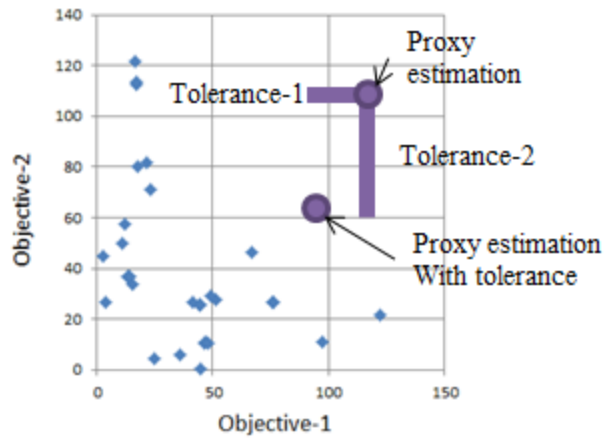


Fig. C.8—Proxy estimated value in multiobjective space. The objective values are subtracted with tolerance for all dimensions.

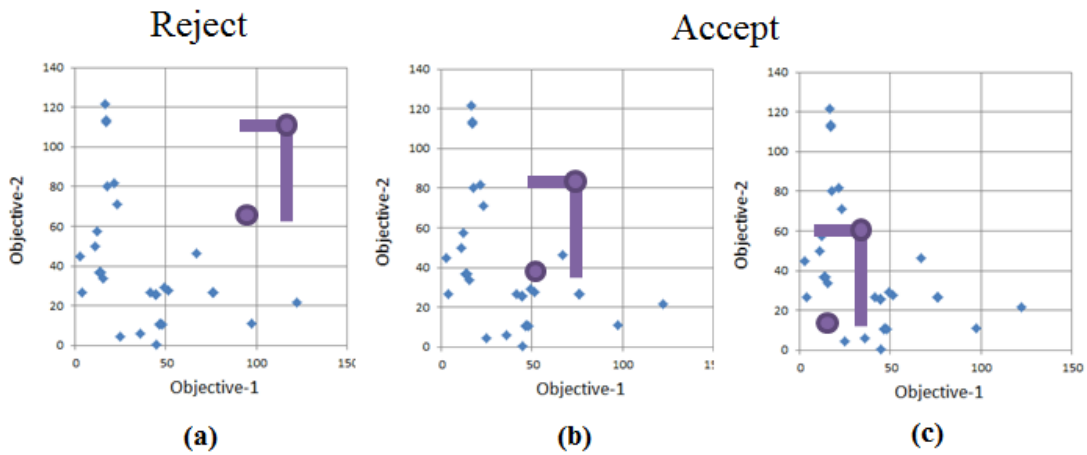


Fig. C.9—Acceptance criteria for the multiobjective problem: the proposed point will be accepted if it dominates some points from the previous generation (b&c)