# Robustness of Multi-Objective Genetic Algorithm in the Optimization of Large Pipeline Networks

**Kamal K. Botros, Ph.D., P.Eng.**
*NOVA Chemicals Corporation*

**Lunch Presentation to:**

**Canadian Operational Research Society**
**Calgary Section**

September 19, 2007

# Published Papers:

1. Botros, K.K., Sennhauser, D.J., Jungowski, K., Poissant, G., Golshan, H. and Stoffregen, J.: “Multi-objective Optimization of Large Pipeline Networks Using Genetic Algorithm”, International Pipeline Conference, Calgary, Alberta, Canada, October 4-8, 2004.
2. Botros, K.K., Sennhauser, D.J., Jungowski, K., Poissant, G., Golshan, H. and Stoffregen, J.: “Effects of Dynamic Penalty Parameters on the Conversion of MOGA in Optimization of a Large Gas Pipeline Network”, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug 30 – 1 September, 2004.
3. Stoffregen, J., Botros, K.K., Sennhauser, D.J., Jungowski, K. and Golshan, H.: “Large Pipeline Network Optimization – A Practical and Technical Assessment”, 37th PSIG Annual Meeting, San Antonio, TX, November 7-9, 2005.
4. Botros, K.K., Sennhauser, D.J., Stoffregen, J., Jungowski, and Golshan, H.: “Large Pipeline Network Optimization – Summary and Conclusions of TransCanada Research Effort”, 6th International Pipeline Conference, Calgary, Alberta, Canada, September 25-29, 2006.

# Outline:

1. Motivation
2. Genetic Algorithm Methodologies
3. Example Application
4. Optimizing the Optimization
5. Conclusions

# Motivation

## Why Optimization in P/L Operation?

1. Reduce hydraulic analysis time through automation.
2. Improve operation.
3. Minimize fuel consumption.

## Three basic objectives:

1. Minimum Fuel consumption.
2. Maximum throughput.
3. Maximum (or desired) linepack.

**1 MW @ 35% Th. Eff.**
**@ $4/GJ**

**=**

**$0.31 million/year of fuel**
**& 4 ktonnes of $CO_2$/year**

# Motivation

| | With Electric | Without Electric |
|---|---|---|
| | MW | MW |
| Alberta | 928.4 | 928.4 |
| Mainline | 2345.9 | 2159.5 |
| TQM | 24.2 | 0 |
| Foothills (incl BC) | 374 | 354.2 |
| Ventures | 4.8 | 4.8 |
| | **3677.3** | **3446.9** |

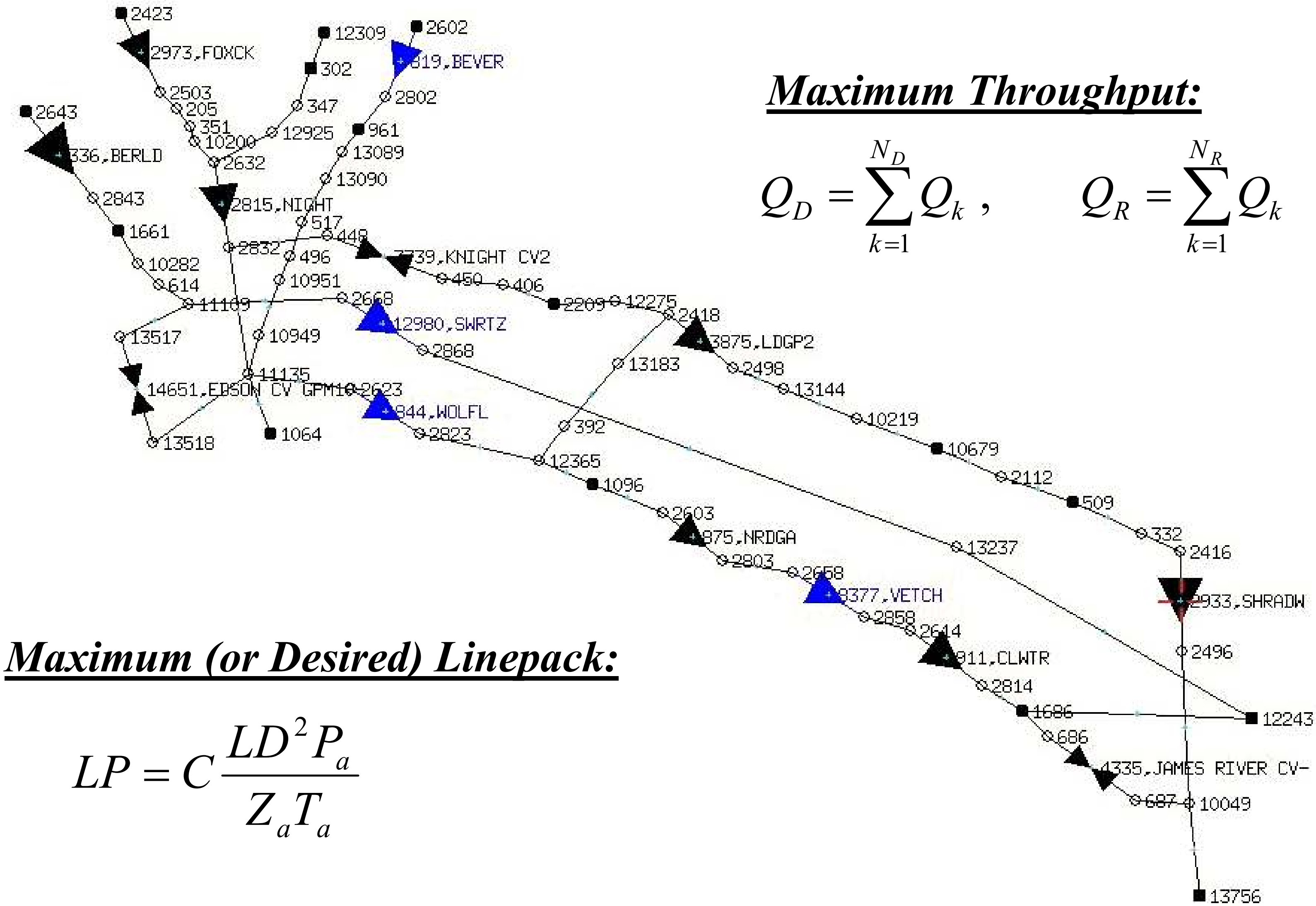
**Cost of Fuel = $1,068 million/year**

**CO2 = 13,790 ktonnes/year**

1% saving
=$10.7 million/year

1% saving
=$137 ktonnes/year

# Motivation




## Maximum Throughput:

$$Q_D = \sum_{k=1}^{N_D} Q_k \ , \qquad Q_R = \sum_{k=1}^{N_R} Q_k$$

## Maximum (or Desired) Linepack:

$$LP = C \frac{LD^2 P_a}{Z_a T_a}$$

# Motivation

## 3. Minimum Fuel Consumption:

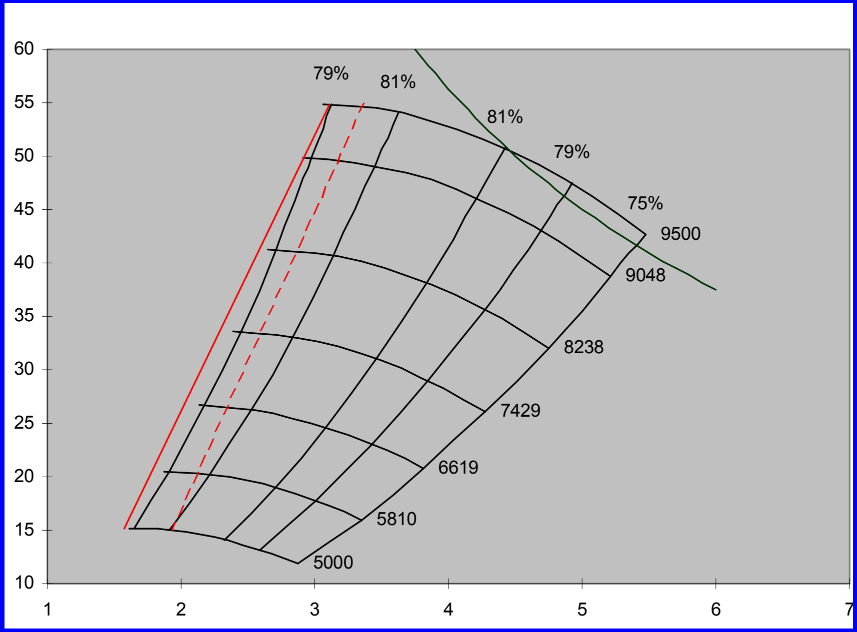
$$g(\dot{m}, P_s, P_d)_r = \alpha \frac{\dot{m}H}{\eta}, \forall(\dot{m}, P_s, P_d) \in D$$

is subject to the following constraints:

1.
$$\left.\begin{array}{c} W_{r,\min} < W < W_{r,\max} \\ N_{r,\min} < N_r < N_{r,\max} \\ Surge < \left(\frac{Q}{N}\right)_r < Stonewall \end{array}\right\} r \in R$$

2. Multi-unit station operation and load sharing strategy

3. Network Hydraulics:
$$P^L \leq P \leq P^U$$
$$A^T P^2 = \phi(\dot{m})$$
$$A\dot{m} = S$$



# Optimization Methodology Selection

Primary Criteria:

- Must find global optimum independent of initialization

Secondary Criteria:

- Multi-objectives
- Robust
- Minimize solution time

Preferred Methodology: Genetic Algorithm (GA)

- Satisfies <u>most</u> criteria

# General Formulation of Optimization Problem

Objective(s)

$$\min\{f(x):\ c_i(x) \le 0, i \in \mathrm{I}, \quad c_i(x) = 0, i \in \mathrm{E}\}$$

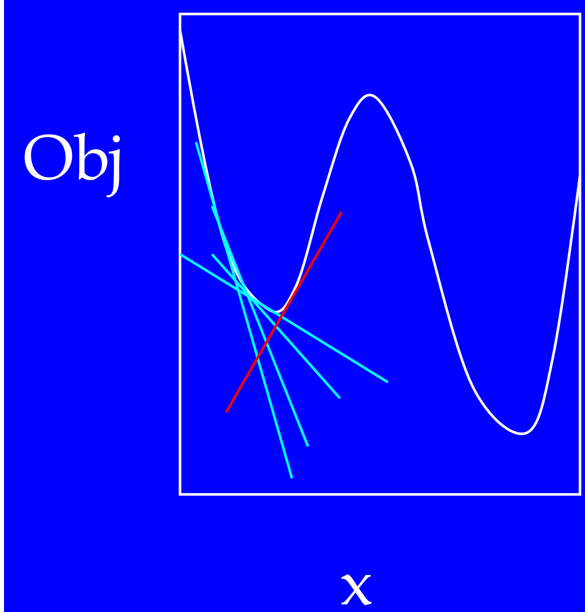
Inequality constraints

Equality constraints

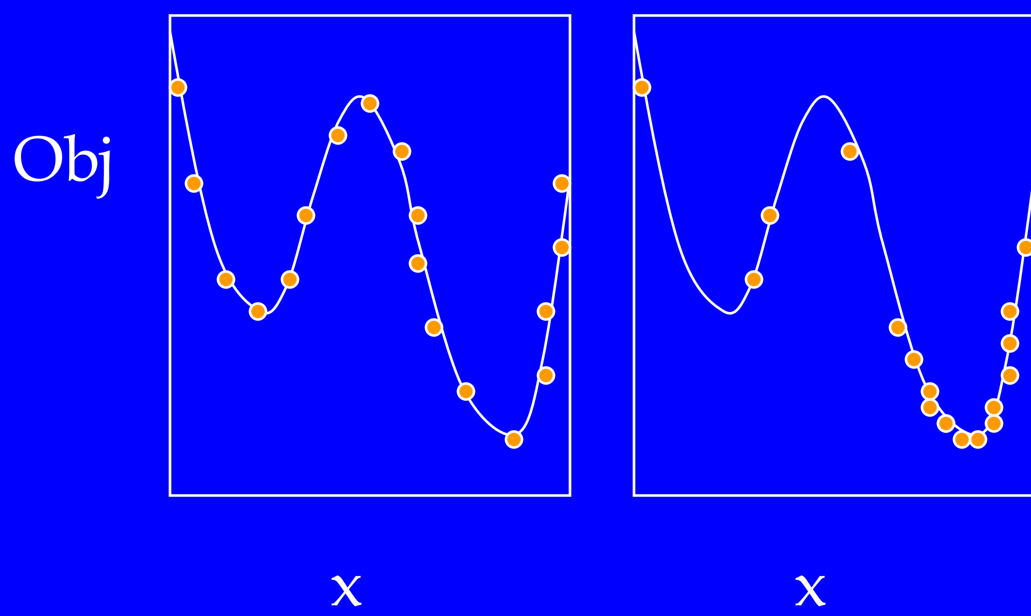
Control Variables
(Decision Variable)

# Classification of Optimization Method

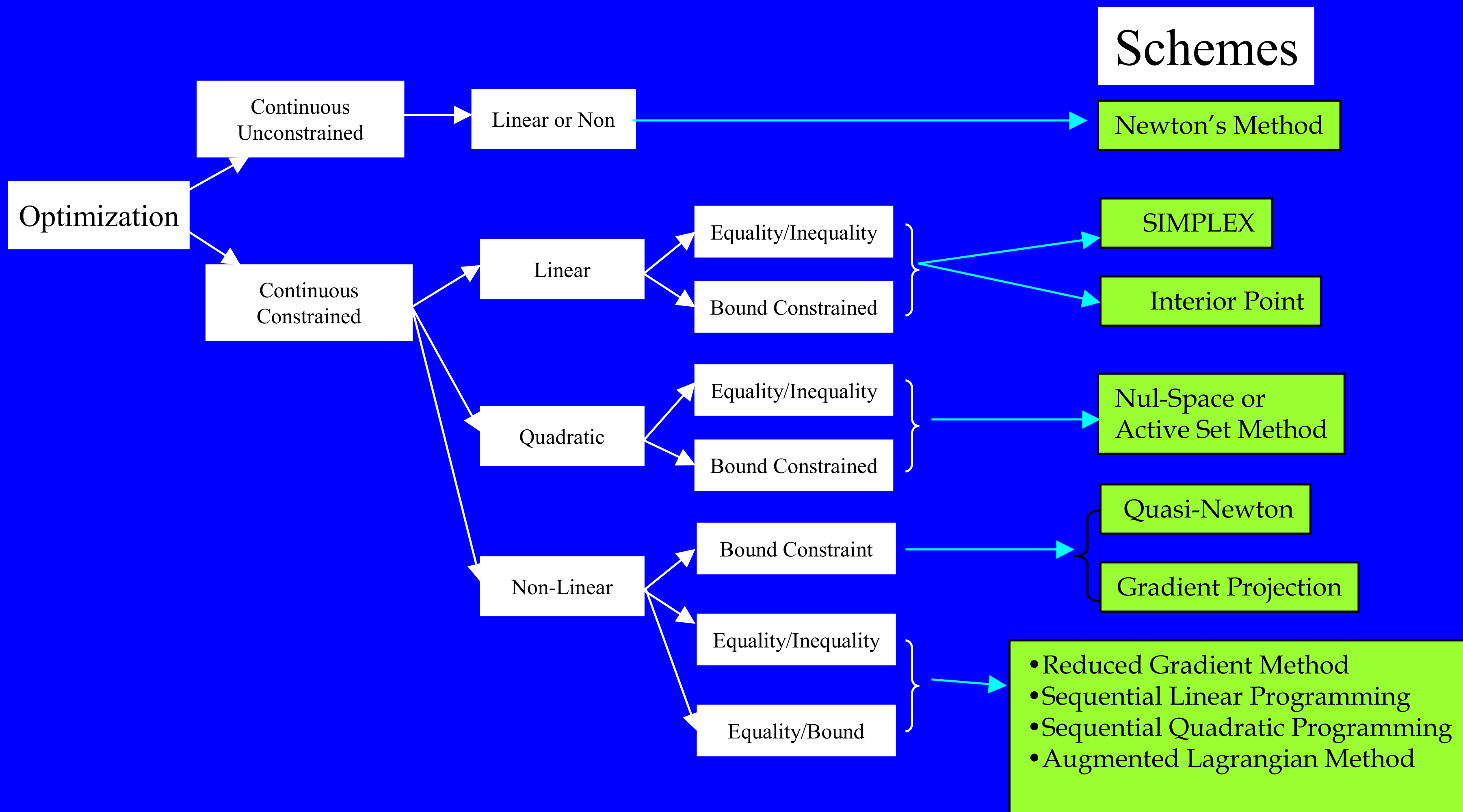
1. Gradient Based Methods.




2. Heuristic (Stochastic or Evolutionary) Methods



# Gradient-Based Optimization Chart



# Advantages and Disadvantages of Gradient-Based Optimization

## Advantages:

- Less number of **function calls**.
- **Converges quickly** if the active set is near local minima.
- **Accurately** finds local minima.
- Less computer **time.**
- Numerous Constraint handling Techniques --> **flexibility in constraint handling.**

## Disadvantages:

- Depends on the **Start Point**.
- Often trapped in **Local Minima** - Not suitable for finding global minima.
- First-order, second order **derivatives**, excessive matrix manipulation
- Not suitable for **large systems**.
- Not suitable for **multio-bjective** optimization.
- Not suitable for **multidisciplinary** problem.

# Fundamentals of Genetic Algorithms

- Algorithm based on natural evolution.
- Form of "**survival of the fittest**".
- GA search from a **population** of points, not a single point.
- GA uses **Fitness** function, not the objective function itself.
- GA uses **probabilistic** transition rules, not deterministic rules.
- GA works with a **coding** of the parameter set, not the parameters themselves.

# Genetic Algorithms

Advantages:

- Good for **global** optima and does not get trapped in local minima.
- **Gradient free**.
- Excellent for Multidisciplinary and **multi-objectives**.
- Does not care about **size of the system**.
- Suitable for **parallelization**.

Disadvantages:

- Large number of **function calls**.
- long computer **time** if parallelization is not used .
- May depend on internal **GA parameters** (crossover, mutation, ranking, selection, fitness function, and constraint handling).

# How Genetic Algorithm Works !

# Fundamentals of Genetic Algorithms

## Binary Encoding

Example: Pressure (Pmin=5000 kPa , Pmax=6000 kPa)

11010

Decoded Value = 26

$$x_i = x_i^{\min} + \frac{x_i^{\max} - x_i^{\min}}{2^{l_i}} DV\left(s_i\right)$$

Pressure Value = 5000 + (6000-5000)x26/$2^5$
=5812.5 kPa

# Fundamentals of Genetic Algorithms

Binary Encoding

Design Case (pipeline decision variables)

1011001011001010111001010101011001110

Q Pd1 Pd2 N1 N2 V% ST

# Fundamentals of Genetic Algorithms

Design Case Chromosome:

101100101 100101011 100101010 101100111 00111
0110101000111 1001100 100111101 0101 101101011
10101011 100111011 010101011110111011001101
10111011101110100001110001 10………

# Example of Decision Variable (Table 1 of 3)

| Description | Min | Max | Resolution | N | String Length (bits) |
|---|---|---|---|---|---|
| Proration Factor | 900 | 1100 | 1 | 201 | 8 |
| DPC at Saddle Hills CV (kPa) | 5000 | 8450 | 10 | 346 | 9 |
| DPC at Gold Creek CV (kPa) | 5865 | 7245 | 10 | 139 | 8 |
| DPC at Knight CV (kPa) | 4685 | 6175 | 10 | 150 | 8 |
| Zama Lake #3 MS (kPa) | 5000 | 8450 | 10 | 346 | 9 |
| DPC at Zama Lake CV (kPa) | 3205 | 5655 | 10 | 246 | 8 |
| DPC at Edson CV (kPa) | 4005 | 6205 | 10 | 221 | 8 |
| Meikle River Status | 0 | 1 | 1 | 2 | 1 |
| SPC node at Meikle River (kPa) | 3200 | 4430 | 10 | 124 | 7 |
| Valleyview Status | 0 | 1 | 1 | 2 | 1 |
| SPC node at Valleyview (kPa) | 3200 | 4430 | 10 | 124 | 7 |
| Gold Creek B Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Gold Creek B (kPa) | 5865 | 8275 | 10 | 242 | 8 |
| Berland River Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Berland River (kPa) | 5865 | 8275 | 10 | 242 | 8 |
| Beaver Creek Status | 0 | 1 | 1 | 2 | 1 |
| SPC node at Beaver Creek (kPa) | 3800 | 6040 | 10 | 225 | 8 |
| Wolf Lake Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Wolf Lake (kPa) | 4950 | 6450 | 10 | 151 | 8 |

# Example of Decision Variable (Table 2 of 3)

| Description | Min | Max | Resolution | N | String Length (bits) |
|---|---|---|---|---|---|
| Nordegg Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Nordegg (kPa) | 4950 | 6450 | 10 | 151 | 8 |
| Clearwater Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Clearwater (kPa) | 4950 | 6450 | 10 | 151 | 8 |
| Knight Status | 0 | 1 | 1 | 2 | 1 |
| SPC node at Knight (kPa) | 3200 | 5050 | 10 | 186 | 8 |
| Schrader Creek Status | 0 | 1 | 1 | 2 | 1 |
| SPC node at Schrader Creek (kPa) | 3450 | 4640 | 10 | 120 | 7 |
| Saddle Hills Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Saddle Hills (kPa) | 5000 | 8450 | 10 | 346 | 9 |
| Clarkson Valley Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Clarkson Valley (kPa) | 5045 | 6895 | 10 | 186 | 8 |
| Fox Creek Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Fox Creek (kPa) | 4685 | 6165 | 10 | 149 | 8 |
| Lodgepole Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Lodgepole (kPa) | 4685 | 6175 | 10 | 150 | 8 |
| Vetchland Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Vetchland (kPa) | 4950 | 6450 | 10 | 151 | 8 |
| Dryden Creek Status | 0 | 1 | 1 | 2 | 1 |
| SPC node at Dryden Creek (kPa) | 3200 | 4430 | 10 | 124 | 7 |

# Example of Decision Variable (Table 3 of 3)

| Description | Min | Max | Resolution | N | String Length (bits) |
|---|---|---|---|---|---|
| Latornell Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Latornell (kPa) | 5865 | 8275 | 10 | 242 | 8 |
| Swartz Creek Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Swartz Creek (kPa) | 6290 | 8690 | 10 | 241 | 8 |
| Pipestone Creek Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Pipestone Creek (kPa) | 5000 | 8450 | 10 | 346 | 9 |
| Hidden Lake Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Hidden Lake (kPa) | 5000 | 8450 | 10 | 346 | 9 |
| Meikle River B Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Meikle River B (kPa) | 5000 | 8450 | 10 | 346 | 9 |
| Alces River B Status | 0 | 1 | 1 | 2 | 1 |
| DPC node at Alecs River B (kPa) | 5000 | 8450 | 10 | 346 | 9 |
| Vetchland Suction BV Status | 0 | 1 | 1 | 2 | 1 |
| Edson CV Bypass BV Status | 0 | 1 | 1 | 2 | 1 |
| Meikle River Discharge BV Status | 0 | 1 | 1 | 2 | 1 |
| | | | | **Total string length:** | **260** |
| | | | | **Total possible cases:** | **1.85267E+78** |

# Fundamentals of Genetic Algorithms

**Step #1:** → Initial Population of the entire Design Space

- Use of **DOE** to populate the entire design space randomly or in an organized, analytical fashion.

- Population size is a very small fraction of the total number of cases: e.g. Upstream James River has 1.85E+78 cases is solved with only 1000 population.

***Rules of Thumb:***

❑ No. of population ≥ 50 times No. of decision variables.

# Fundamentals of Genetic Algorithms

**Step #2:** → Assign Fitness Value

This is the most challenging and diverse part of GA.
Vast Literature on Constraint handling in the fitness function;
for example: *Exterior Penalty Function*:

Penalty Vector

$$f(x) = Obj(x) + r\sum_{i=1}^{K}\left|<\phi_i>\right| + r\sum_{j=1}^{L}\left|\psi_j\right|$$

*Fitness* — *Penalty* — *Penalty*

$$\phi_i(x) \geq 0 \qquad i = 1, K$$
$$\psi_j = 0 \qquad j = 1, L$$

$$<\alpha> = \alpha \qquad if \quad \alpha \leq 0$$
$$<\alpha> = 0 \qquad if \quad \alpha > 0$$

# Fundamentals of Genetic Algorithms

## Other Constraint Handling Strategies:

**1. Fiacco and McCormick (Exterior and Interior):**

$$f(x) = Obj(x) + r^2 \sum_{i=1}^{K} \frac{1}{\phi_i^S} + \frac{1}{r} \sum_{i=1}^{K} \left| < \phi_i > \right|^2 + \frac{1}{r} \sum_{j=1}^{L} \left| \psi_j \right|^2$$

**2. Powells (Exterior and Distorted Interior):**

$$f(x) = Obj(x) + r \sum_{i=1}^{K} < \phi_i - \sigma_i >^2 + r \sum_{j=1}^{L} (\psi_j - \tau_j)^2$$

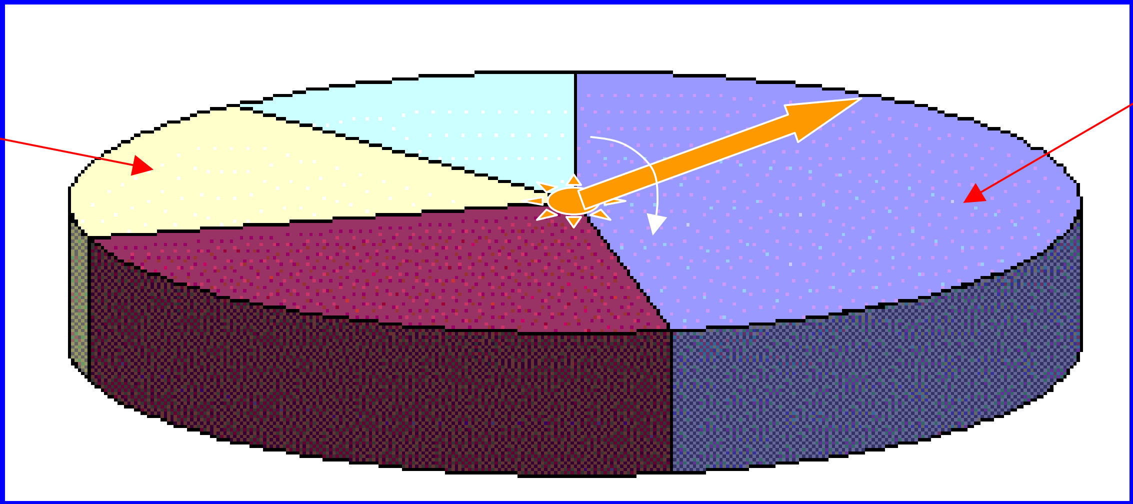
**3. Schuldt et al (Exterior and Distorted Interior):**

$$f(x) = Obj(x) + r \sum_{i=1}^{K} [< \phi_i - \sigma_i >^2 - \sigma_i^2] + r \sum_{j=1}^{L} [(\psi_j - \tau_j)^2 - \tau_j^2]$$

# Fundamentals of Genetic Algorithms

**Step #4:** → Parent Selection

Low Fitness Parents



High Fitness Parents

Roulette Wheel Selection (among others)

# Fundamentals of Genetic Algorithms

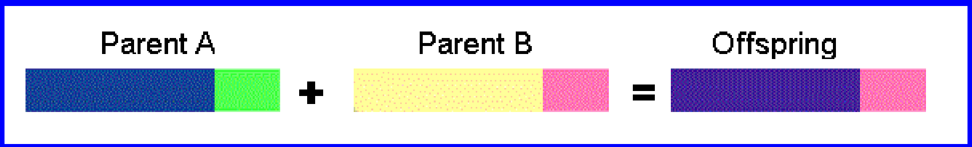
**Step #5:** → Crossover

| | |
|---|---|
| Chromosome 1 | 11011 \| 00100110110 |
| Chromosome 2 | 11011 \| 11000011110 |
| Offspring 1 | 11011 \| 11000011110 |
| Offspring 2 | 11011 \| 00100110110 |

# Fundamentals of Genetic Algorithms

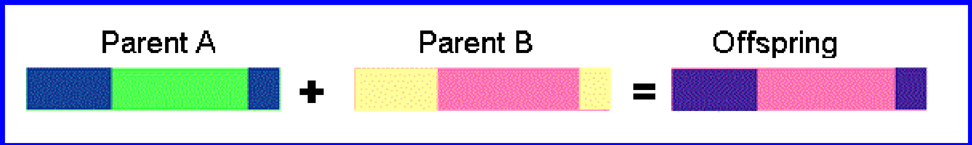
**Step #5:** → Crossover

**Single point crossover**

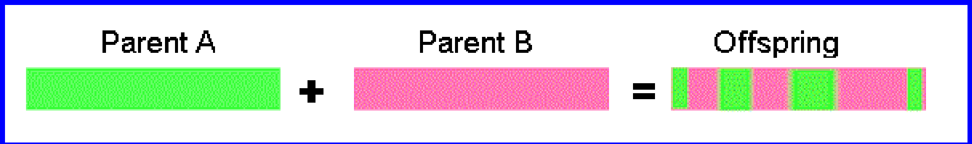



11001011+11011111 = 11001111

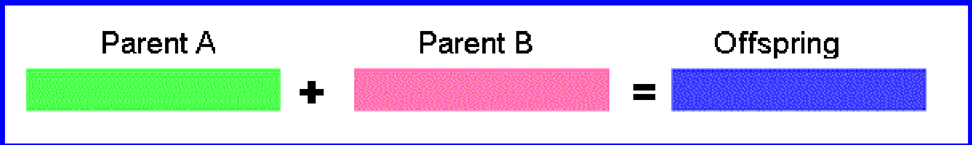
**Two point crossover**




**Uniform crossover**




**Arithmetic crossover**



# Fundamentals of Genetic Algorithms

**Step #6:** → Mutation

| | |
|---|---|
| Original offspring 1 | 1101111000011110 |
| Original offspring 2 | 1101100100110110 |
| Mutated offspring 1 | 1100111000011110 |
| Mutated offspring 2 | 1101101100110110 |

- *Mutation of a portion of offsprings is important to maintain the search space open.*

# Fundamentals of Genetic Algorithms

**Step #7:** → Elitism

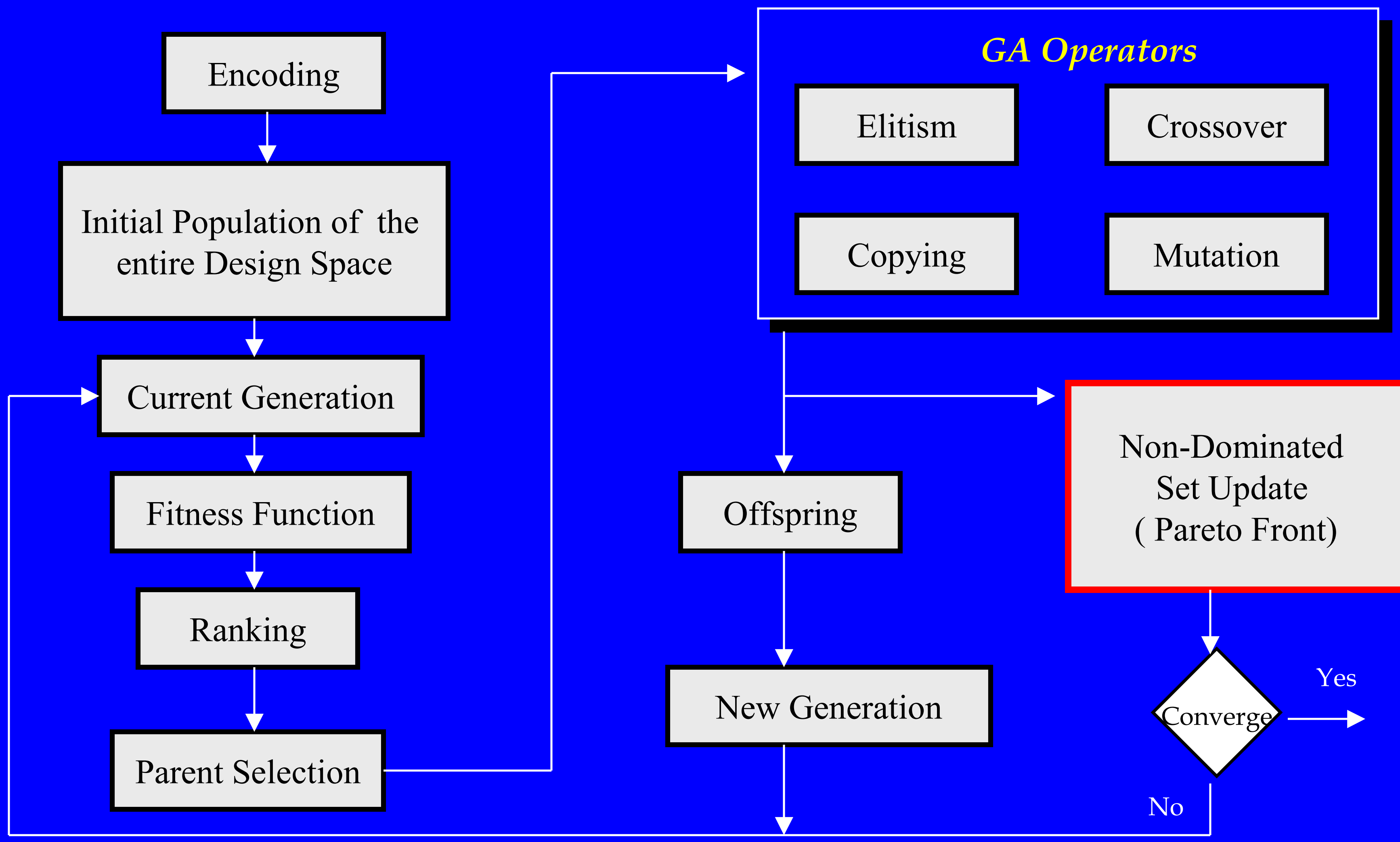
- A portion of best parents are copied directly into the new generation.
- Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution.

# Fundamentals of Genetic Algorithms

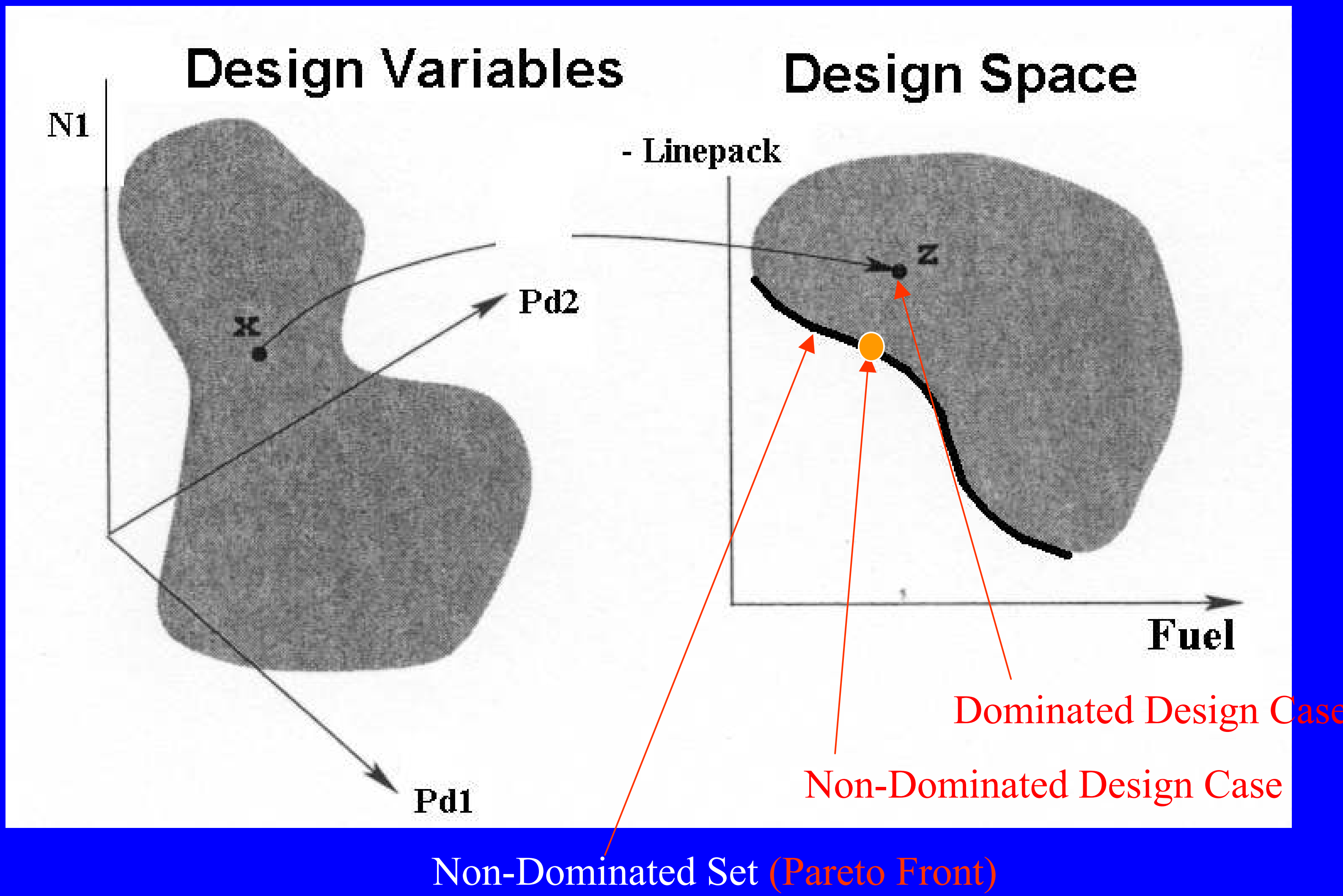
**Step #8:** → Copying

- Random copying of parents regardless of fitness into the new generation is important to maintain the search space open.

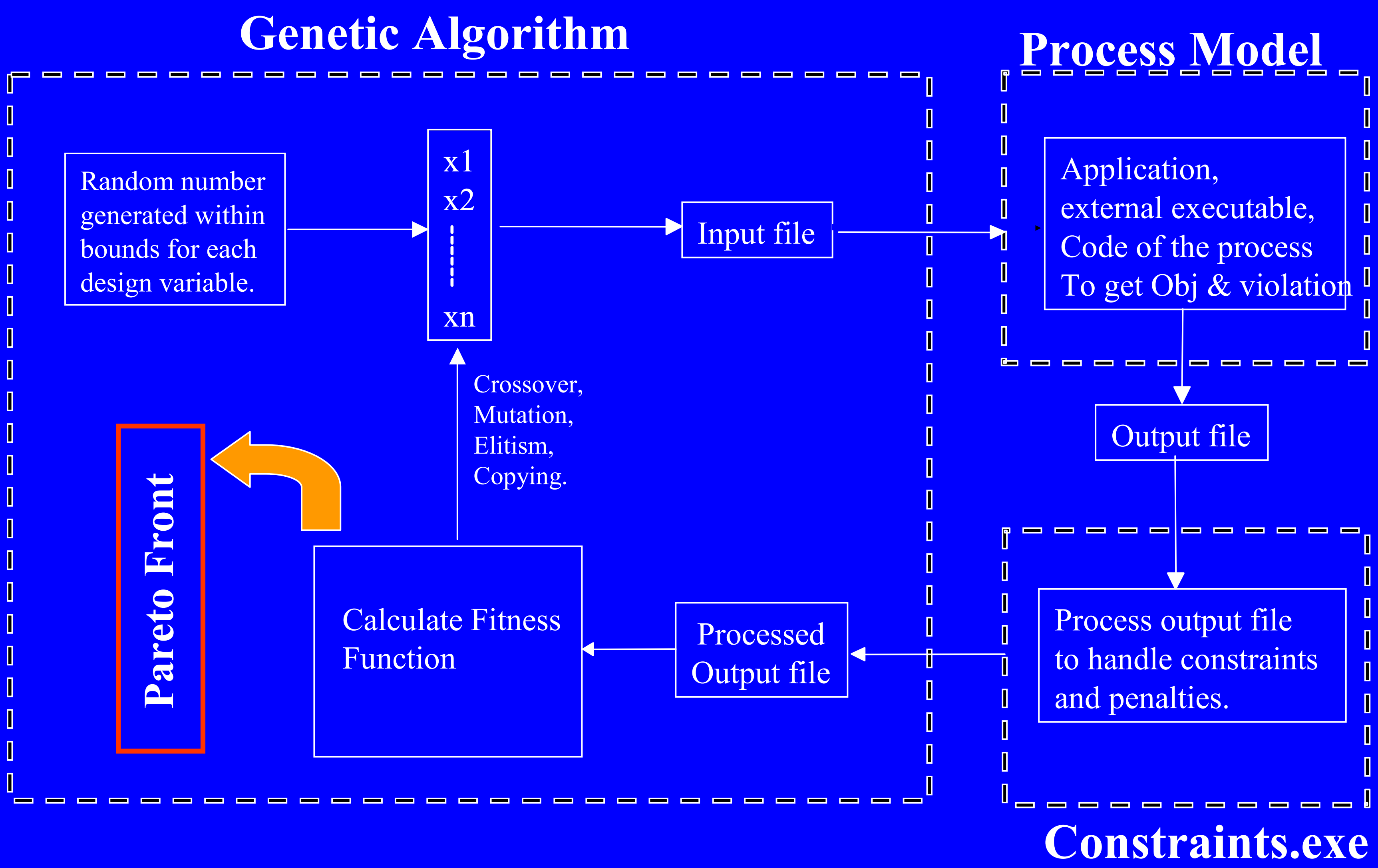
# Fundamentals of Genetic Algorithms



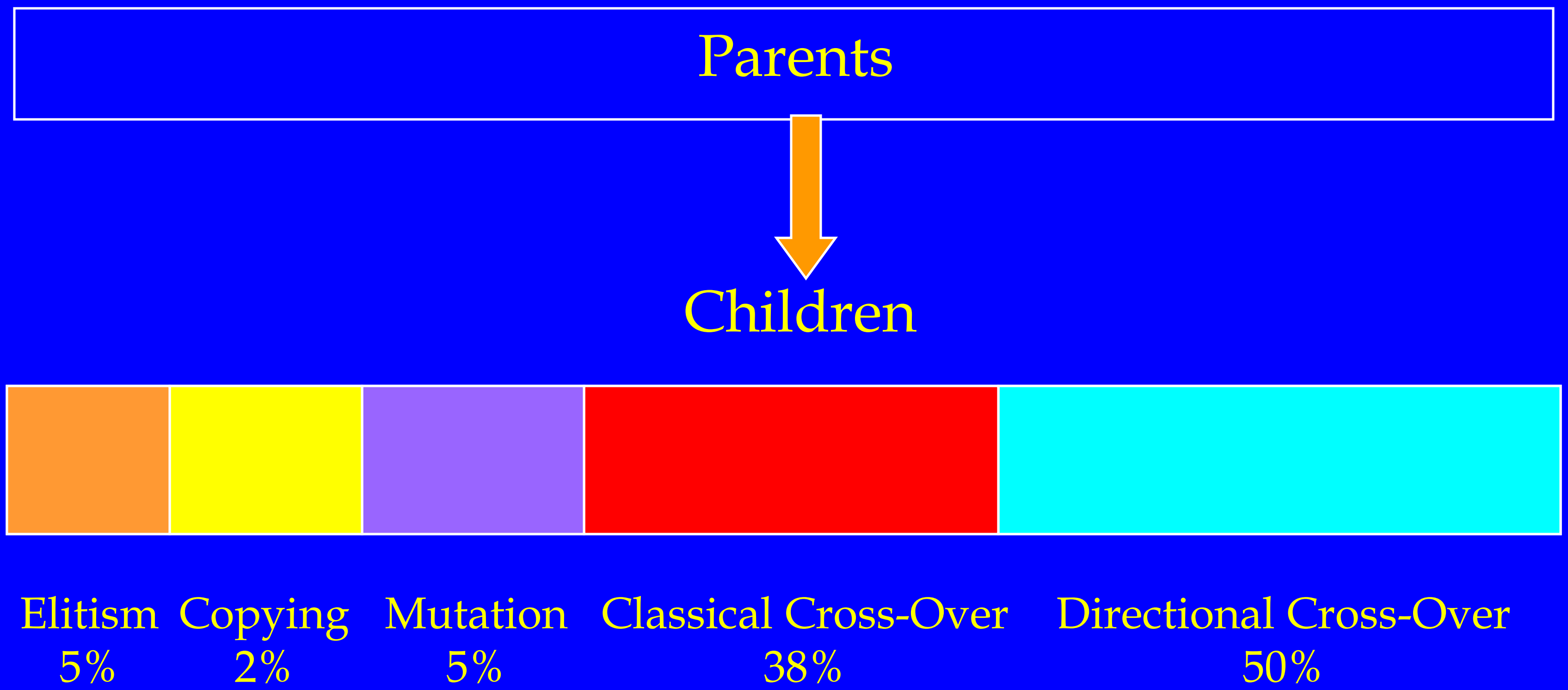
# Fundamentals of Genetic Algorithms



# Simulation Methodology

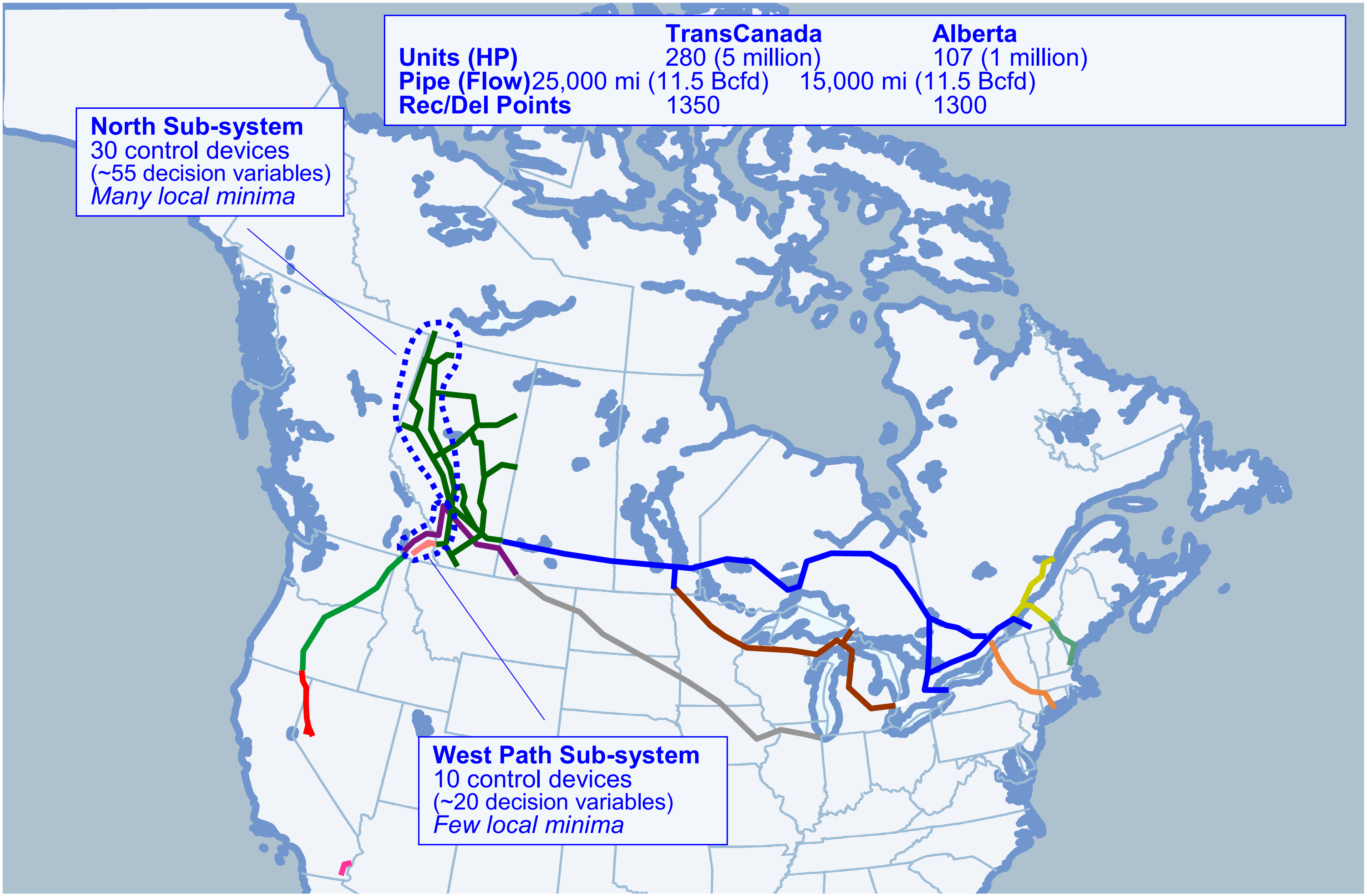


# MOGA Parameters



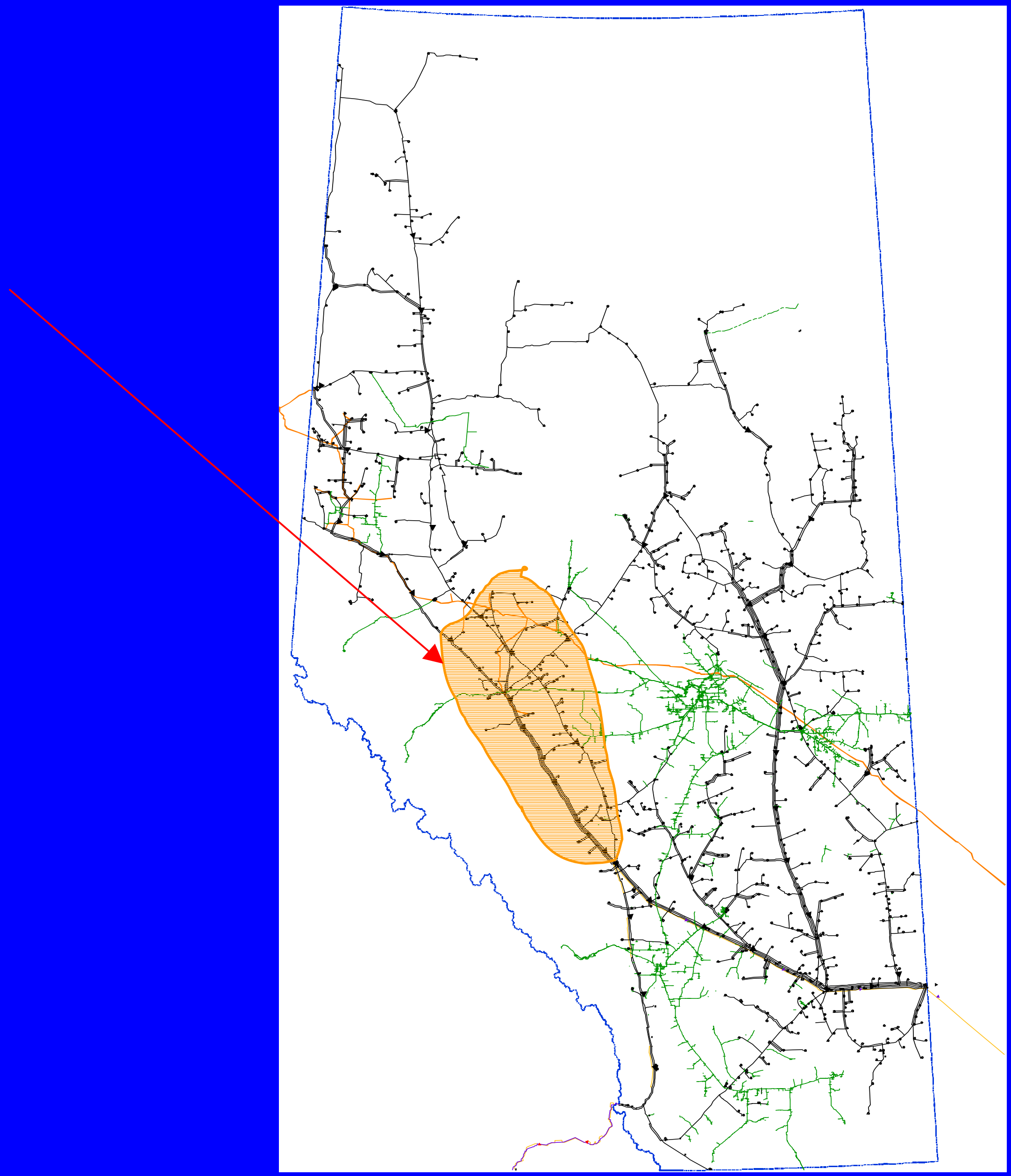
# Three Examples

# of Single and Multi-Objective Optimizations

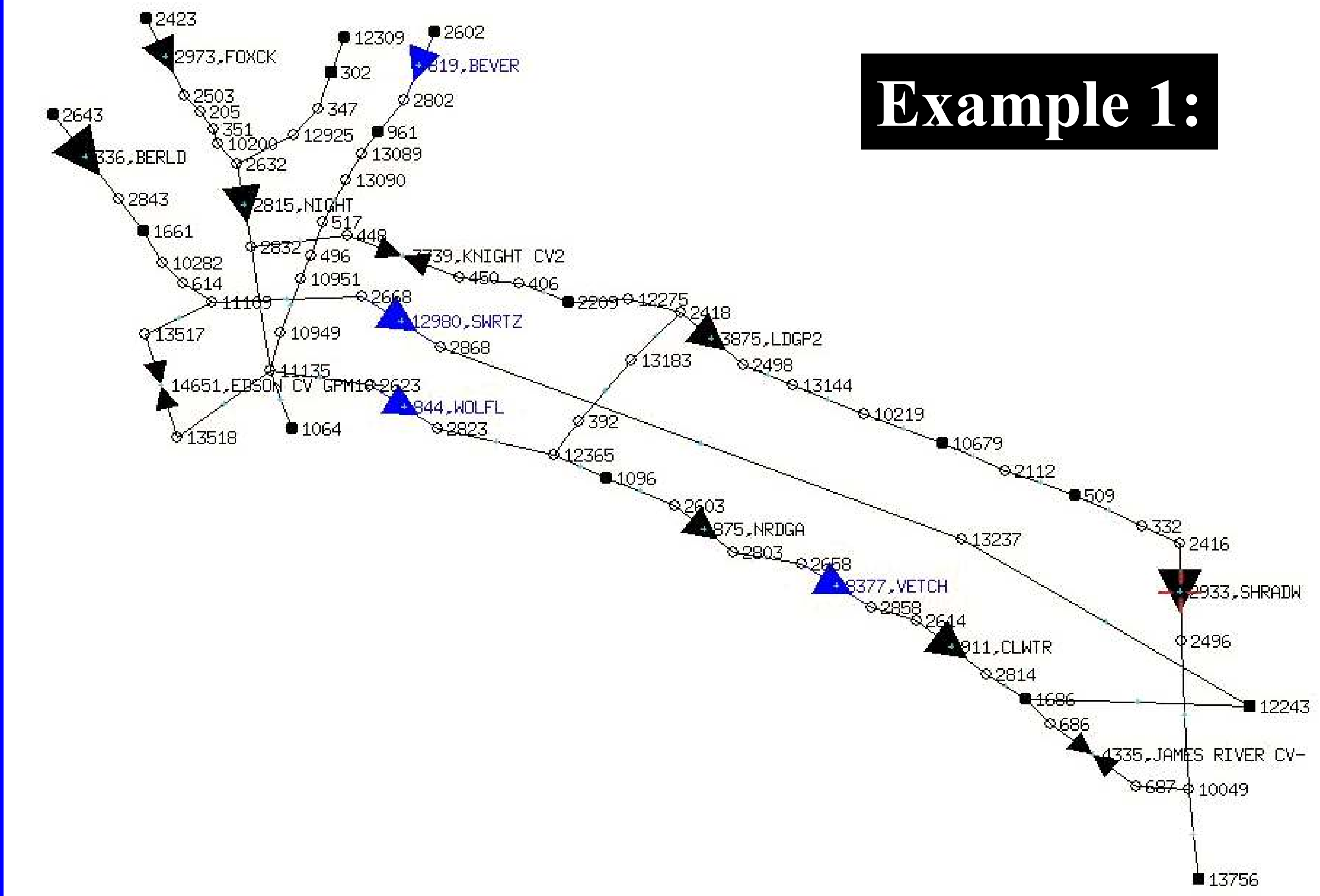



# Alberta System Optimization – Study Areas

# Example 1:



# Example 1:


2423
2973,FOXCK
12309
302
2602
819,BEVER
2643
336,BERLD
2815,NIGHT
7739,KNIGHT CV2
12980,SWRTZ
3875,LDGP2
14651,EDSON CV GPM10
844,WOLFL
875,NRDGA
8377,VETCH
911,CLWTR
2933,SHRADW
4335,JAMES RIVER CV-
12243
13756
10049
1064
13518
13517

# Decision Variables

- 1 flow variable
- 2 control valve node pressures
- **8 compressor** station discharge pressures
- 8 compressor station statuses (on/off)
- 1 block valve status (open/closed)

- Total of **20** decision variables

# Constraint Handling

- Constraint penalty parameters:

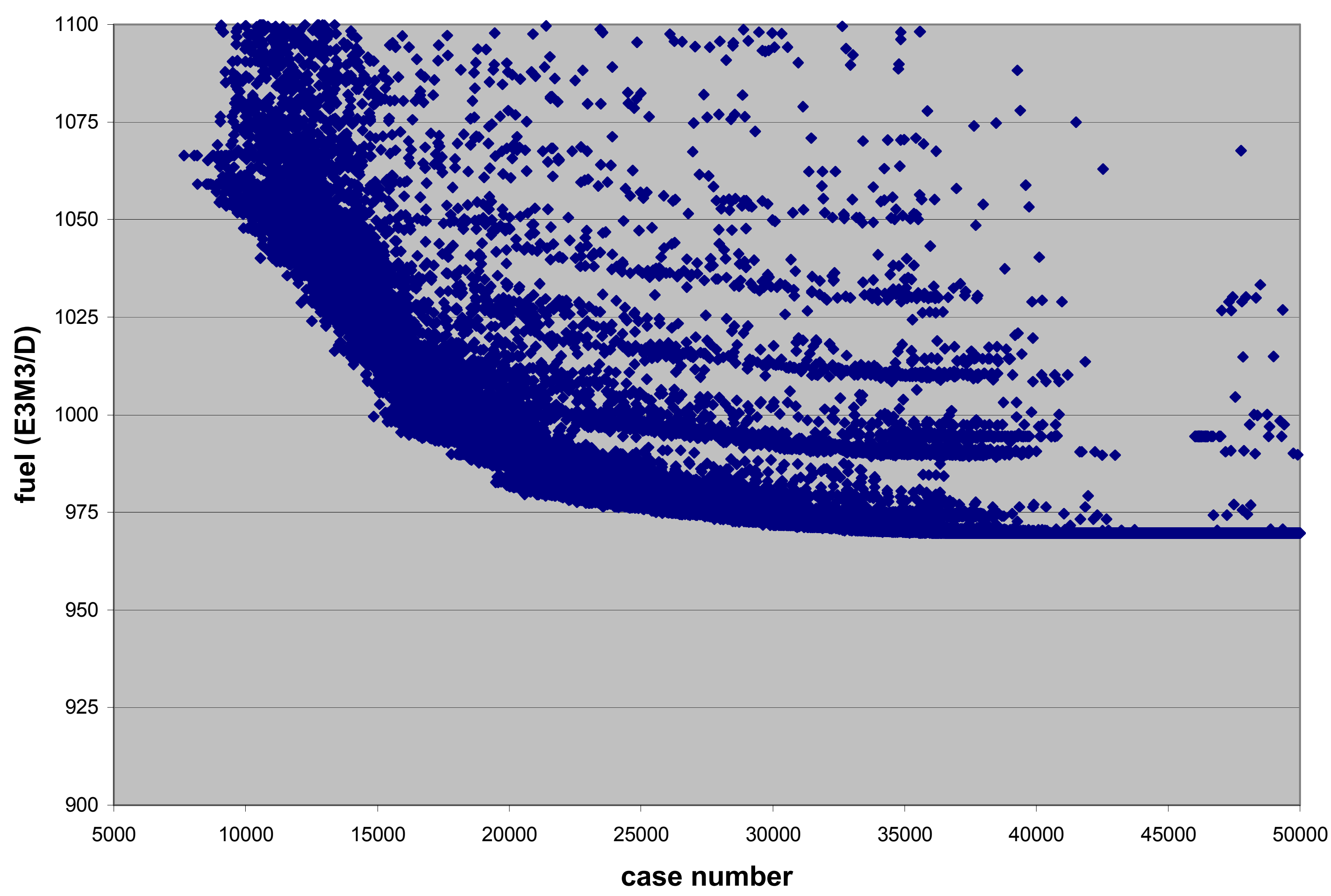
| Objective | Penalty Parameter Matrix pressure* | power* | control valve* | reverse flow** | reverse pressure*** | linepack equality* | run status error |
|---|---|---|---|---|---|---|---|
| Fuel Consumption (1000 $m^3$/d) | 10 | 10 | 10 | 1,000 | 10 | 5 | +1000 |
| Linepack (1000 $m^3$) | -1,000 | -1,000 | -1,000 | -100,000 | -1,000 | -500 | -10000 |
| Throughput (1000 $m^3$/d) | -1,250 | -1,250 | -1,250 | -125,000 | -1,250 | -625 | -100000 |

*: Units are units of objective per one percent of violation

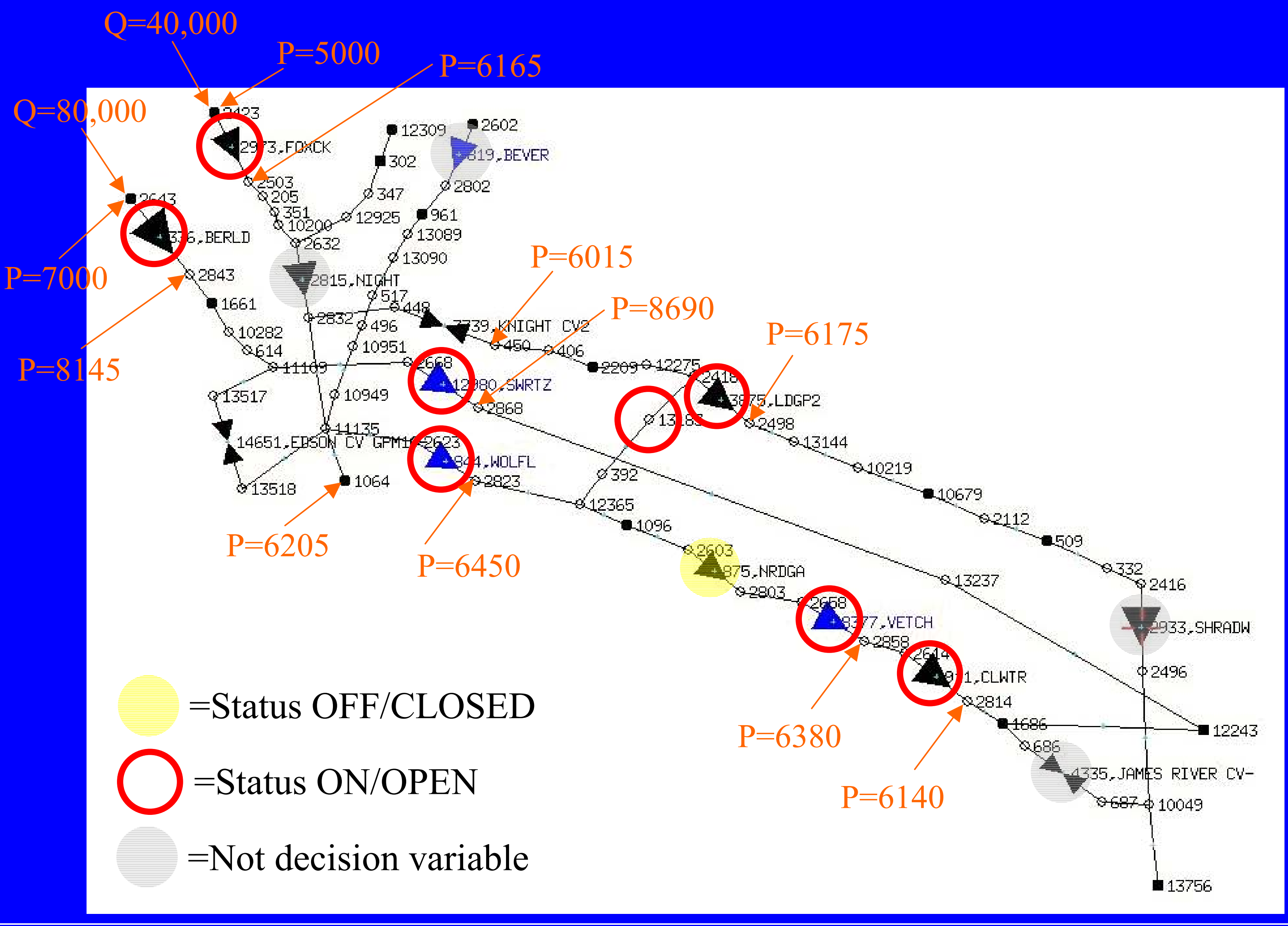
**: Units are units of objective per one reverse flow situation

***: Units are units of objective per one percent of violation of equation shown:

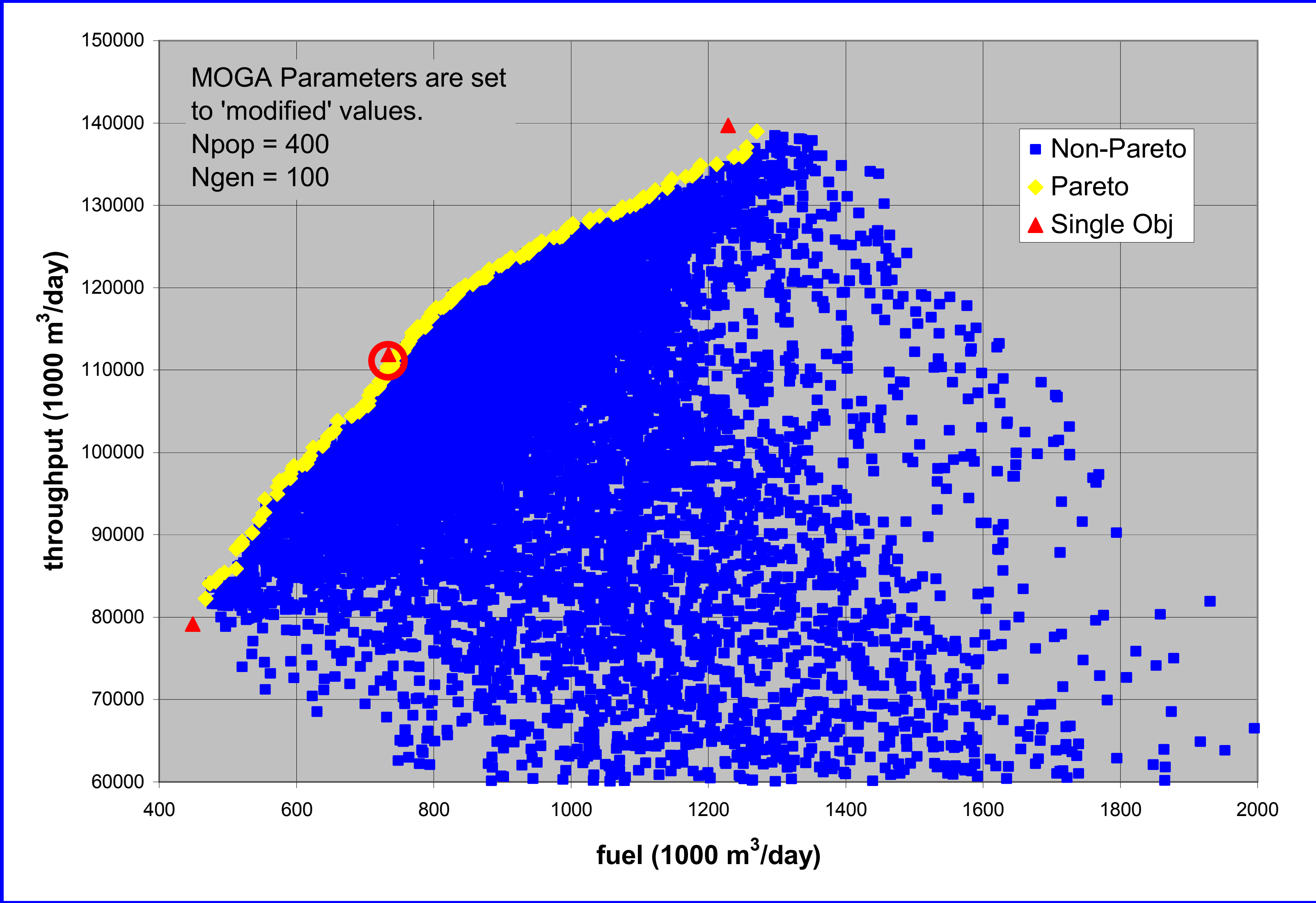
# Single Objective



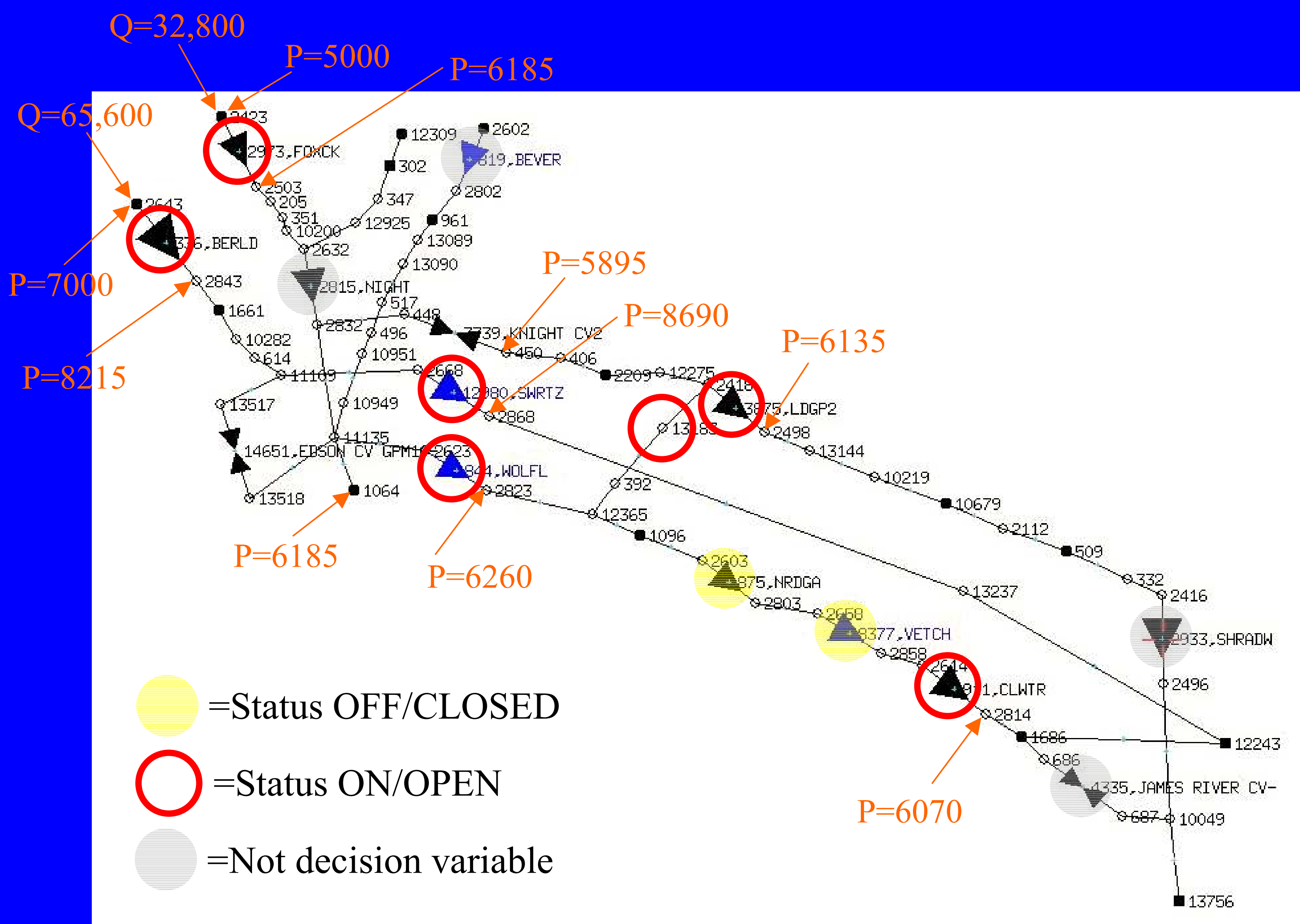
# Single Objective



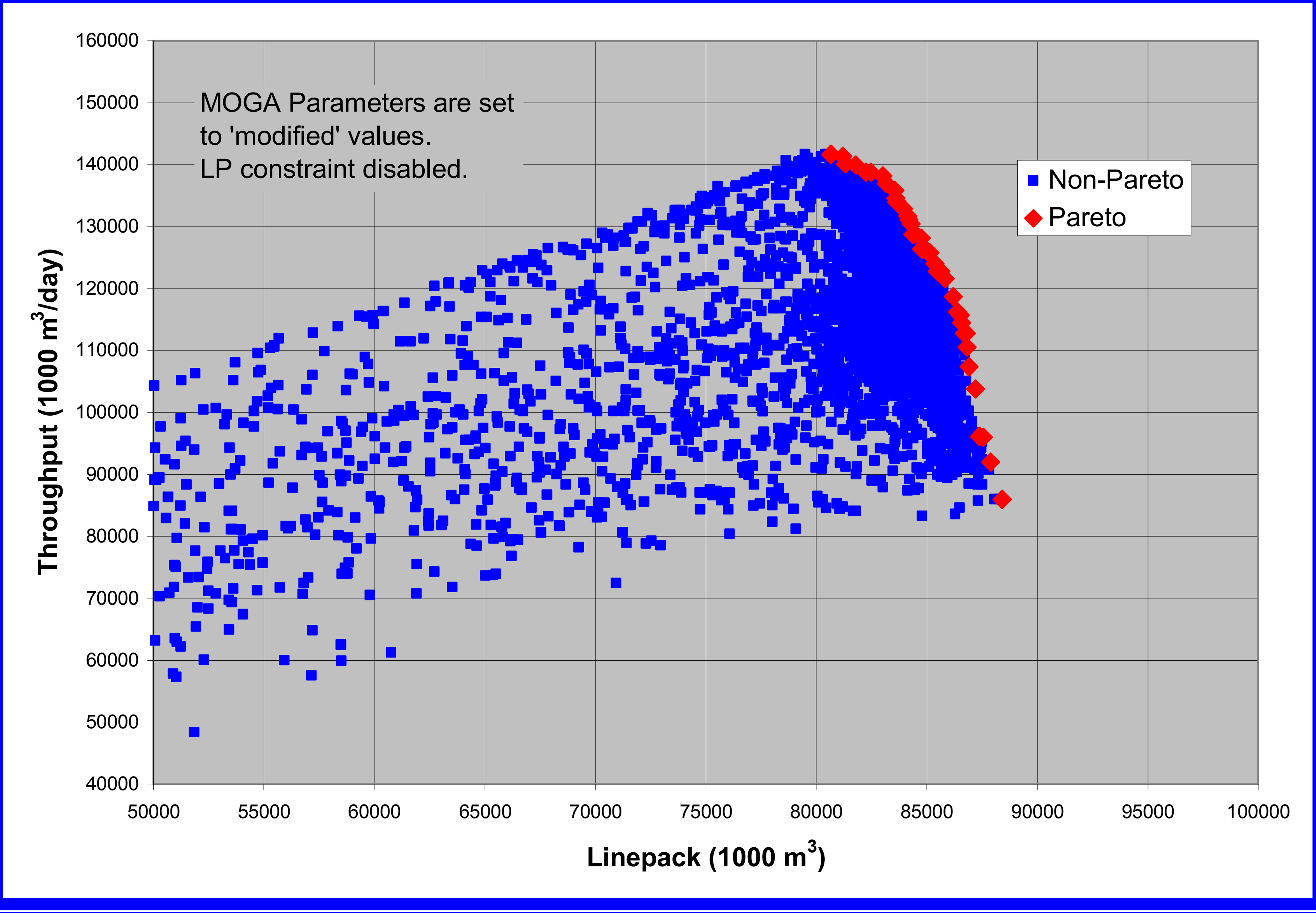
# Two Objectives



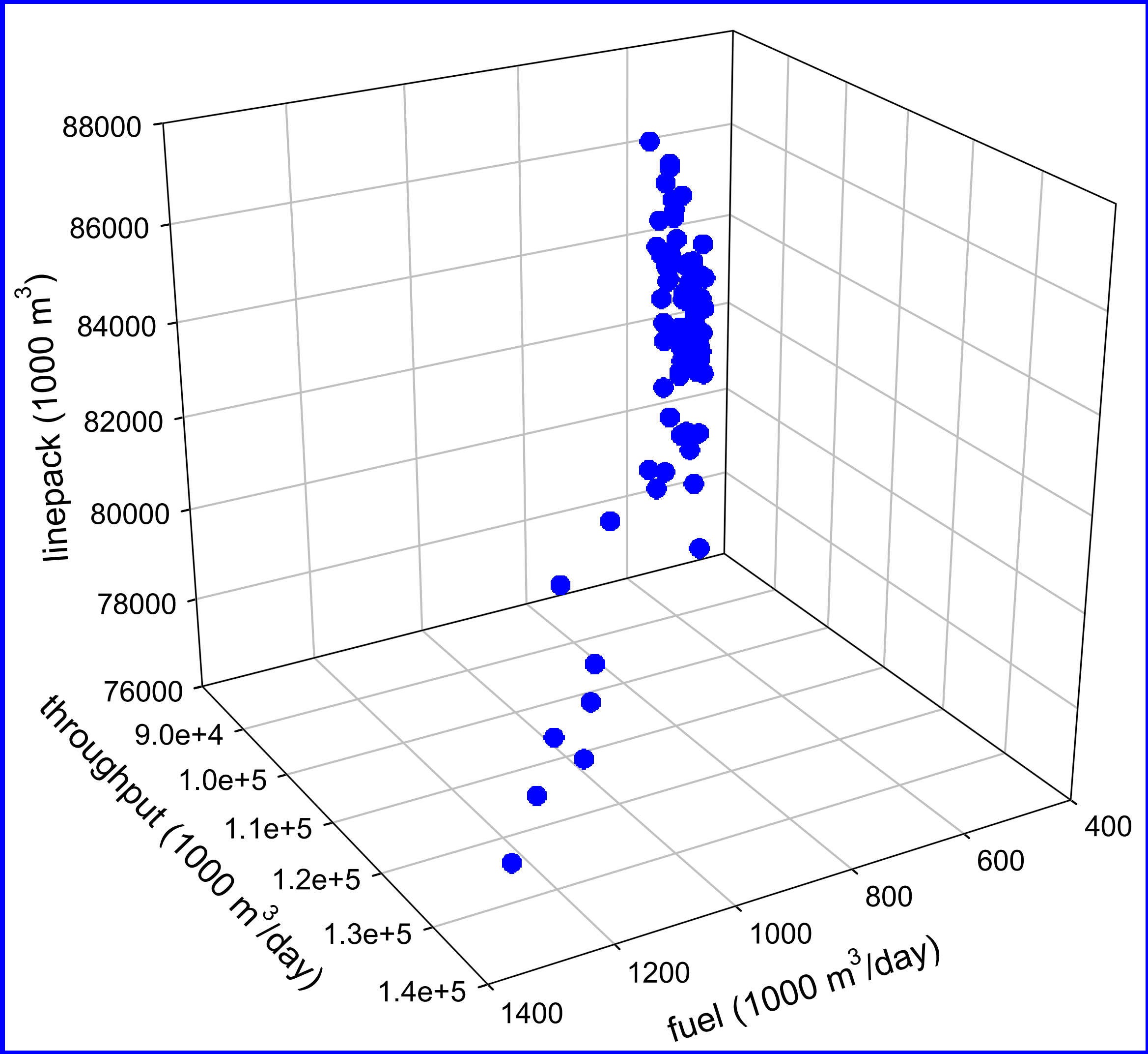
# A Selected Pareto Case



# Max Linepack & Max Throughput



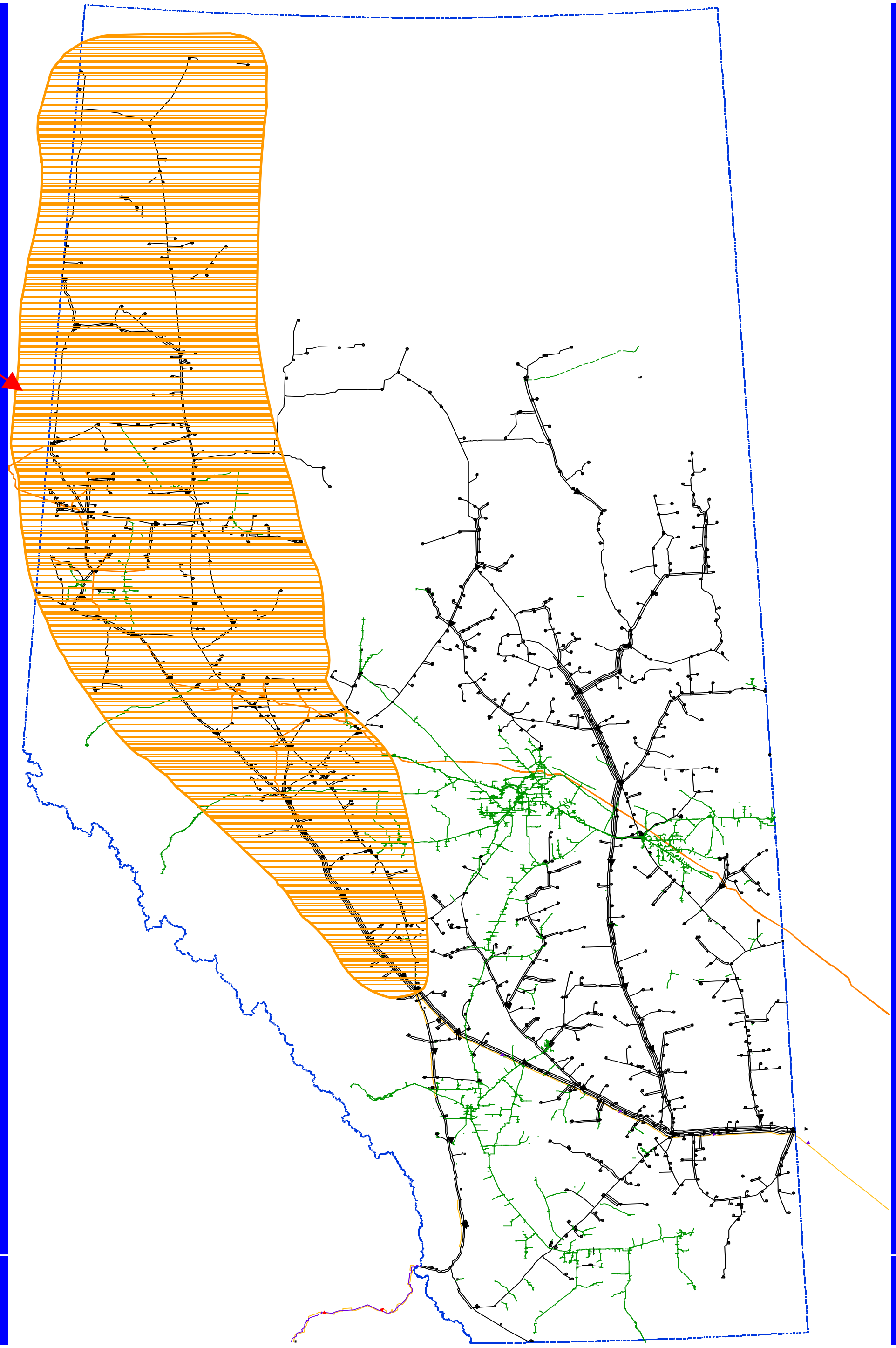
# Three Objectives



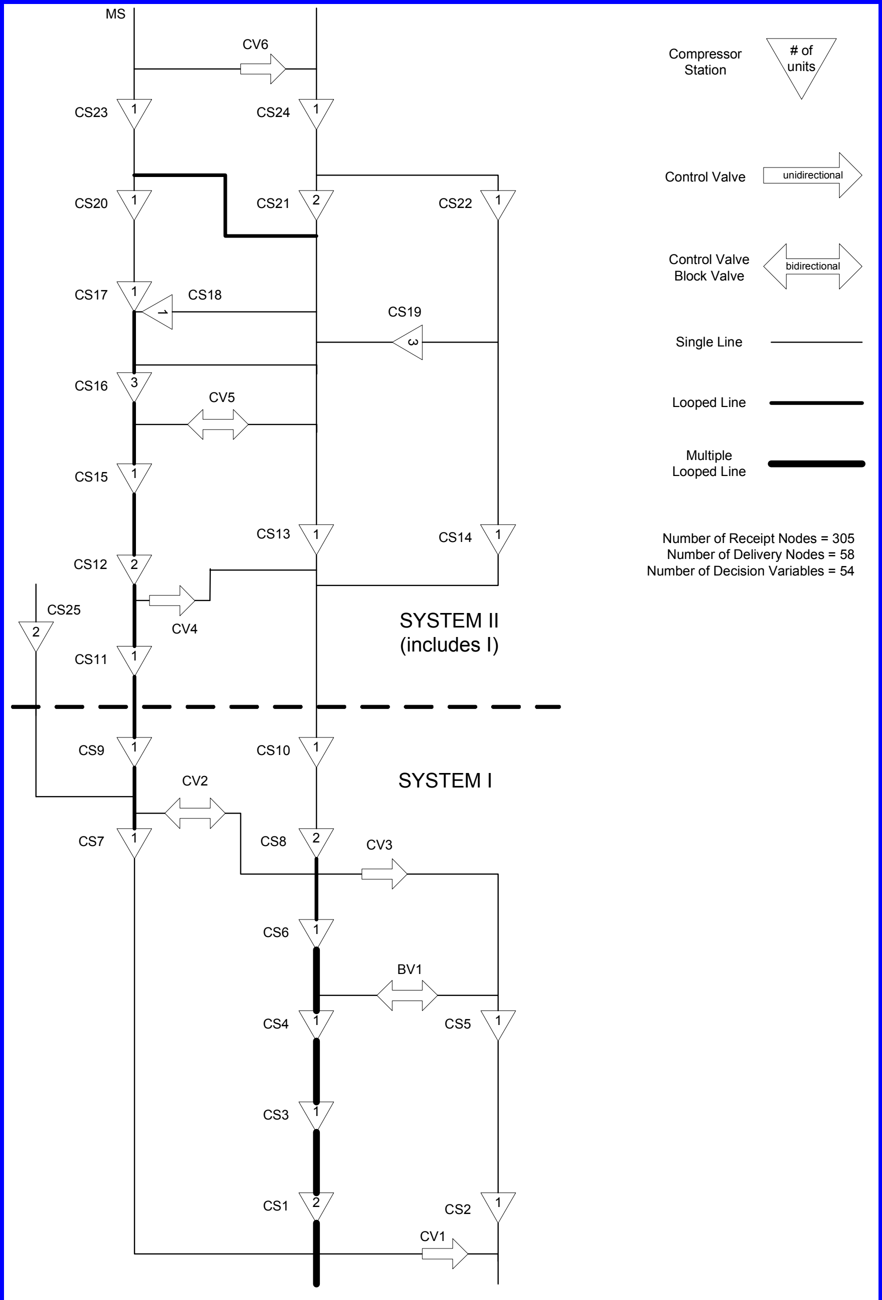
# Example 2:

## *Notion of Dynamic Penalty Parameters*

# Example 2:



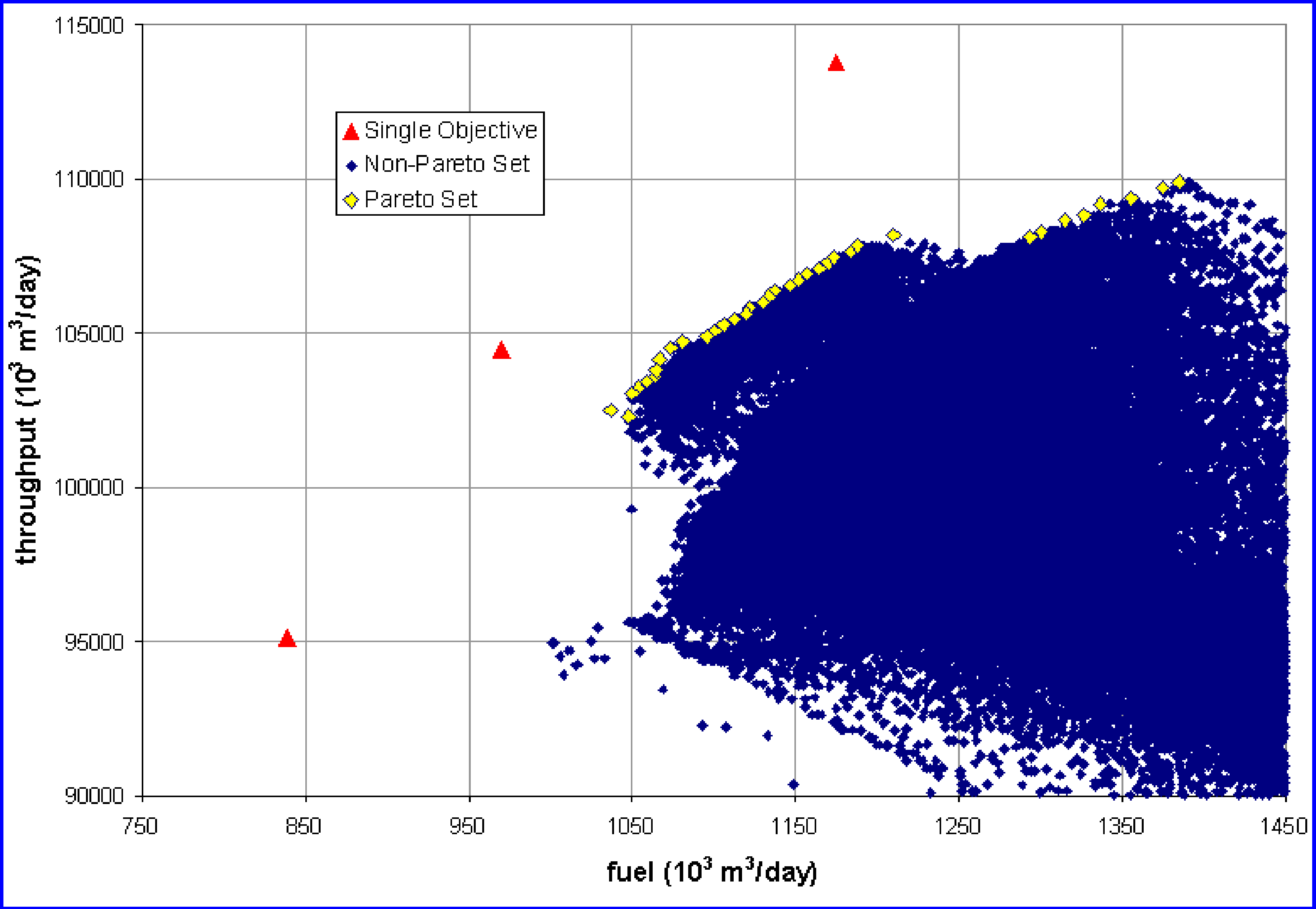
# Example 2:


MS
CV6
CS23
CS24
CS20
CS21
CS22
CS17
CS18
CS19
CS16
CV5
CS15
CS13
CS14
CS12
CS25
CV4
CS11
SYSTEM II
(includes I)
CS9
CS10
CV2
SYSTEM I
CS7
CS8
CV3
CS6
BV1
CS4
CS5
CS3
CS1
CS2
CV1
Compressor Station
# of units
Control Valve
unidirectional
Control Valve Block Valve
bidirectional
Single Line
Looped Line
Multiple Looped Line
Number of Receipt Nodes = 305
Number of Delivery Nodes = 58
Number of Decision Variables = 54

# Decision Variables

- Decision Variables:
  - 1 flow variable
  - 6 control valve node pressures
  - **22 compressor** station control pressures (suction or discharge);
  - 22 compressor station statuses (on/off)
  - 3 block valve status (open/closed)

- Total of **54** decision variables
- Searching through $10^{78}$ possible design cases



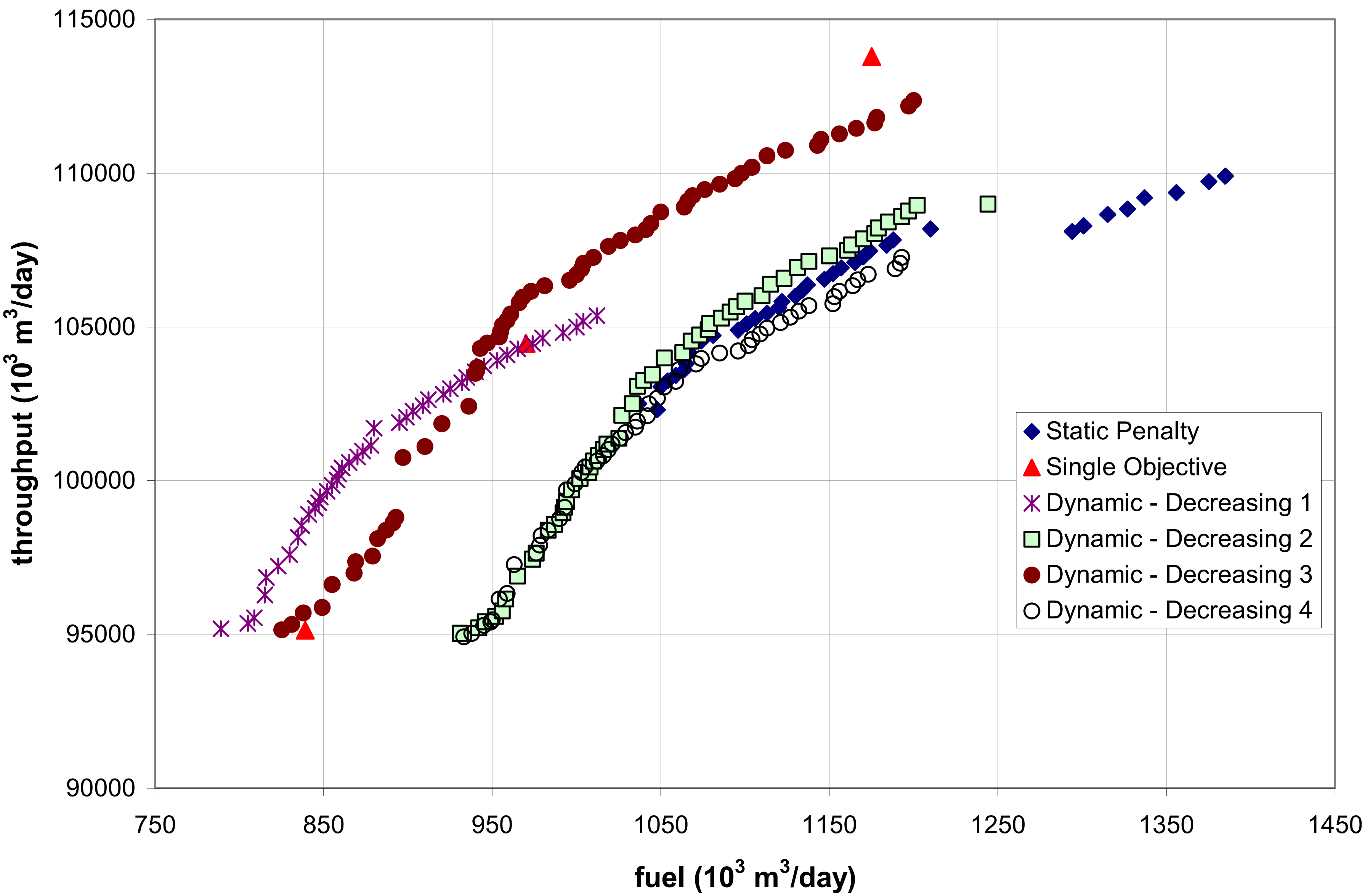
115000
110000
105000
100000
95000
90000
throughput ($10^3$ m$^3$/day)
750
850
950
1050
1150
1250
1350
1450
fuel ($10^3$ m$^3$/day)
Single Objective
Non-Pareto Set
Pareto Set

- Change the penalty scheme to incorporate a dynamic factor (DF) that can be changed with generations:

$$fitness = objective + DF \cdot \sum R_i \cdot Penalty_i$$

$$where:$$

$$DF = f\left(Generation\ \#\right)$$

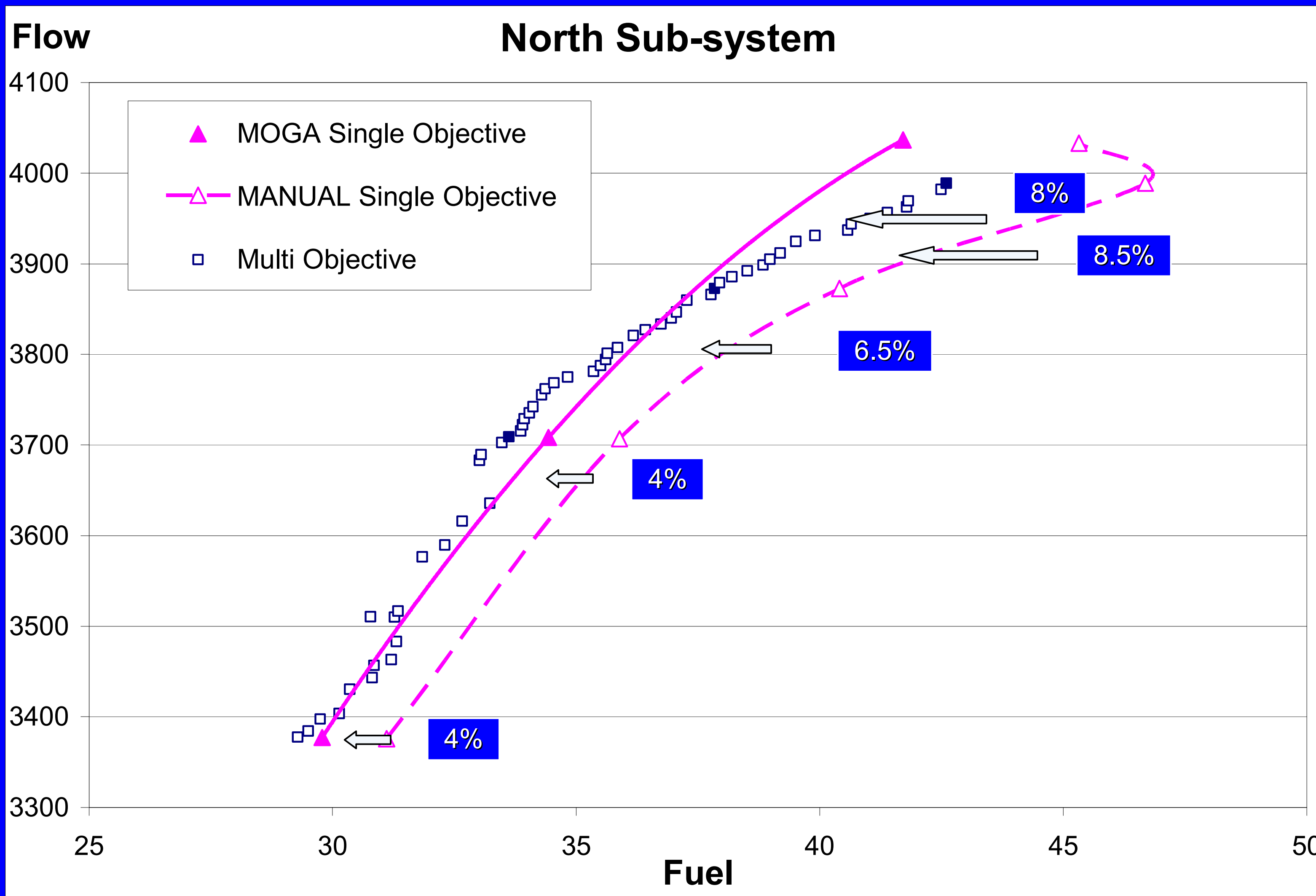

throughput ($10^3$ $m^3$/day)
115000
110000
105000
100000
95000
90000
750
850
950
1050
1150
1250
1350
1450
fuel ($10^3$ $m^3$/day)
Static Penalty
Single Objective
Dynamic - Decreasing 1
Dynamic - Decreasing 2
Dynamic - Decreasing 3
Dynamic - Decreasing 4

# Computational Effort

- Computation time becomes a limiting factor as level of complexity of the system increases.
- Current processor: 2.8 GHz, 1 GB of 1066 MHz RAM, PC

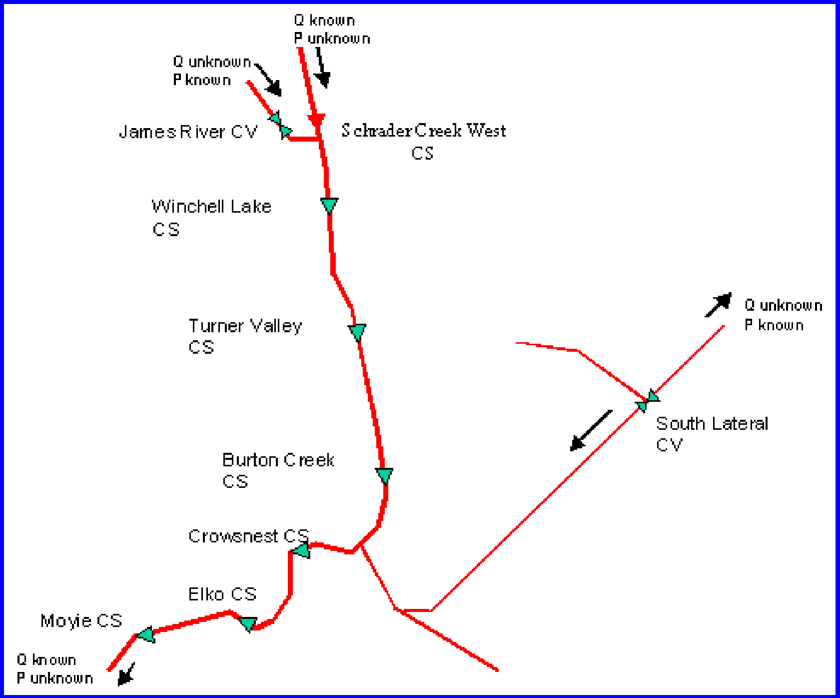
| | Compressor model | Population size | Number of generations | Cases run simultaneously | Total time required |
|---|---|---|---|---|---|
| **System 1** | Block Power | 200 | 50 | 1 | 3:10:00 |
| | Block Power | 400 | 100 | 1 | 9:20:00 |
| | DSM | 400 | 100 | 1 | 9:44:00 |
| | DSM | 400 | 100 | 4 | 7:08:00 |
| **System 2** | Block Power | 500 | 100 | 1 | 23:23:00 |
| | DSM | 500 | 100 | 1 | 29:58:00 |
| | DSM | 500 | 100 | 4 | 17:47:00 |
| | DSM | 2000 | 100 | 4 | 91:05:00 |
| | DSM | 500 | 200 | 4 | 37:07:00 |

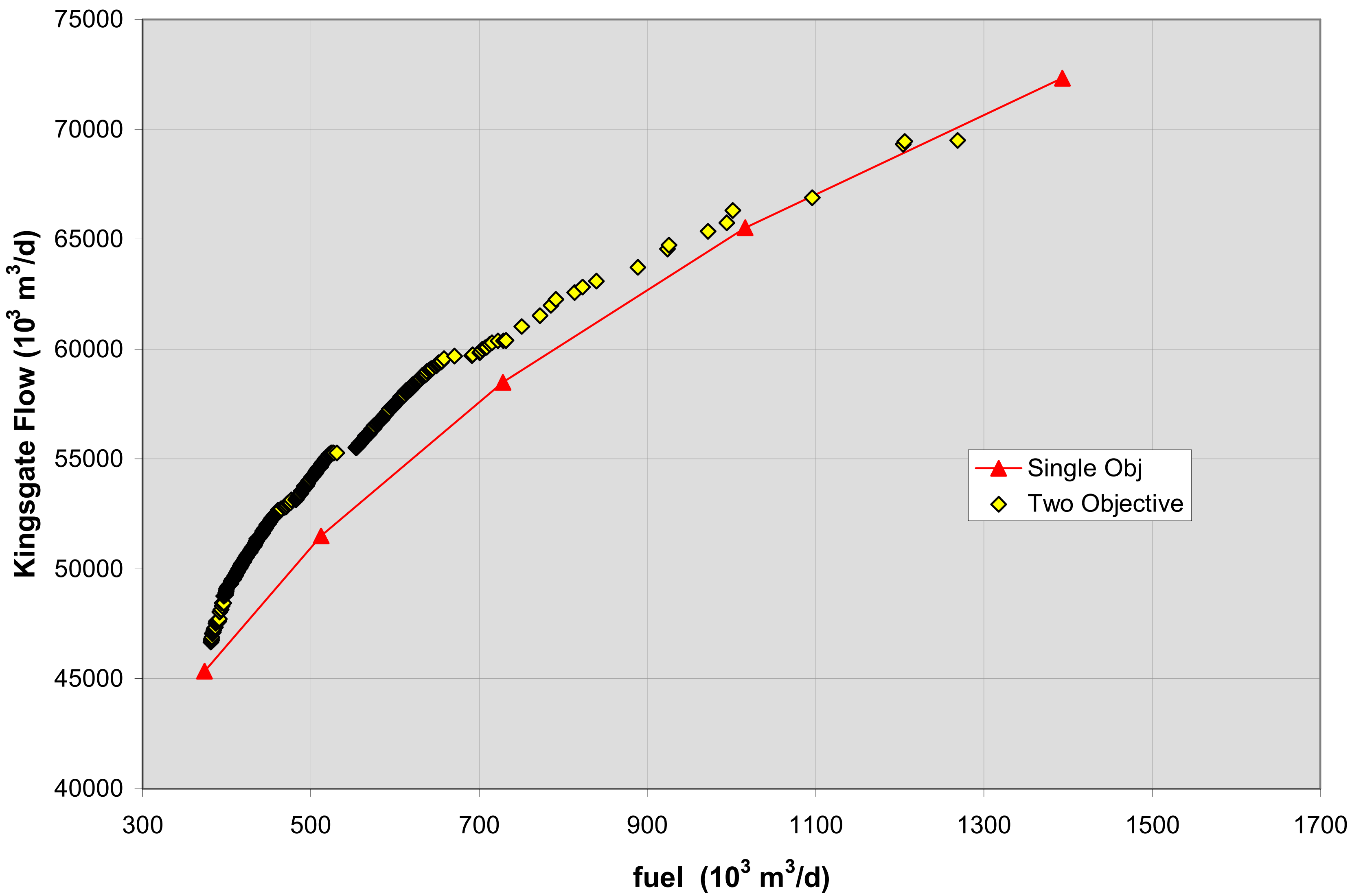
# Results Validation



# Third Example

- West Path Sub-System:
  - searching $10^{24}$ possible cases
  - single and two-objective
  - static penalty produced satisfactory results





75000
70000
65000
60000
55000
50000
45000
40000
Kingsgate Flow (10^3 m^3/d)
300
500
700
900
1100
1300
1500
1700
fuel (10^3 m^3/d)
Single Obj
Two Objective

# Computational Effort

- Computation time becomes a limiting factor as level of complexity of the system increases.
- Current processor: 2.8 GHz, 1 GB of 1066 MHz RAM, PC

| | Single Objective | Two Objective |
|---|---|---|
| **NORTH Sub-system** | GA, 500x200<br>static penalties<br>~2 days | GA, 1000x200<br>dynamic penalties<br>~4 days |
| **WPATH Sub-system** | GA, 500x100<br>static penalties<br>~5 hours | GA, 1000x200<br>static penalties<br>~1 day |

# Reducing Computation Time

- **Option A**: reduce number of design cases evaluated
- **Option B**: simplify model to reduce computational effort per case
- **Option C**: more computer power
- Combination of above

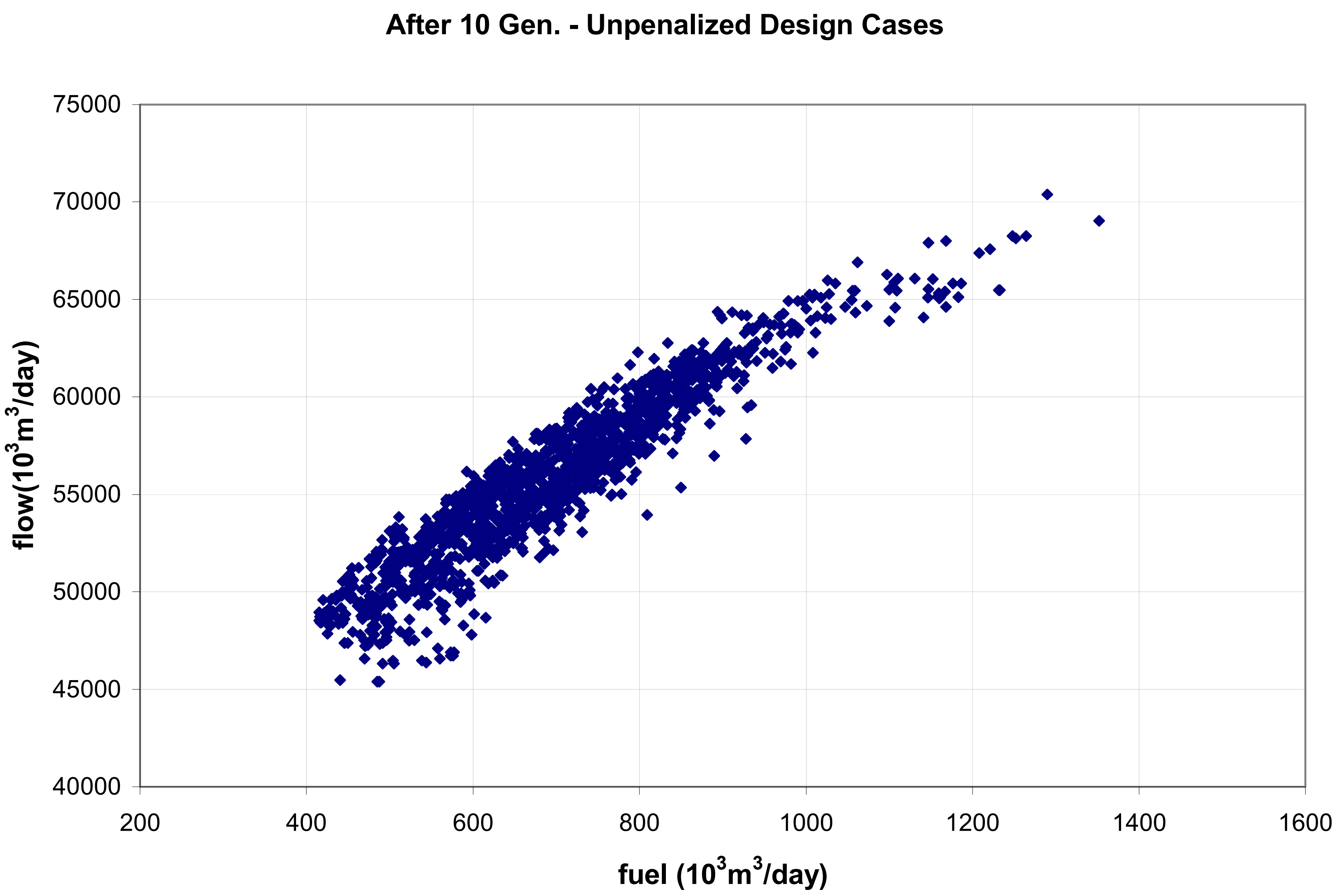
# Surrogate Methods

- <u>**A combination of ‘A’ and ‘B’**</u>:
  - Hydraulic model:
    - High-fidelity model,
    - high computational effort.
  - Surrogate model:
    - Low-fidelity model,
    - lower detail,
    - “training” required,
    - low computational effort.

# Surrogate Methods

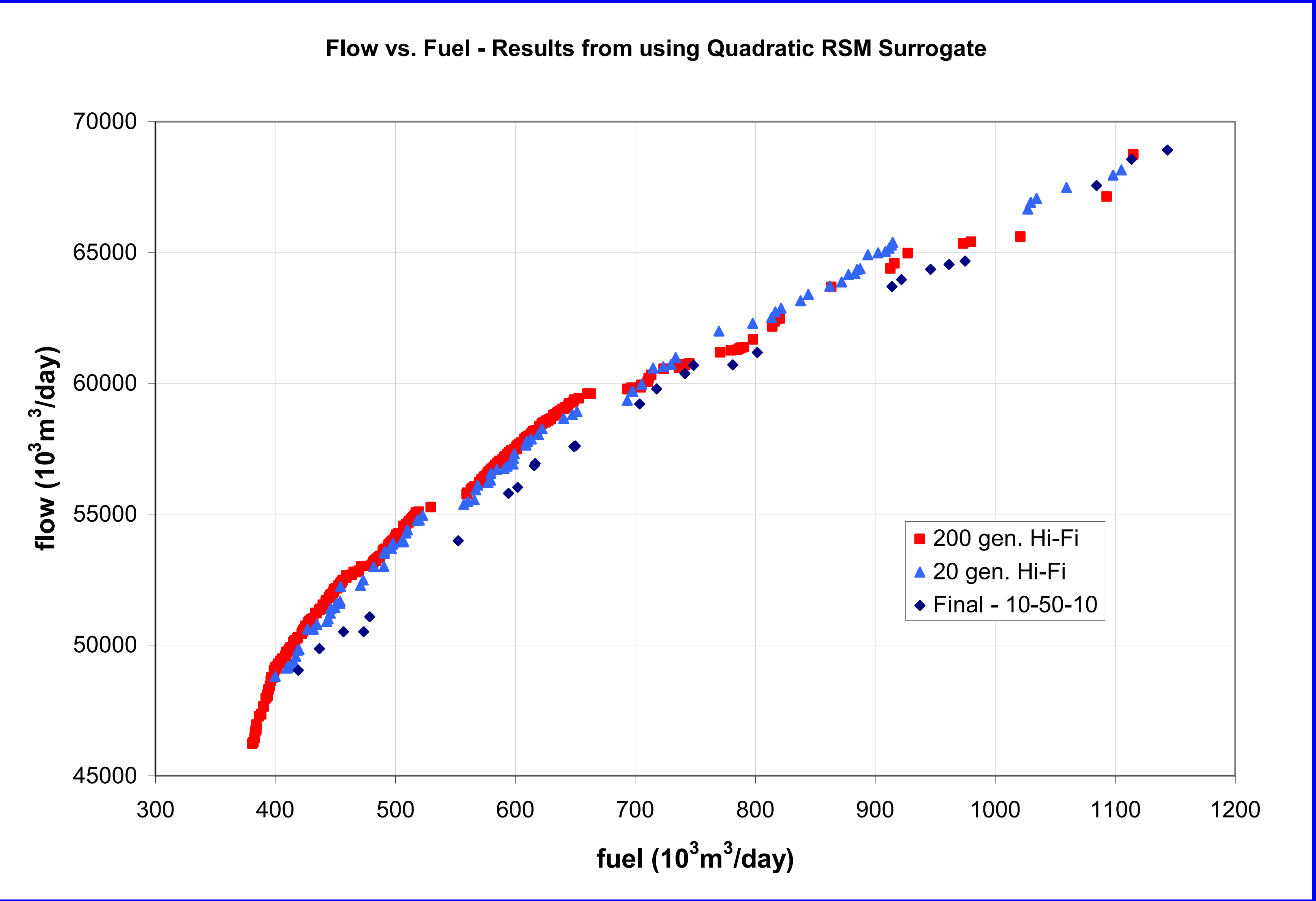
- Test three low-fidelity (surrogate) models:
  - Kriging
  - Neural Networks
  - Quadratic Response Surface

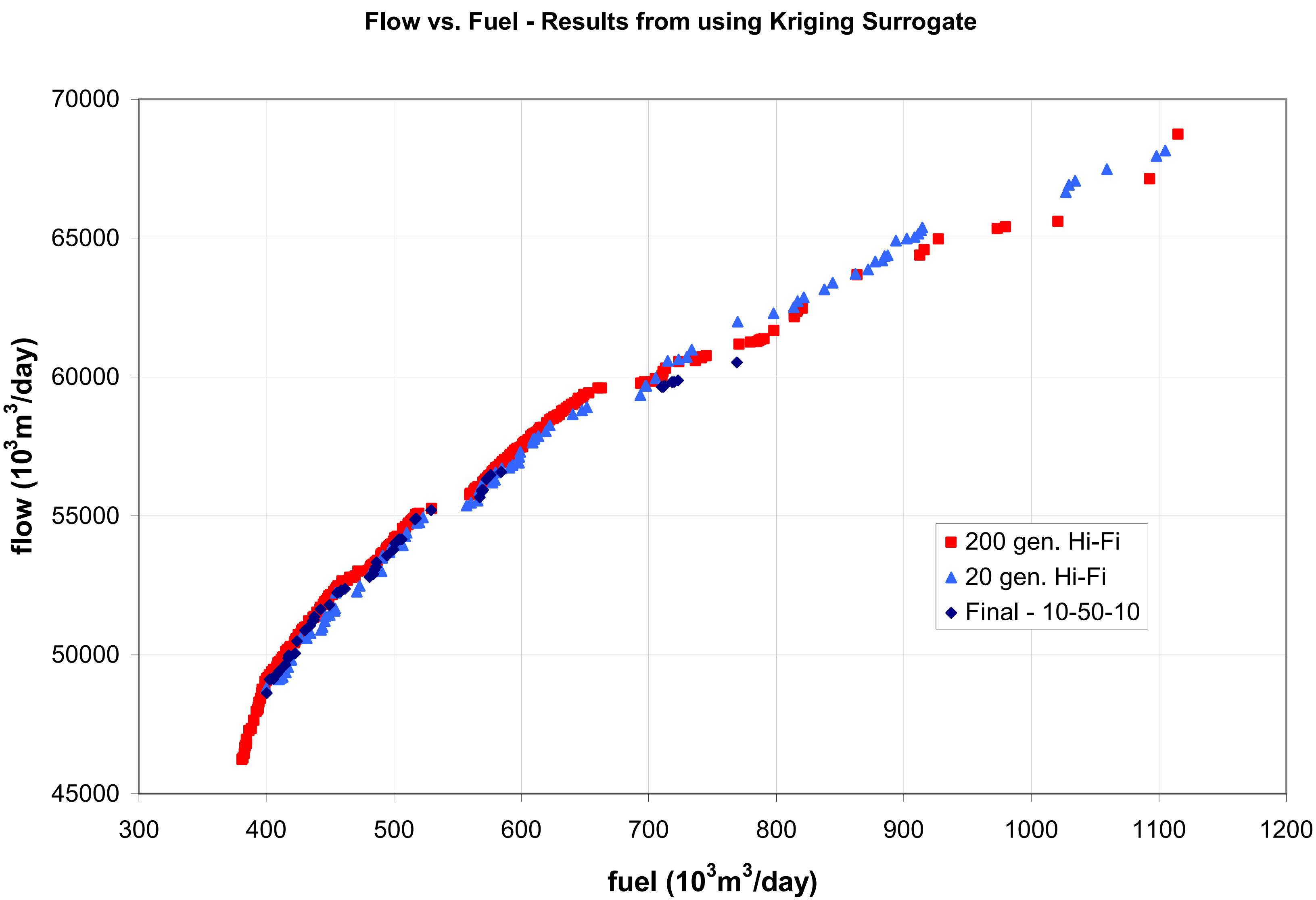
- Training data from a short GA run (e.g. best population after 10 generations)


After 10 Gen. - Unpenalized Design Cases
75000
70000
65000
60000
55000
50000
45000
40000
flow(10^3m^3/day)
200
400
600
800
1000
1200
1400
1600
fuel (10^3m^3/day)

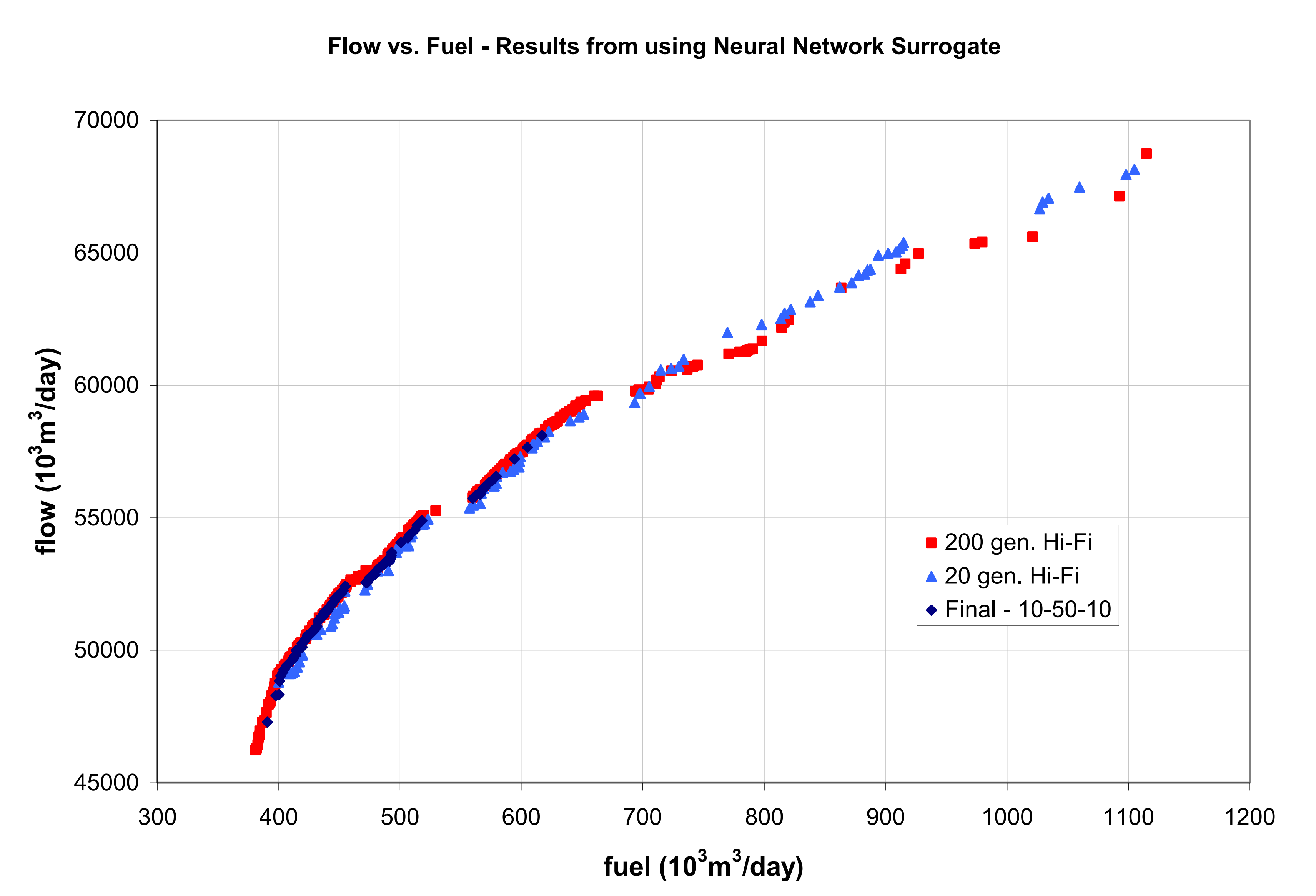
# Surrogate Methods

- Each model was “trained”.
- 50 generations using surrogate.
- Best results fed back into additional 10 generations using hydraulic simulator.


Flow vs. Fuel - Results from using Quadratic RSM Surrogate
70000
65000
60000
55000
50000
45000
flow (10³m³/day)
300
400
500
600
700
800
900
1000
1100
1200
fuel (10³m³/day)
200 gen. Hi-Fi
20 gen. Hi-Fi
Final - 10-50-10



Flow vs. Fuel - Results from using Kriging Surrogate
70000
65000
60000
55000
50000
45000
flow (10$^3$m$^3$/day)
300
400
500
600
700
800
900
1000
1100
1200
fuel (10$^3$m$^3$/day)
200 gen. Hi-Fi
20 gen. Hi-Fi
Final - 10-50-10


Flow vs. Fuel - Results from using Neural Network Surrogate
flow (10³m³/day)
70000
65000
60000
55000
50000
45000
300
400
500
600
700
800
900
1000
1100
1200
fuel (10³m³/day)
200 gen. Hi-Fi
20 gen. Hi-Fi
Final - 10-50-10

# Computational Effort

- Benefit may not be significant:

| | 10 Hi-Fi | 50 Lo-Fi | 10 Hi-Fi | Total (hrs) |
|---|---|---|---|---|
| **Quad. RSM** | 1.57 | 0.02 | 1.57 | **3.15** |
| **Neural Net** | 1.57 | 2.47 | 2.43 | **6.47** |
| **Kriging** | 1.57 | 1.30 | 1.45 | **4.32** |
| **20 Gen Hi-Fi only** | | | | **3.05** |
| **200 Gen Hi-Fi only** | | | | **35.00** |

# Conclusions

- Application of GA is robust.
- Careful consideration to GA "tuning" parameters and proper constraint handling
- Application of surrogate methods can improve computational time.

# Acknowledgment

Jeff Stoffregen
Hossein Golshan
Krystyna Jungowski
Gene Poissant
Daniel Sennhauser
Landen Stein
Thomas Robinson
&
TransCanada: for granting permission to present results