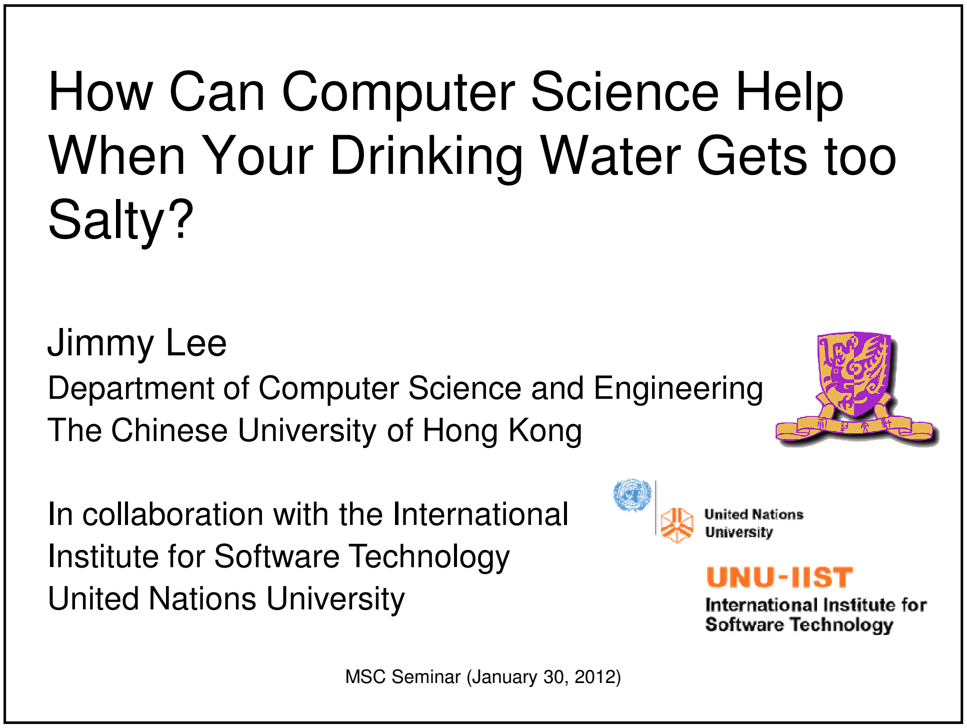


How Can Computer Science Help When Your Drinking Water Gets too Salty?

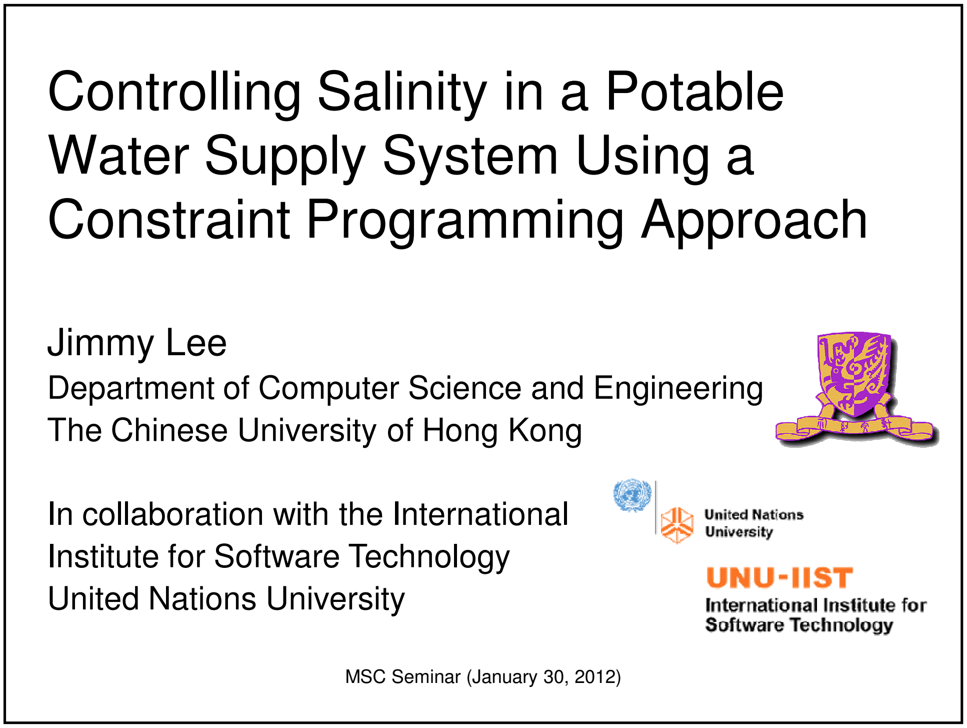
Jimmy Lee
Department of Computer Science and Engineering
The Chinese University of Hong Kong

In collaboration with the International
Institute for Software Technology
United Nations University

United Nations University

UNU-IIST
International Institute for Software Technology

MSC Seminar (January 30, 2012)



Controlling Salinity in a Potable Water Supply System Using a Constraint Programming Approach

Jimmy Lee
Department of Computer Science and Engineering
The Chinese University of Hong Kong

In collaboration with the International
Institute for Software Technology
United Nations University

United Nations University

UNU-IIST
International Institute for Software Technology

MSC Seminar (January 30, 2012)

Outline

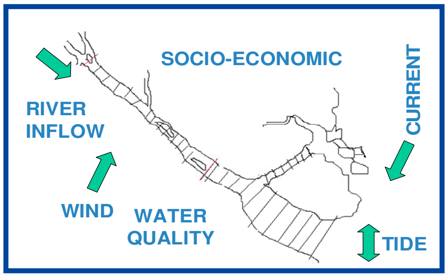
- Domain Description
- Constraint Programming (CP)
- Problem Modelling
- Improvements
- Concluding Remarks

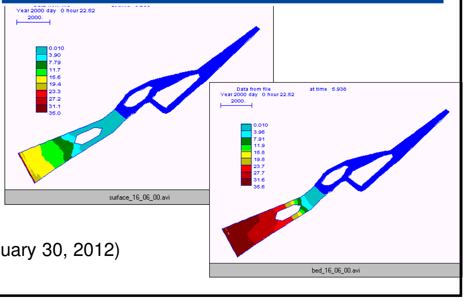
Outline

- Domain Description
- Constraint Programming (CP)
- Problem Modelling
- Improvements
- Concluding Remarks

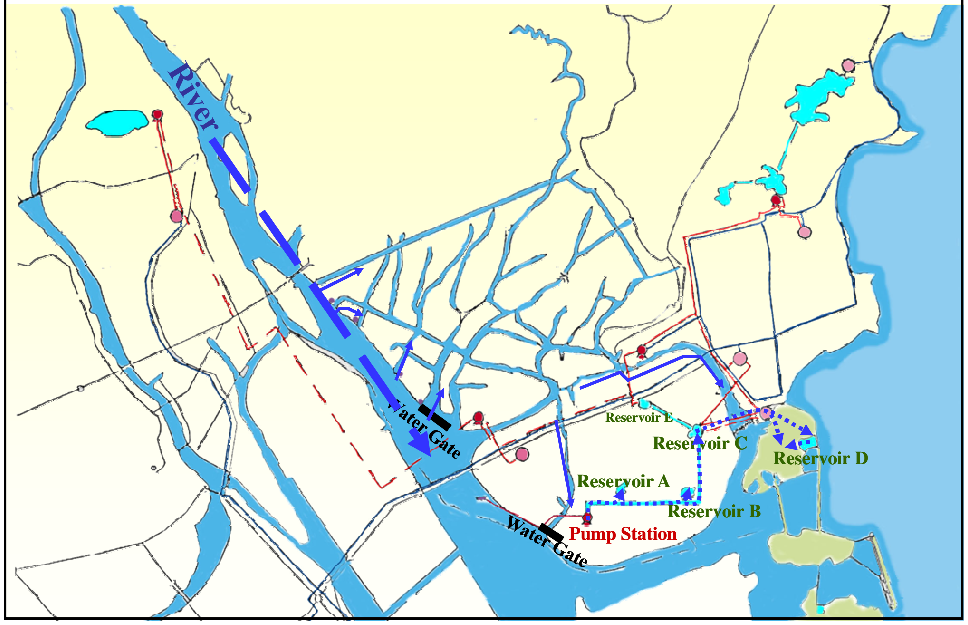
Increasing Salinity in Water

- Salinity is the concentration of salts in water
- Decrease of river flow during dry seasons
- Intrusion of sea water
- Duration depends on unforeseen factors such as tidal flow and weather conditions

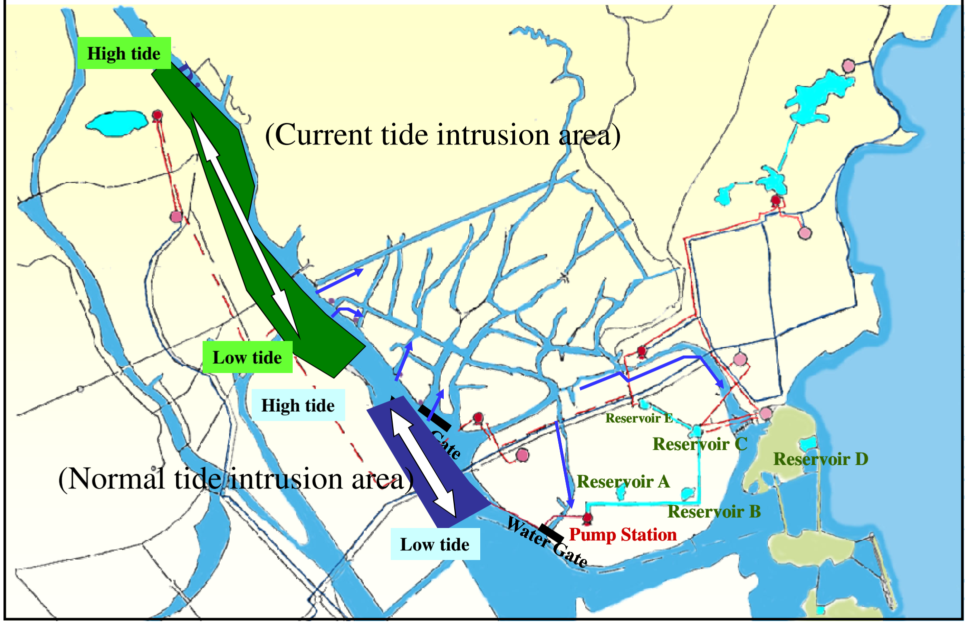




The Raw Water System



The Salinity Problem



How Serious is the Problem?

- Year 2004 was one of the driest years in the past 50 years and the situation has got only worse
- Rainfall reduced by more than 30% of average
- Salinity of raw water can drastically rise to such levels as 2500 ppm (c.f. ≤ 250 ppm ideally)
- Affecting approx. 450,000 residents of the city

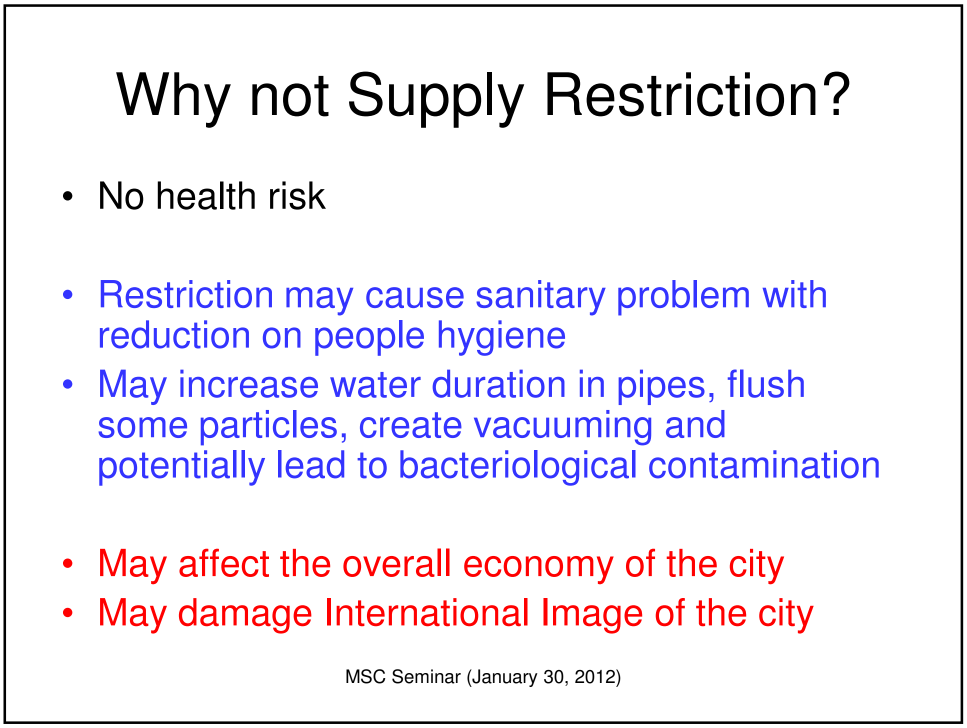
Consequence of Salinity

- The salinity is more a matter of taste (increased content of chloride) than health while it is less than 2000 ppm

Salinity level	Taste
200 - 300 ppm	Very slight saline taste
>300 - 600 ppm	Noticeable discomfort in taste
>600 - 800 ppm	Increasing discomfort in taste
>800 - 1000 ppm	Very strong discomfort in taste
>1000 - 2000 ppm	Extremely strong discomfort in taste

W.H.O. Guidelines

- Average dietary intake of chloride for human
  - Ranging from 6 g/day to 12 g/day
- The consequence is that daily intake of salt from water is usually less than 5% to 10% of total intake from foods
- No evidence on health effect of prolonged intake of large amounts of chloride in diet
  - Except in the special case of impaired sodium chloride metabolism, e.g. in congestive heart failure

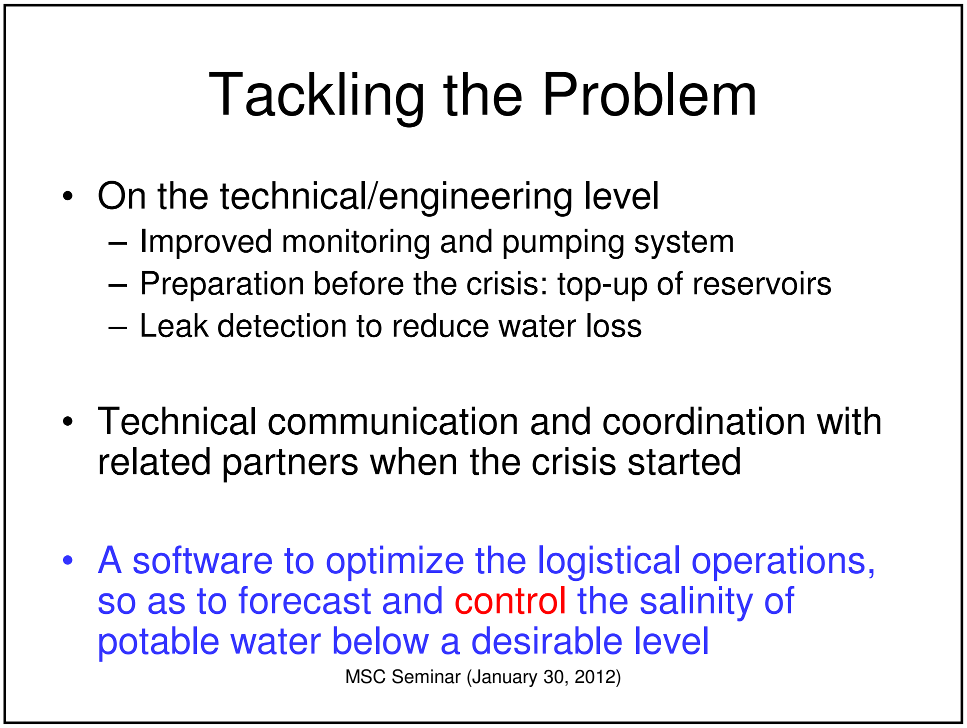


Why not Supply Restriction?

- No health risk

- Restriction may cause sanitary problem with reduction on people hygiene
- May increase water duration in pipes, flush some particles, create vacuuming and potentially lead to bacteriological contamination

- May affect the overall economy of the city
- May damage International Image of the city



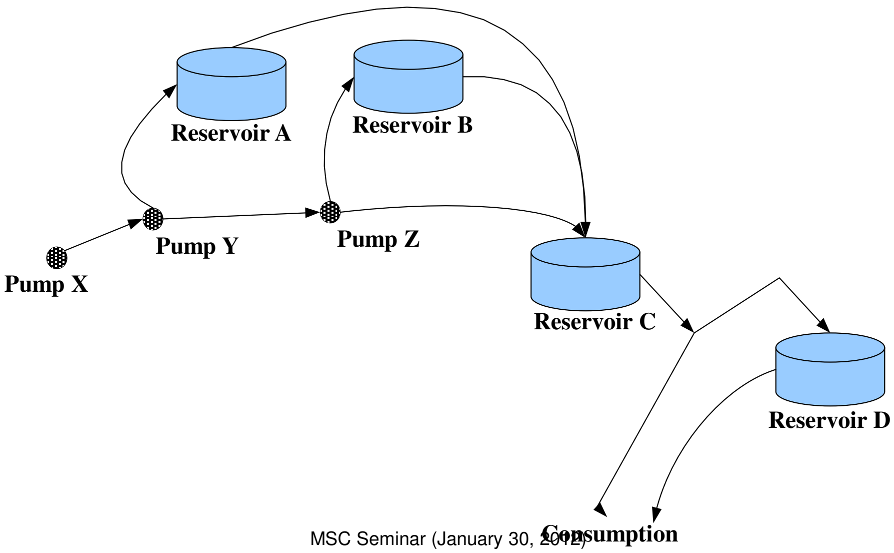
Tackling the Problem

- On the technical/engineering level
  - Improved monitoring and pumping system
  - Preparation before the crisis: top-up of reservoirs
  - Leak detection to reduce water loss

- Technical communication and coordination with related partners when the crisis started

- A software to optimize the logistical operations, so as to forecast and control the salinity of potable water below a desirable level

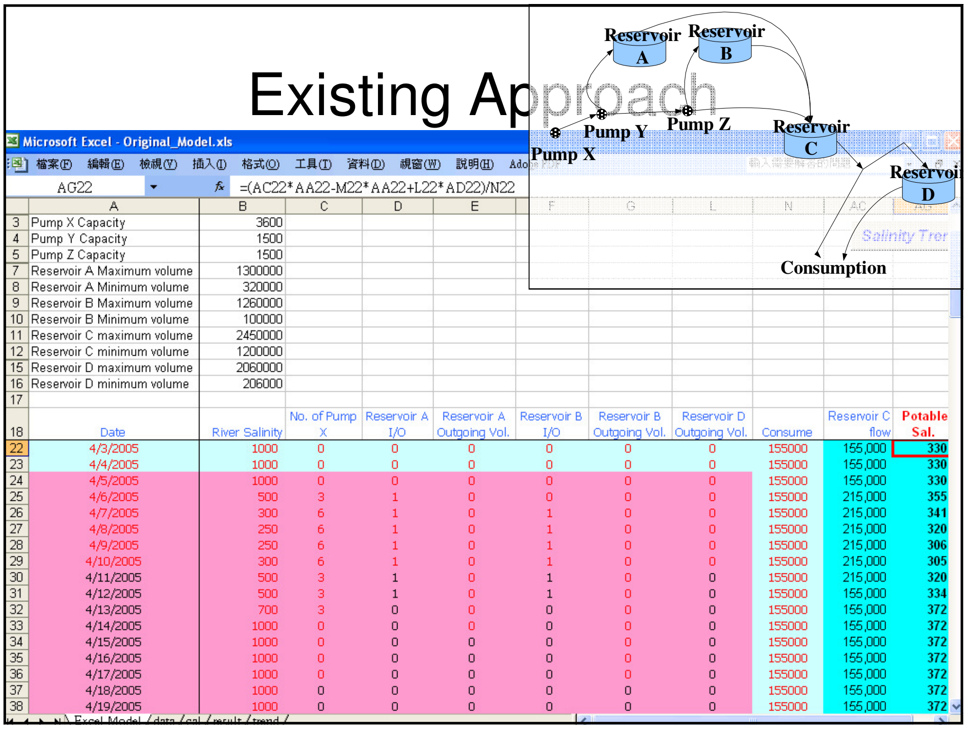
Raw Water System Model



Objective

To maximize the number of days in which the potable salinity level is below the desirable level

Existing Approach

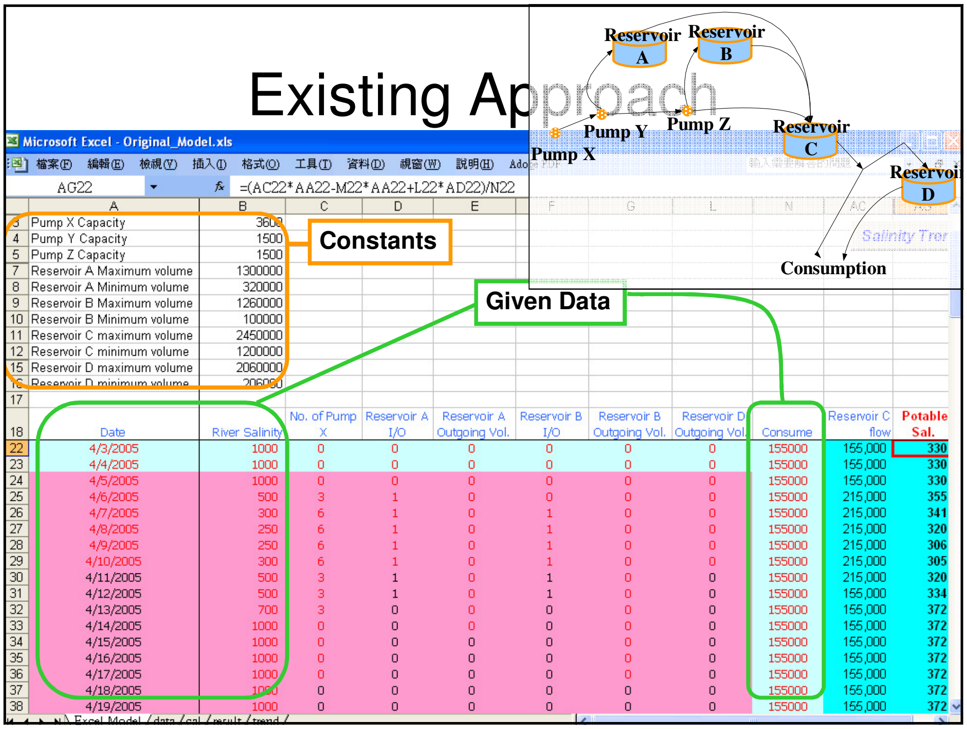

Reservoir A
Reservoir B
Pump X
Pump Y
Pump Z
Reservoir C
Reservoir D
Consumption
Microsoft Excel - Original_Model.xls
檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H)
Salinity Trer
Excel Model / data / cal / result / trend /

AG22 =(AC22*AA22-M22*AA22+L22*AD22)/N22

	A	B
3	Pump X Capacity	3600
4	Pump Y Capacity	1500
5	Pump Z Capacity	1500
7	Reservoir A Maximum volume	1300000
8	Reservoir A Minimum volume	320000
9	Reservoir B Maximum volume	1260000
10	Reservoir B Minimum volume	100000
11	Reservoir C maximum volume	2450000
12	Reservoir C minimum volume	1200000
15	Reservoir D maximum volume	2060000
16	Reservoir D minimum volume	206000
17		

18	Date	River Salinity	No. of Pump X	Reservoir A I/O	Reservoir A Outgoing Vol.	Reservoir B I/O	Reservoir B Outgoing Vol.	Reservoir D Outgoing Vol.	Consume	Reservoir C flow	Potable Sal.
22	4/3/2005	1000	0	0	0	0	0	0	155000	155,000	330
23	4/4/2005	1000	0	0	0	0	0	0	155000	155,000	330
24	4/5/2005	1000	0	0	0	0	0	0	155000	155,000	330
25	4/6/2005	500	3	1	0	0	0	0	155000	215,000	355
26	4/7/2005	300	6	1	0	1	0	0	155000	215,000	341
27	4/8/2005	250	6	1	0	1	0	0	155000	215,000	320
28	4/9/2005	250	6	1	0	1	0	0	155000	215,000	306
29	4/10/2005	300	6	1	0	1	0	0	155000	215,000	305
30	4/11/2005	500	3	1	0	1	0	0	155000	215,000	320
31	4/12/2005	500	3	1	0	1	0	0	155000	155,000	334
32	4/13/2005	700	3	0	0	0	0	0	155000	155,000	372
33	4/14/2005	1000	0	0	0	0	0	0	155000	155,000	372
34	4/15/2005	1000	0	0	0	0	0	0	155000	155,000	372
35	4/16/2005	1000	0	0	0	0	0	0	155000	155,000	372
36	4/17/2005	1000	0	0	0	0	0	0	155000	155,000	372
37	4/18/2005	1000	0	0	0	0	0	0	155000	155,000	372
38	4/19/2005	1000	0	0	0	0	0	0	155000	155,000	372

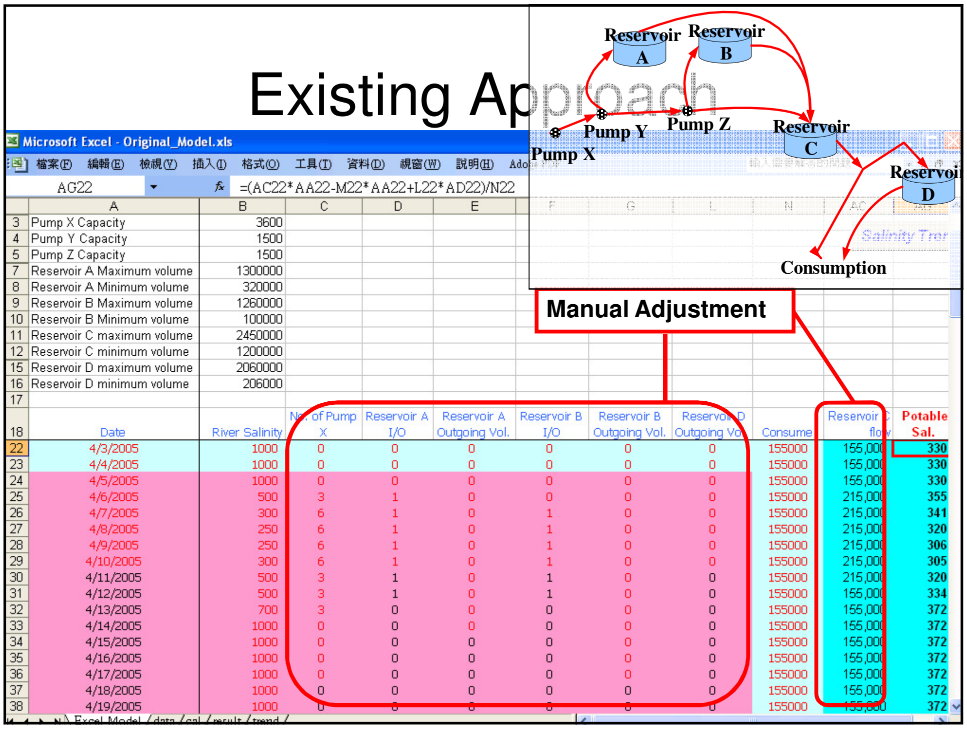
Existing Approach


Reservoir A
Reservoir B
Pump X
Pump Y
Pump Z
Reservoir C
Reservoir D
Consumption
Microsoft Excel - Original_Model.xls
檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H)
Salinity Trer
Constants
Given Data
Excel Model / data / cal / result / trend /

AG22 =(AC22*AA22-M22*AA22+L22*AD22)/N22

	A	B
3	Pump X Capacity	3600
4	Pump Y Capacity	1500
5	Pump Z Capacity	1500
7	Reservoir A Maximum volume	1300000
8	Reservoir A Minimum volume	320000
9	Reservoir B Maximum volume	1260000
10	Reservoir B Minimum volume	100000
11	Reservoir C maximum volume	2450000
12	Reservoir C minimum volume	1200000
15	Reservoir D maximum volume	2060000
16	Reservoir D minimum volume	206000
17		

18	Date	River Salinity	No. of Pump X	Reservoir A I/O	Reservoir A Outgoing Vol.	Reservoir B I/O	Reservoir B Outgoing Vol.	Reservoir D Outgoing Vol.	Consume	Reservoir C flow	Potable Sal.
22	4/3/2005	1000	0	0	0	0	0	0	155000	155,000	330
23	4/4/2005	1000	0	0	0	0	0	0	155000	155,000	330
24	4/5/2005	1000	0	0	0	0	0	0	155000	155,000	330
25	4/6/2005	500	3	1	0	0	0	0	155000	215,000	355
26	4/7/2005	300	6	1	0	1	0	0	155000	215,000	341
27	4/8/2005	250	6	1	0	1	0	0	155000	215,000	320
28	4/9/2005	250	6	1	0	1	0	0	155000	215,000	306
29	4/10/2005	300	6	1	0	1	0	0	155000	215,000	305
30	4/11/2005	500	3	1	0	1	0	0	155000	215,000	320
31	4/12/2005	500	3	1	0	1	0	0	155000	155,000	334
32	4/13/2005	700	3	0	0	0	0	0	155000	155,000	372
33	4/14/2005	1000	0	0	0	0	0	0	155000	155,000	372
34	4/15/2005	1000	0	0	0	0	0	0	155000	155,000	372
35	4/16/2005	1000	0	0	0	0	0	0	155000	155,000	372
36	4/17/2005	1000	0	0	0	0	0	0	155000	155,000	372
37	4/18/2005	1000	0	0	0	0	0	0	155000	155,000	372
38	4/19/2005	1000	0	0	0	0	0	0	155000	155,000	372


Reservoir A
Reservoir B
Pump X
Pump Y
Pump Z
Reservoir C
Reservoir D
Consumption
Microsoft Excel - Original_Model.xls
檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H) Ado
輸入需要解答的問題
Salinity Trer
Excel Model / data / cal / result / trend /

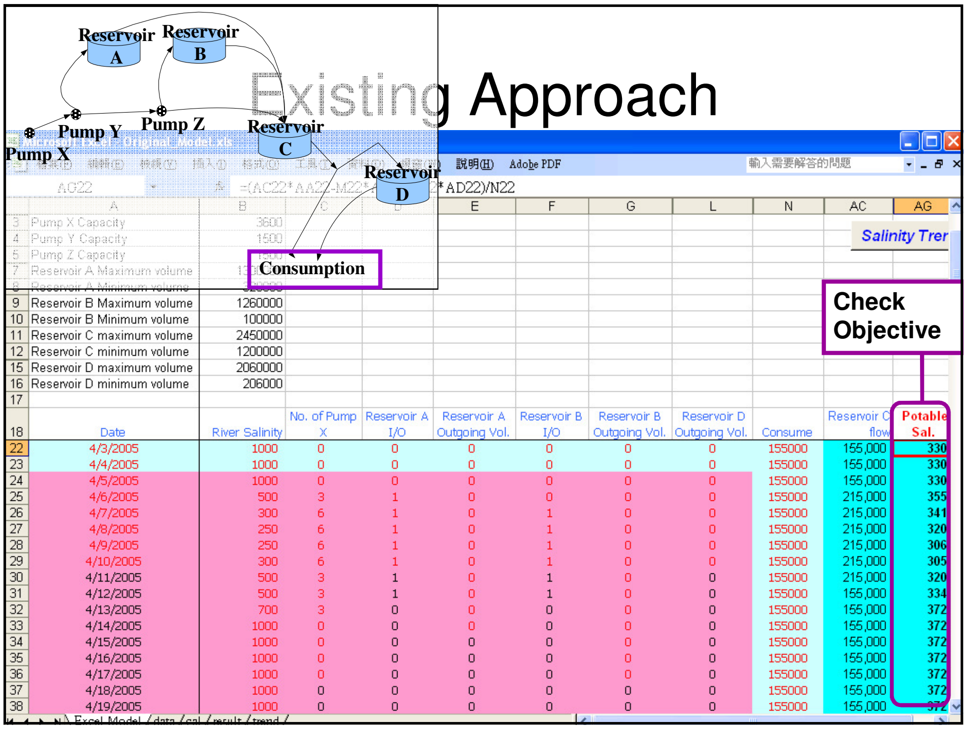
Existing Approach

AG22 =(AC22*AA22-M22*AA22+L22*AD22)/N22

	A	B
3	Pump X Capacity	3600
4	Pump Y Capacity	1500
5	Pump Z Capacity	1500
7	Reservoir A Maximum volume	1300000
8	Reservoir A Minimum volume	320000
9	Reservoir B Maximum volume	1260000
10	Reservoir B Minimum volume	100000
11	Reservoir C maximum volume	2450000
12	Reservoir C minimum volume	1200000
15	Reservoir D maximum volume	2060000
16	Reservoir D minimum volume	206000
17		

18	Date	River Salinity	No. of Pump X	Reservoir A I/O	Reservoir A Outgoing Vol.	Reservoir B I/O	Reservoir B Outgoing Vol.	Reservoir D Outgoing Vo	Consume	Reservoir C flow	Potable Sal.
22	4/3/2005	1000	0	0	0	0	0	0	155000	155,000	330
23	4/4/2005	1000	0	0	0	0	0	0	155000	155,000	330
24	4/5/2005	1000	0	0	0	0	0	0	155000	155,000	330
25	4/6/2005	500	3	1	0	0	0	0	155000	215,000	355
26	4/7/2005	300	6	1	0	1	0	0	155000	215,000	341
27	4/8/2005	250	6	1	0	1	0	0	155000	215,000	320
28	4/9/2005	250	6	1	0	1	0	0	155000	215,000	306
29	4/10/2005	300	6	1	0	1	0	0	155000	215,000	305
30	4/11/2005	500	3	1	0	1	0	0	155000	215,000	320
31	4/12/2005	500	3	1	0	1	0	0	155000	155,000	334
32	4/13/2005	700	3	0	0	0	0	0	155000	155,000	372
33	4/14/2005	1000	0	0	0	0	0	0	155000	155,000	372
34	4/15/2005	1000	0	0	0	0	0	0	155000	155,000	372
35	4/16/2005	1000	0	0	0	0	0	0	155000	155,000	372
36	4/17/2005	1000	0	0	0	0	0	0	155000	155,000	372
37	4/18/2005	1000	0	0	0	0	0	0	155000	155,000	372
38	4/19/2005	1000	0	0	0	0	0	0	155000	155,000	372

Manual Adjustment



Existing Approach

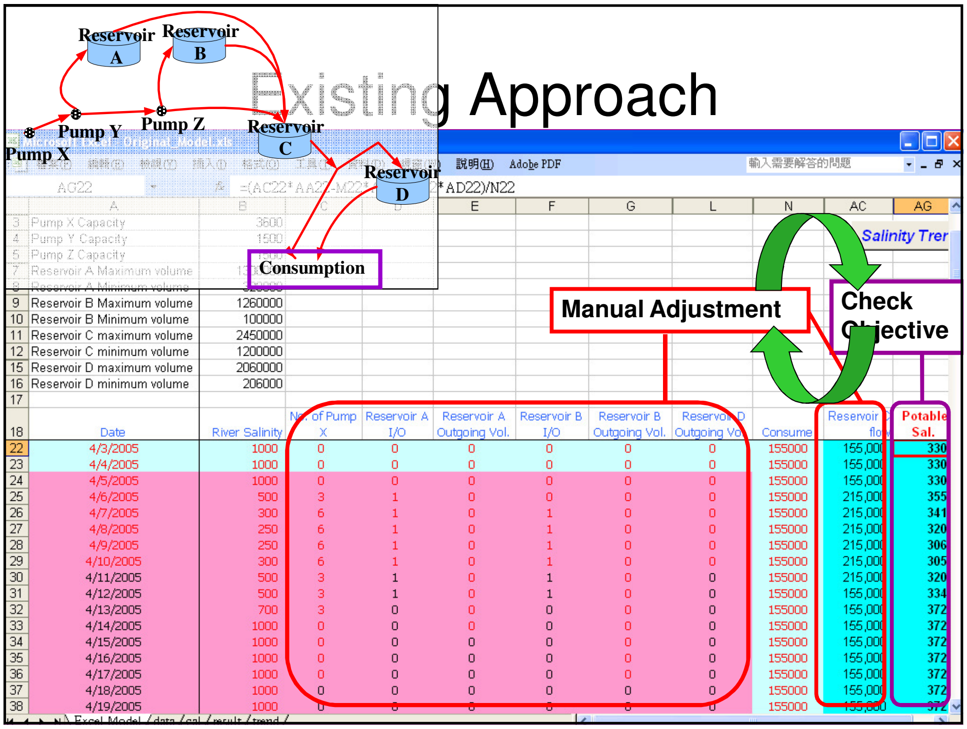
AG22 =(AC22*AA22-M22*AA22+L22*AD22)/N22

	A	B
3	Pump X Capacity	3600
4	Pump Y Capacity	1500
5	Pump Z Capacity	1500
7	Reservoir A Maximum volume	1300000
8	Reservoir A Minimum volume	320000
9	Reservoir B Maximum volume	1260000
10	Reservoir B Minimum volume	100000
11	Reservoir C maximum volume	2450000
12	Reservoir C minimum volume	1200000
15	Reservoir D maximum volume	2060000
16	Reservoir D minimum volume	206000
17		

	A	B	C	D	E	F	G	L	N	AC	AG
18	Date	River Salinity	No. of Pump X	Reservoir A I/O	Reservoir A Outgoing Vol.	Reservoir B I/O	Reservoir B Outgoing Vol.	Reservoir D Outgoing Vol.	Consume	Reservoir C flow	Potable Sal.
22	4/3/2005	1000	0	0	0	0	0	0	155000	155,000	330
23	4/4/2005	1000	0	0	0	0	0	0	155000	155,000	330
24	4/5/2005	1000	0	0	0	0	0	0	155000	155,000	330
25	4/6/2005	500	3	1	0	0	0	0	155000	215,000	355
26	4/7/2005	300	6	1	0	1	0	0	155000	215,000	341
27	4/8/2005	250	6	1	0	1	0	0	155000	215,000	320
28	4/9/2005	250	6	1	0	1	0	0	155000	215,000	306
29	4/10/2005	300	6	1	0	1	0	0	155000	215,000	305
30	4/11/2005	500	3	1	0	1	0	0	155000	215,000	320
31	4/12/2005	500	3	1	0	1	0	0	155000	155,000	334
32	4/13/2005	700	3	0	0	0	0	0	155000	155,000	372
33	4/14/2005	1000	0	0	0	0	0	0	155000	155,000	372
34	4/15/2005	1000	0	0	0	0	0	0	155000	155,000	372
35	4/16/2005	1000	0	0	0	0	0	0	155000	155,000	372
36	4/17/2005	1000	0	0	0	0	0	0	155000	155,000	372
37	4/18/2005	1000	0	0	0	0	0	0	155000	155,000	372
38	4/19/2005	1000	0	0	0	0	0	0	155000	155,000	372

Check Objective

Existing Approach



	A	B
3	Pump X Capacity	3600
4	Pump Y Capacity	1500
5	Pump Z Capacity	1500
7	Reservoir A Maximum volume	[illegible]
8	Reservoir A Minimum volume	[illegible]
9	Reservoir B Maximum volume	1260000
10	Reservoir B Minimum volume	100000
11	Reservoir C maximum volume	2450000
12	Reservoir C minimum volume	1200000
15	Reservoir D maximum volume	2060000
16	Reservoir D minimum volume	206000

	A	B	C	D	E	F	G	L	N	AC	AG
18	Date	River Salinity	No. of Pump X	Reservoir A I/O	Reservoir A Outgoing Vol.	Reservoir B I/O	Reservoir B Outgoing Vol.	Reservoir D Outgoing Vo	Consume	Reservoir C flo	Potable Sal.
22	4/3/2005	1000	0	0	0	0	0	0	155000	155,000	330
23	4/4/2005	1000	0	0	0	0	0	0	155000	155,000	330
24	4/5/2005	1000	0	0	0	0	0	0	155000	155,000	330
25	4/6/2005	500	3	1	0	0	0	0	155000	215,000	355
26	4/7/2005	300	6	1	0	1	0	0	155000	215,000	341
27	4/8/2005	250	6	1	0	1	0	0	155000	215,000	320
28	4/9/2005	250	6	1	0	1	0	0	155000	215,000	306
29	4/10/2005	300	6	1	0	1	0	0	155000	215,000	305
30	4/11/2005	500	3	1	0	1	0	0	155000	215,000	320
31	4/12/2005	500	3	1	0	1	0	0	155000	155,000	334
32	4/13/2005	700	3	0	0	0	0	0	155000	155,000	372
33	4/14/2005	1000	0	0	0	0	0	0	155000	155,000	372
34	4/15/2005	1000	0	0	0	0	0	0	155000	155,000	372
35	4/16/2005	1000	0	0	0	0	0	0	155000	155,000	372
36	4/17/2005	1000	0	0	0	0	0	0	155000	155,000	372
37	4/18/2005	1000	0	0	0	0	0	0	155000	155,000	372
38	4/19/2005	1000	0	0	0	0	0	0	155000	155,000	372

Weaknesses

- Manual trial-and-error is tedious and time consuming
- Problem is too large and too complex for such manual optimization process, and the model is overly simplified
- Require an experienced operator
  - Knowledge to use a spreadsheet
  - Domain knowledge of the logistical operations

Project Goals

- Automate the optimization process and reduce errors
- More realistic model
- Even novice users can operate
- Generate better quality solutions in a shorter time

Outline

- Domain Description
- Constraint Programming (CP)
- Problem Modelling
- Improvements
- Concluding Remarks

Real-life Applications of CP

- Options trading
- Gate allocation to aircrafts in airports
- Selection and scheduling of observations performed by satellites
- DNA sequencing, construction of 3D models of proteins
- Locating faults in the circuits, computing circuit layouts, testing and verification of design
- Many other scheduling, planning and optimization problems

Constraint Satisfaction Problems

A CSP is a triple < **Z**, **D**, **C** >

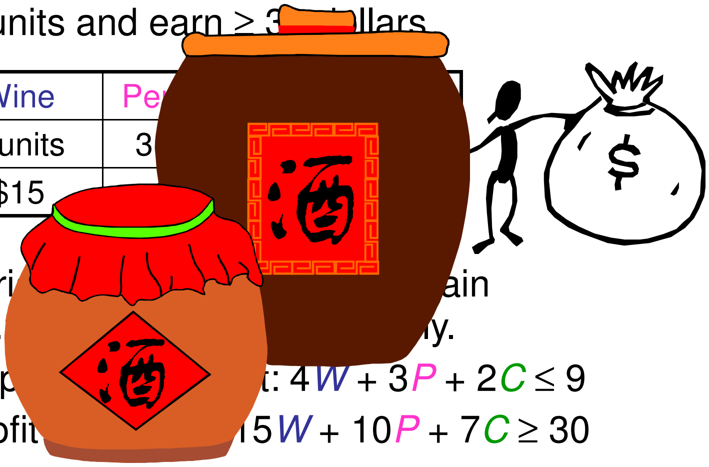
- **Z** is a finite set of variables $\{x_1, x_2, \ldots, x_n\}$
- **D** is a finite set $\{D_1, D_2, \ldots, D_n\}$, where D_i is a finite set of possible values for variable x_i
- **C** is a set of constraints, each on a subset of **Z** limiting the possible combinations of values that the variables can take

Goal: to find a *consistent* variable assignment

Confession of a HK Smuggler

Take ≤ 9 units and earn ≥ 3 llars

Goods	Wine	Pe
Size	4 units	3
$$$	$15	



- Vari ain
  [0.. ly.
- Cap t: $4W + 3P + 2C \leq 9$
- Profit $15W + 10P + 7C \geq 30$

Constrained Optimization

Goods	Wine	Perfume	Cigarettes
Size	4 units	3 units	2 units
$$$	$15	$10	$7

- Constraint:
  $4W + 3P + 2C \leq 9$
  $15W + 10P + 7C \geq 30$

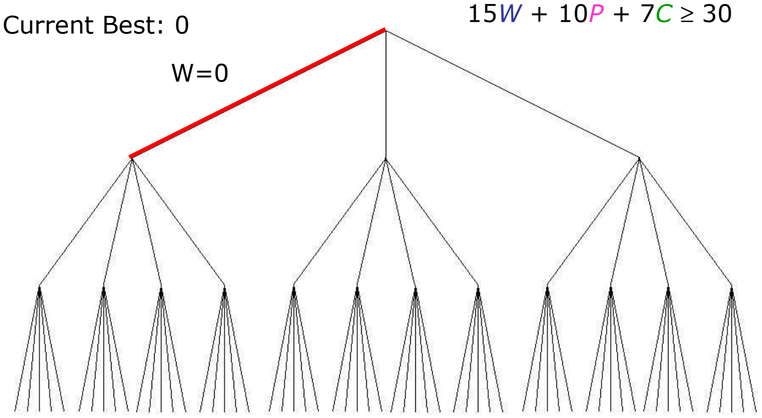
Objective: $15W + 10P + 7C$

Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 0

$4W + 3P + 2C \leq 9$
$15W + 10P + 7C \geq 30$



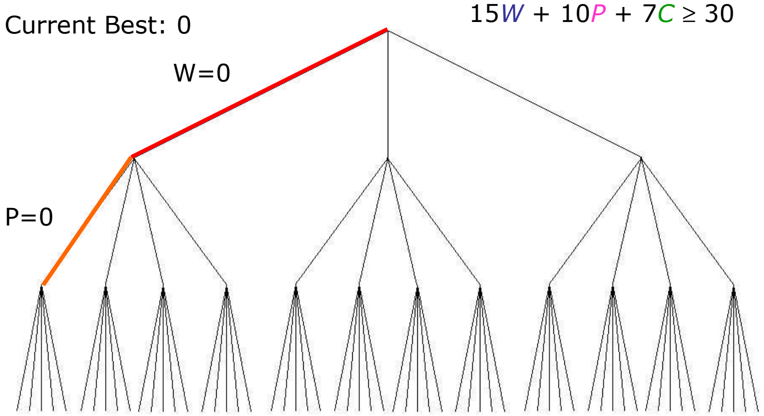
$15*0 + 10*max(P) + 7*max(C) = 0 + 30 + 28 = 58$

Basic Branch & Bound

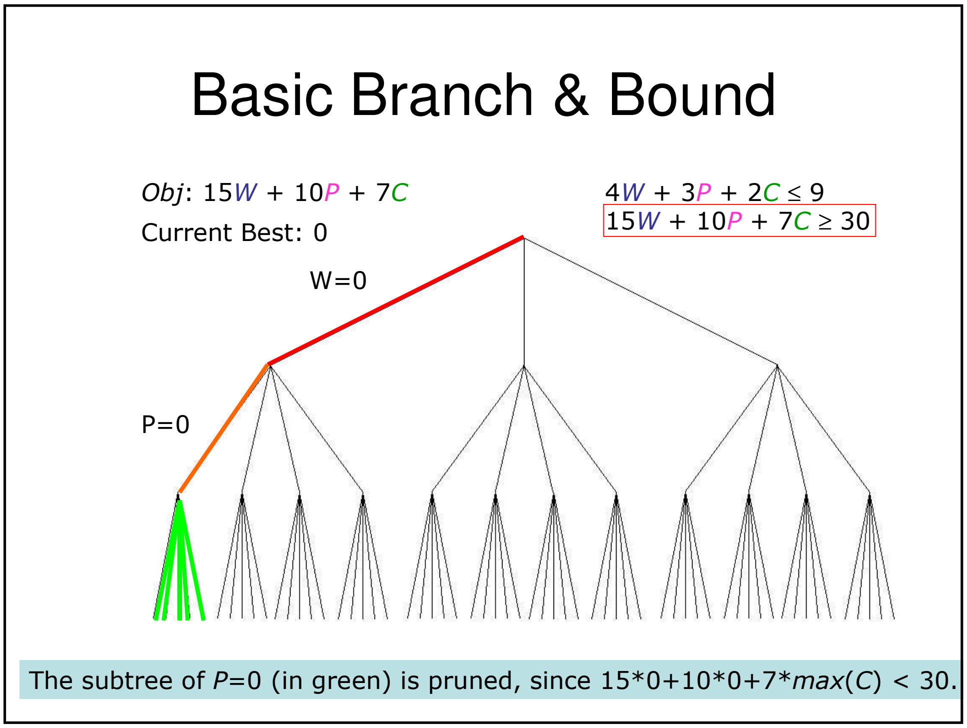
Obj: $15W + 10P + 7C$

Current Best: 0

$4W + 3P + 2C \leq 9$
$15W + 10P + 7C \geq 30$



$15*0 + 10*0 + 7*max(C) = 0 + 0 + 28 = 28$



Basic Branch & Bound

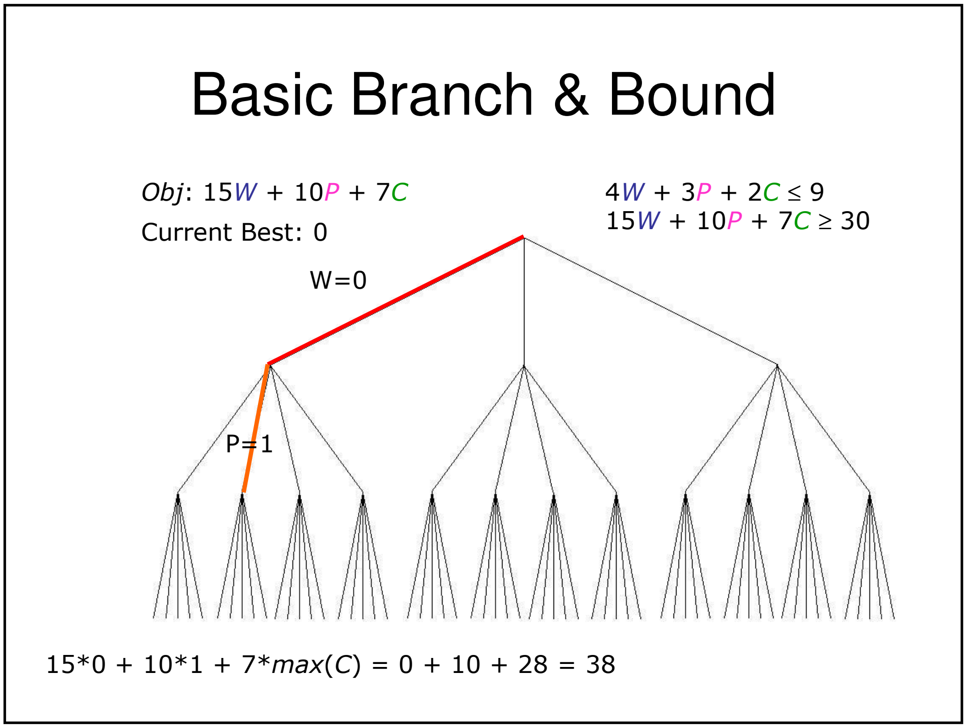
Obj: 15*W* + 10*P* + 7*C*

Current Best: 0

$4W + 3P + 2C \leq 9$

$\boxed{15W + 10P + 7C \geq 30}$

The subtree of *P*=0 (in green) is pruned, since 15*0+10*0+7**max*(*C*) < 30.



Basic Branch & Bound

Obj: 15*W* + 10*P* + 7*C*

Current Best: 0

$4W + 3P + 2C \leq 9$

$15W + 10P + 7C \geq 30$

15*0 + 10*1 + 7**max*(*C*) = 0 + 10 + 28 = 38

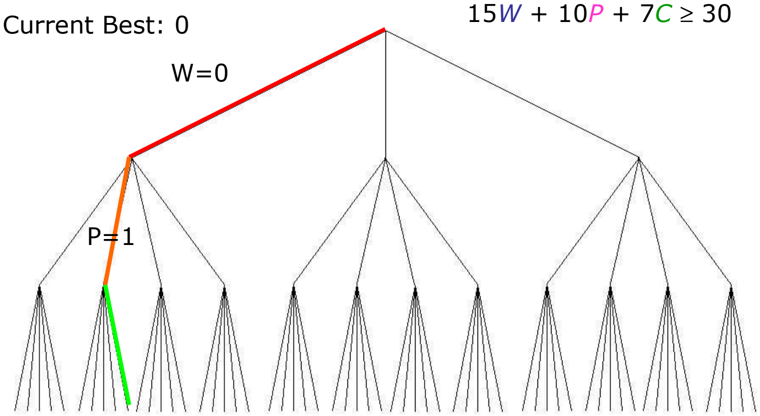
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 0

$4W + 3P + 2C \le 9$

$15W + 10P + 7C \ge 30$



The subtree of C=4 (in green) is pruned, since 4*0+3*1+2*4 > 9.

Now *max*(C) = 3

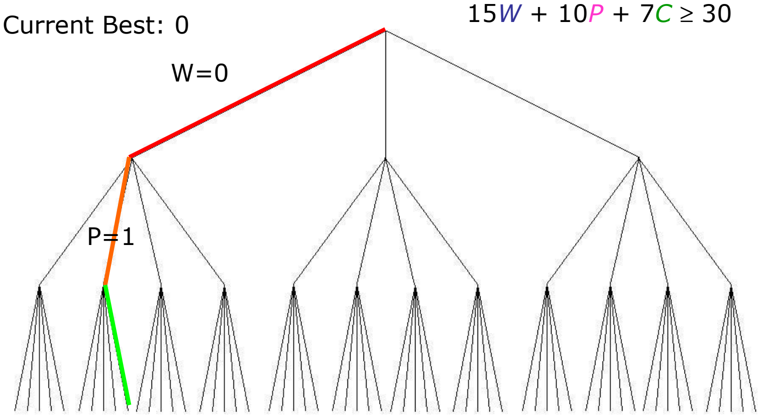
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 0

$4W + 3P + 2C \le 9$

$15W + 10P + 7C \ge 30$



15*0 + 10*1 + 7**max*(C) = 0 + 10 + 21 = 31

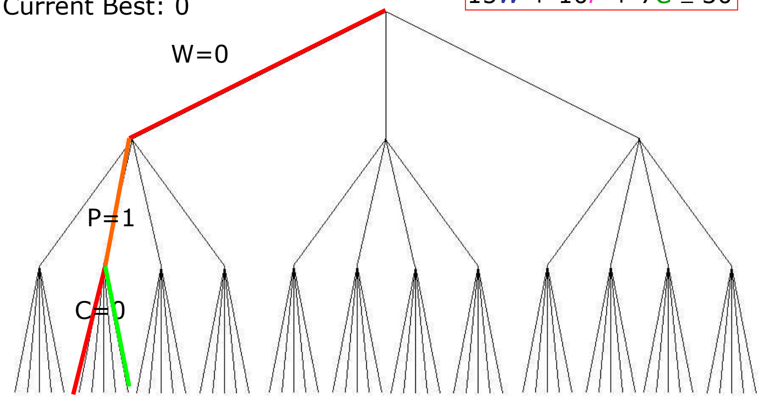
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 0

$4W + 3P + 2C \leq 9$

$\boxed{15W + 10P + 7C \geq 30}$



15*0 + 10*1 + 7*0 = 0 + 10 + 0 = 10 < 30

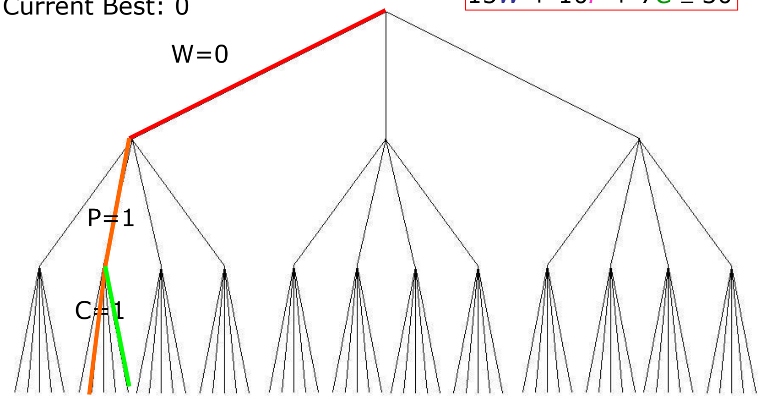
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 0

$4W + 3P + 2C \leq 9$

$\boxed{15W + 10P + 7C \geq 30}$



15*0 + 10*1 + 7*1 = 0 + 10 + 7 = 17 < 30

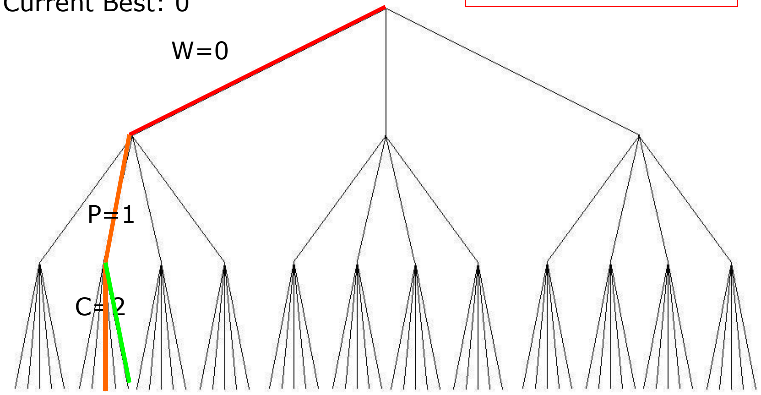
Basic Branch & Bound

Obj: 15*W* + 10*P* + 7*C*

Current Best: 0

4*W* + 3*P* + 2*C* ≤ 9

$\boxed{15W + 10P + 7C \geq 30}$



15*0 + 10*1 + 7*2 = 0 + 10 + 14 = 24 < 30

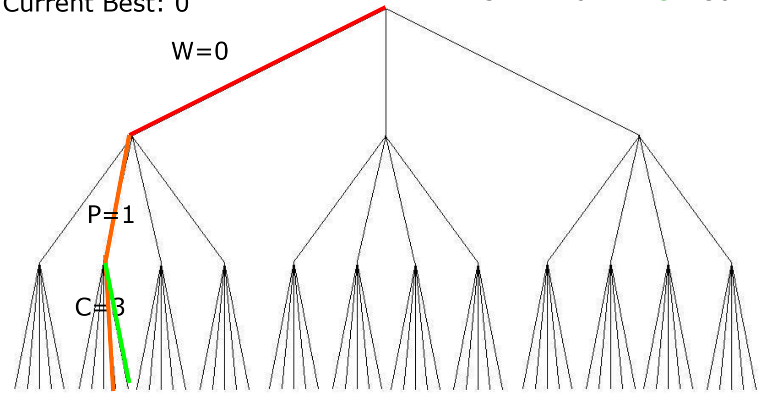
Basic Branch & Bound

Obj: 15*W* + 10*P* + 7*C*

Current Best: 0

4*W* + 3*P* + 2*C* ≤ 9

15*W* + 10*P* + 7*C* ≥ 30



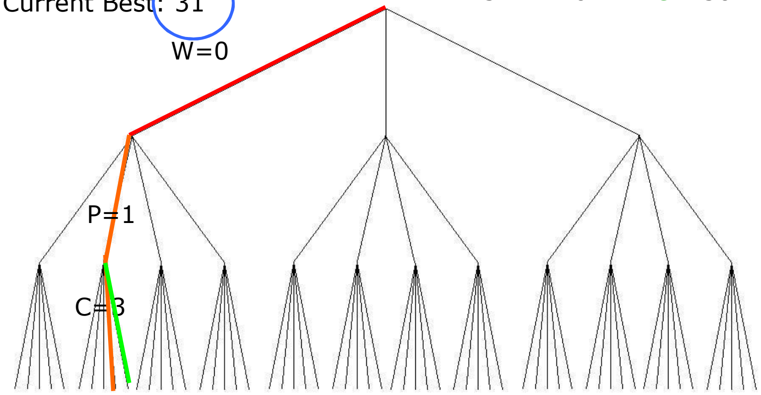
15*0 + 10*1 + 7*3 = 0 + 10 + 21 = 31

Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \le 9$
$15W + 10P + 7C \ge 30$



$15*0 + 10*1 + 7*3 = 0 + 10 + 21 = 31$

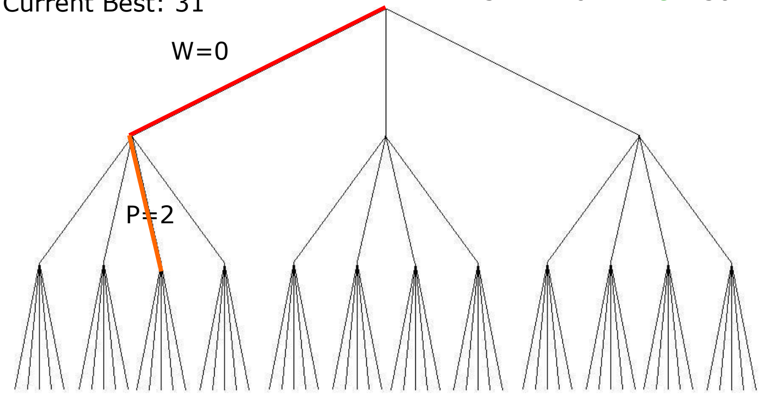
Since *obj* > current best value, current value is updated to 31.

Basic Branch & Bound

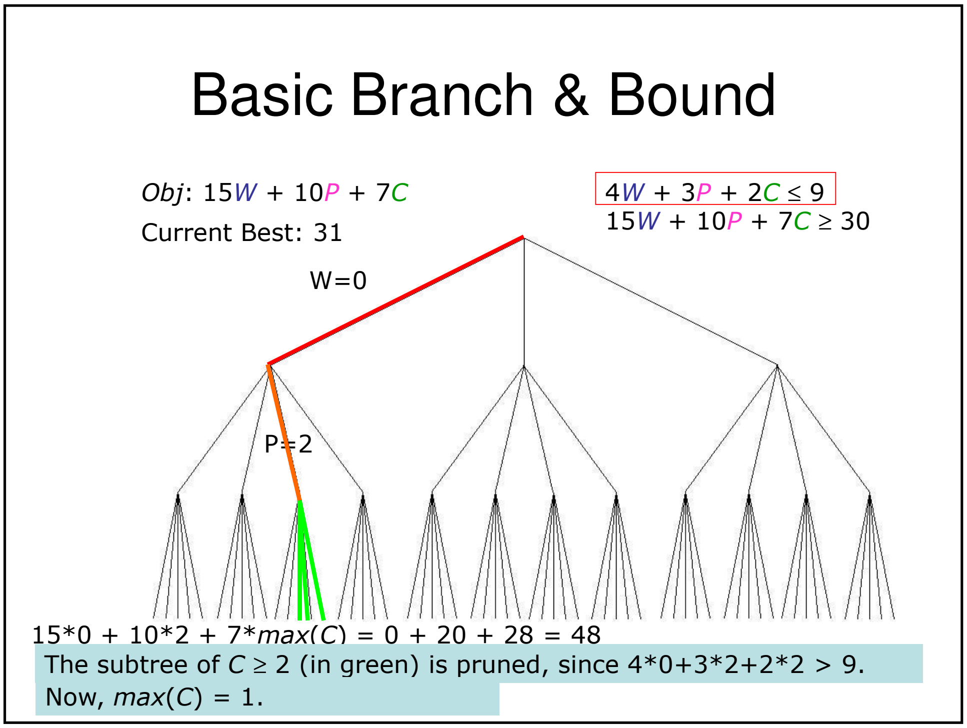
Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \le 9$
$15W + 10P + 7C \ge 30$



$15*0 + 10*2 + 7*max(C) = 0 + 20 + 28 = 48$



Basic Branch & Bound

Obj: $15W + 10P + 7C$

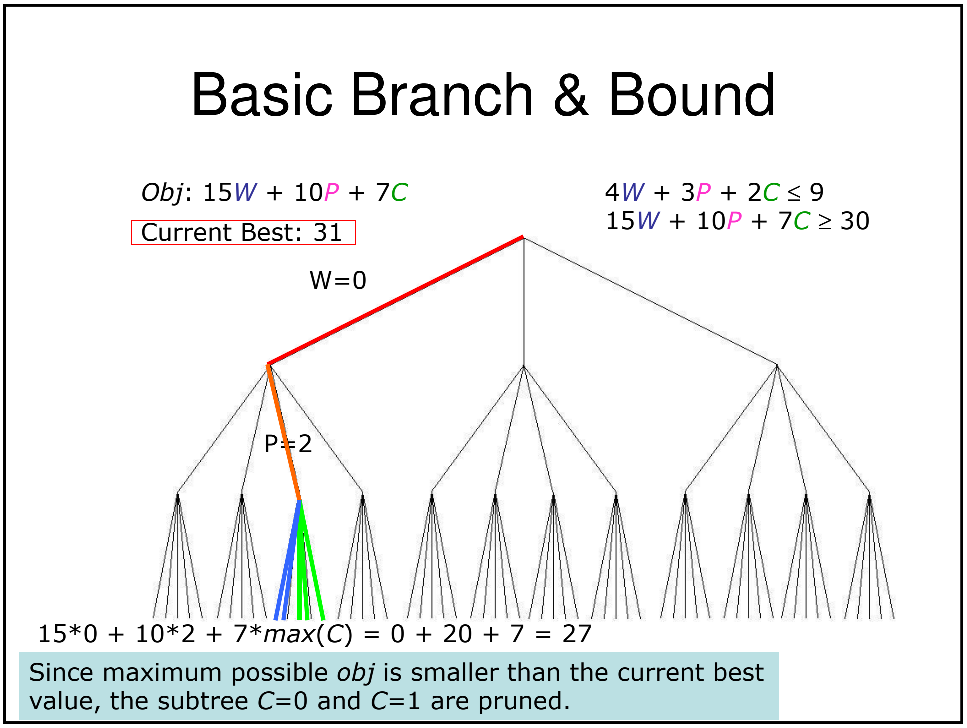
Current Best: 31

$4W + 3P + 2C \leq 9$

$15W + 10P + 7C \geq 30$

$15*0 + 10*2 + 7*max(C) = 0 + 20 + 28 = 48$

The subtree of $C \geq 2$ (in green) is pruned, since $4*0+3*2+2*2 > 9$.
Now, $max(C) = 1$.



Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \leq 9$

$15W + 10P + 7C \geq 30$

$15*0 + 10*2 + 7*max(C) = 0 + 20 + 7 = 27$

Since maximum possible *obj* is smaller than the current best value, the subtree $C=0$ and $C=1$ are pruned.

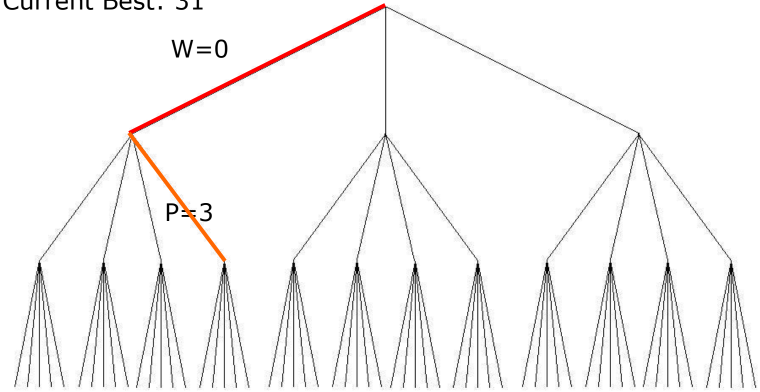
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \le 9$

$15W + 10P + 7C \ge 30$



$15*0 + 10*3 + 7*max(C) = 0 + 30 + 28 = 58$

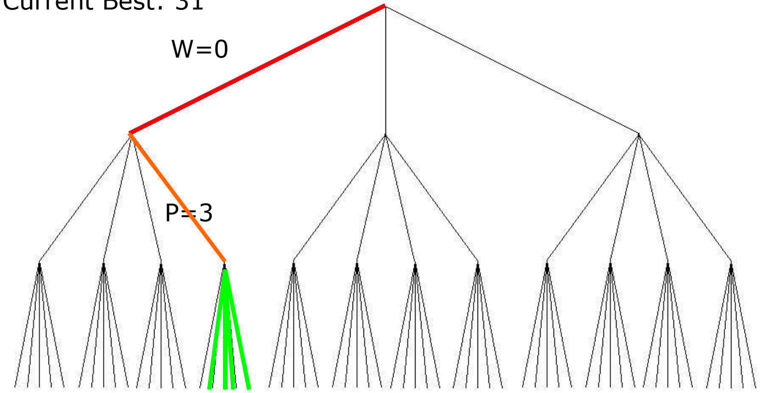
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$\boxed{4W + 3P + 2C \le 9}$

$15W + 10P + 7C \ge 30$



$15*0 + 10*3 + 7*max(C) = 0 + 30 + 28 = 58$

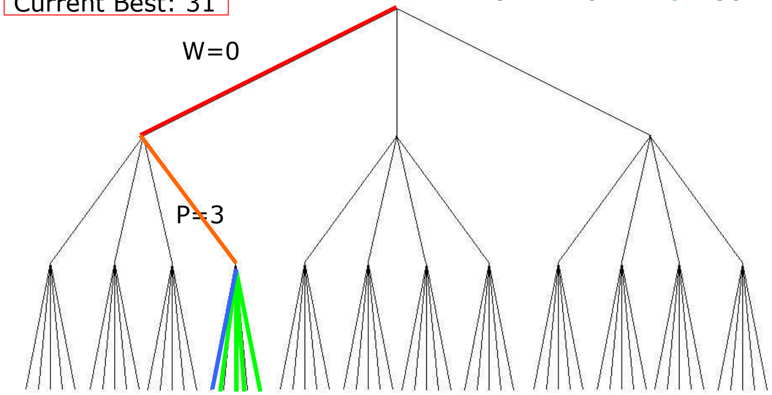
The subtree of $C \ge 1$ (in green) is pruned, since $4*0+3*3+2*1 > 9$.
Now, $max(C) = 0$.

Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \le 9$
$15W + 10P + 7C \ge 30$



$15*0 + 10*3 + 7*max(C) = 0 + 30 + 0 = 30$

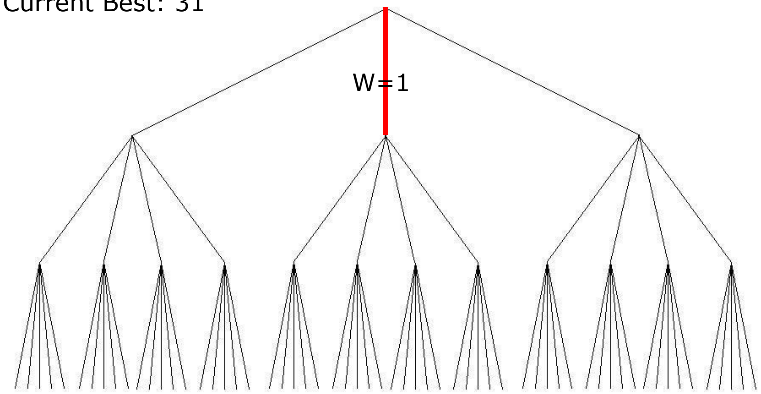
Since maximum possible *obj* is smaller than the current best value, the subtree *C*=0 is pruned.

Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \le 9$
$15W + 10P + 7C \ge 30$



$15*1 + 10*max(P) + 7*max(C) = 15 + 30 + 28 = 73$

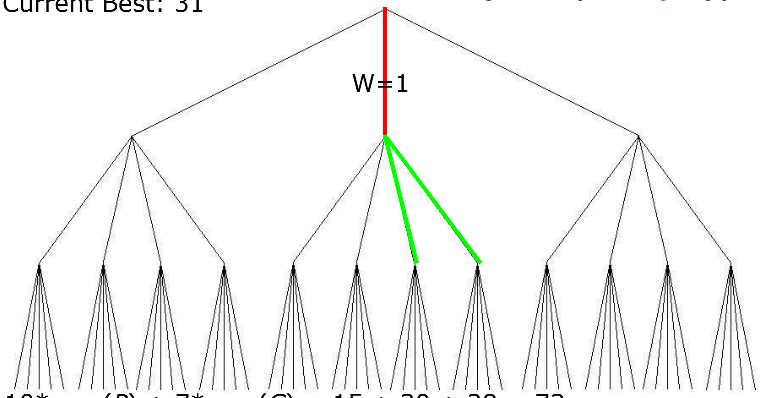
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \le 9$

$15W + 10P + 7C \ge 30$



$15*1 + 10*max(P) + 7*max(C) = 15 + 30 + 28 = 73$

The subtree $P \ge 2$ is pruned, since 4*1+3*2 > 9.

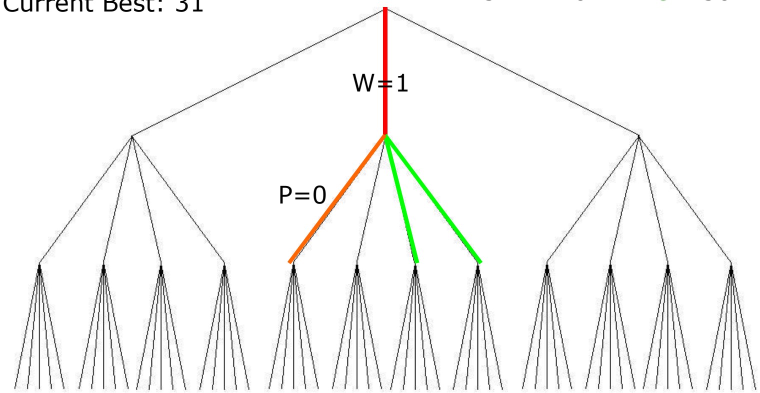
Basic Branch & Bound

Obj: $15W + 10P + 7C$

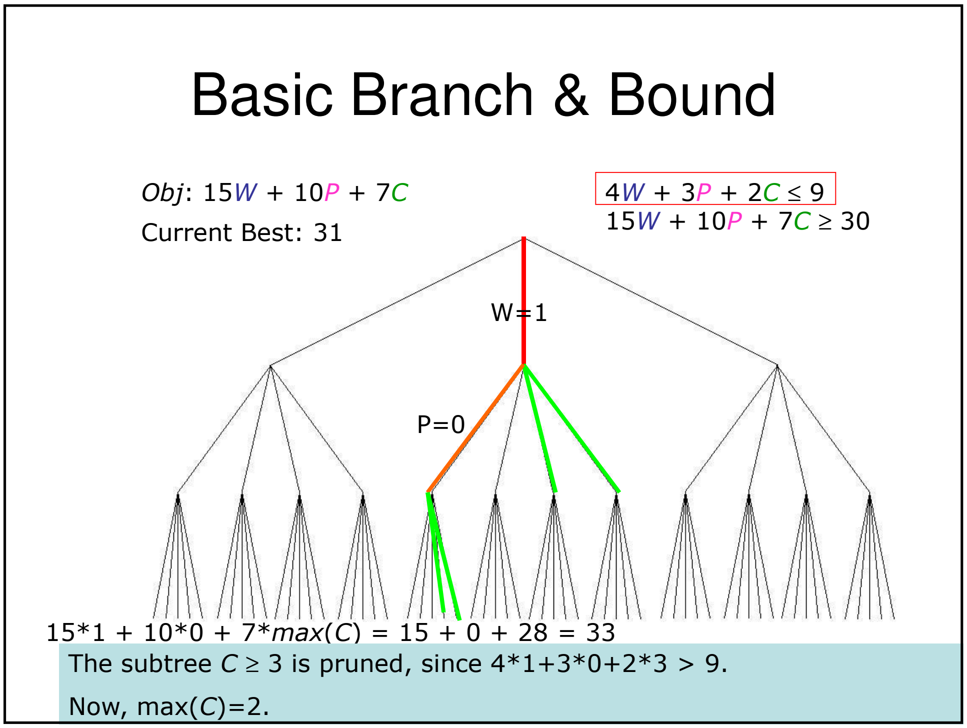
Current Best: 31

$4W + 3P + 2C \le 9$

$15W + 10P + 7C \ge 30$



$15*1 + 10*0 + 7*max(C) = 15 + 0 + 28 = 33$



Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

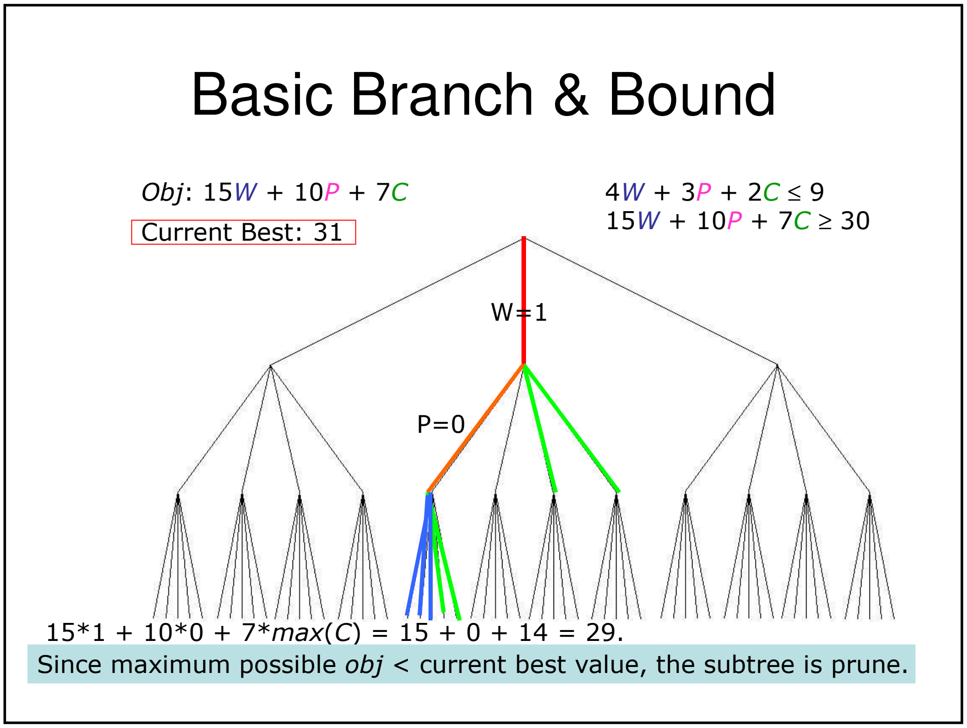
$4W + 3P + 2C \leq 9$

$15W + 10P + 7C \geq 30$

$15*1 + 10*0 + 7*max(C) = 15 + 0 + 28 = 33$

The subtree $C \geq 3$ is pruned, since 4*1+3*0+2*3 > 9.

Now, max(C)=2.



Basic Branch & Bound

Obj: $15W + 10P + 7C$

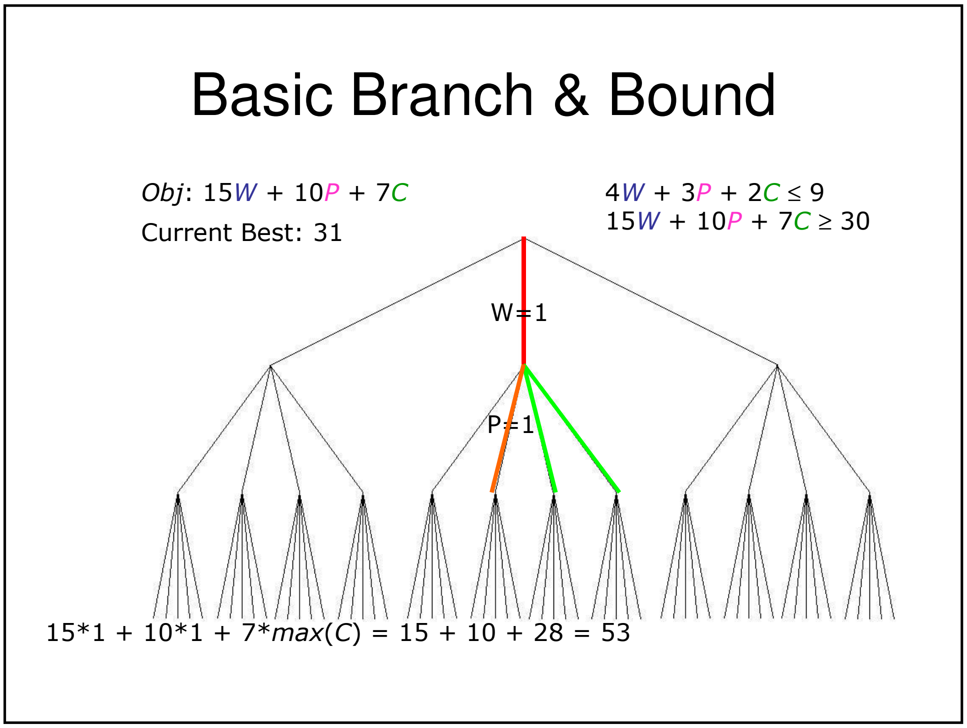
Current Best: 31

$4W + 3P + 2C \leq 9$

$15W + 10P + 7C \geq 30$

$15*1 + 10*0 + 7*max(C) = 15 + 0 + 14 = 29.$

Since maximum possible *obj* < current best value, the subtree is prune.



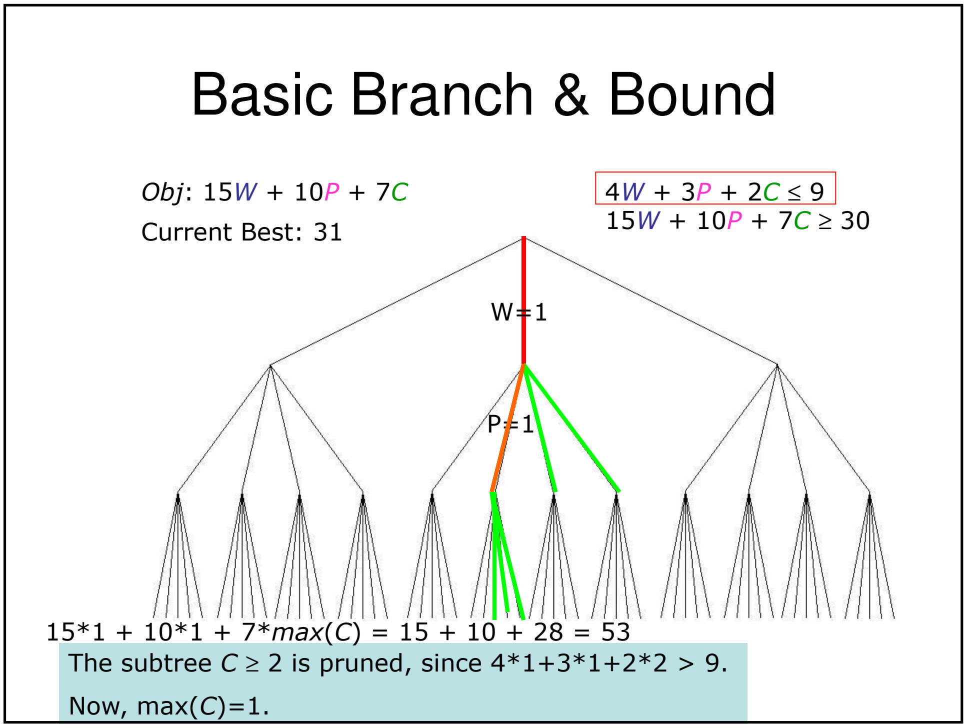
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$4W + 3P + 2C \le 9$
$15W + 10P + 7C \ge 30$

$15*1 + 10*1 + 7*max(C) = 15 + 10 + 28 = 53$



Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 31

$\boxed{4W + 3P + 2C \le 9}$
$15W + 10P + 7C \ge 30$

$15*1 + 10*1 + 7*max(C) = 15 + 10 + 28 = 53$

The subtree $C \ge 2$ is pruned, since $4*1+3*1+2*2 > 9$.

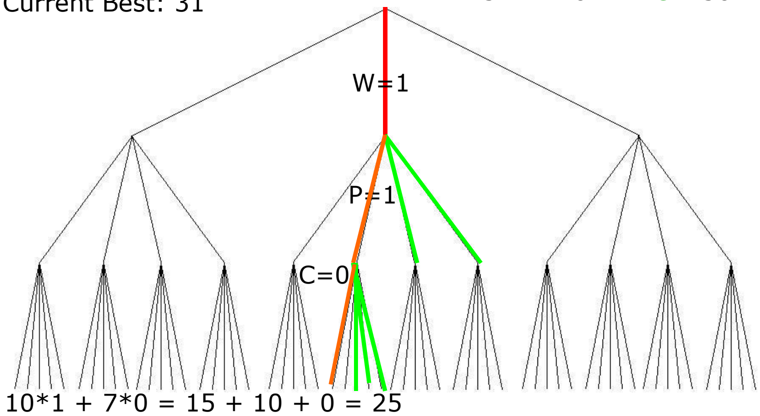
Now, max(C)=1.

Basic Branch & Bound

Obj: 15*W* + 10*P* + 7*C*

Current Best: 31

4*W* + 3*P* + 2*C* ≤ 9
15*W* + 10*P* + 7*C* ≥ 30



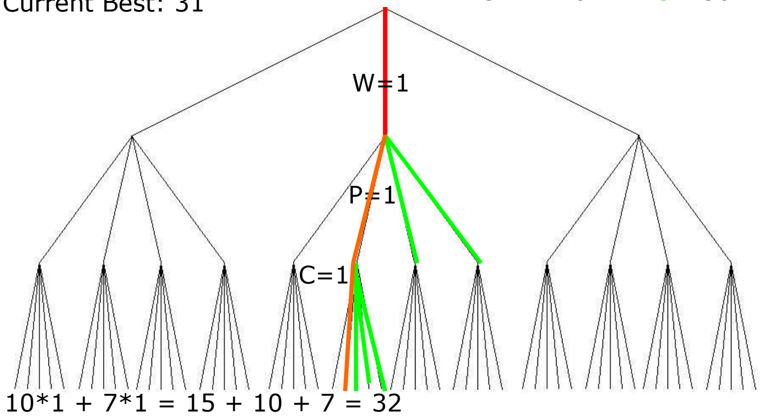
15*1 + 10*1 + 7*0 = 15 + 10 + 0 = 25

Basic Branch & Bound

Obj: 15*W* + 10*P* + 7*C*

Current Best: 31

4*W* + 3*P* + 2*C* ≤ 9
15*W* + 10*P* + 7*C* ≥ 30



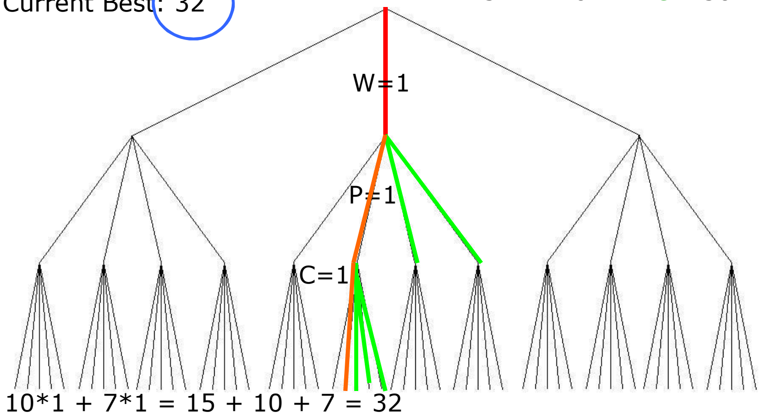
15*1 + 10*1 + 7*1 = 15 + 10 + 7 = 32

Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 32

$4W + 3P + 2C \le 9$
$15W + 10P + 7C \ge 30$



15*1 + 10*1 + 7*1 = 15 + 10 + 7 = 32

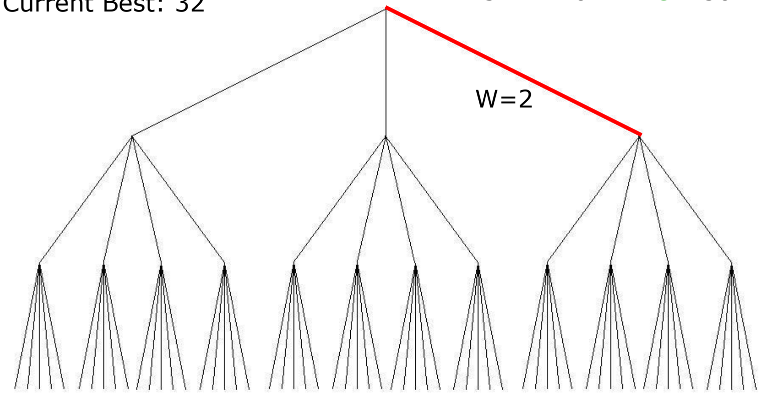
Since *obj* > current best value, current value is updated to 32.

Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 32

$4W + 3P + 2C \le 9$
$15W + 10P + 7C \ge 30$



15*2 + 10**max*(*P*) + 7**max*(*C*) = 30 + 30 + 28 = 88

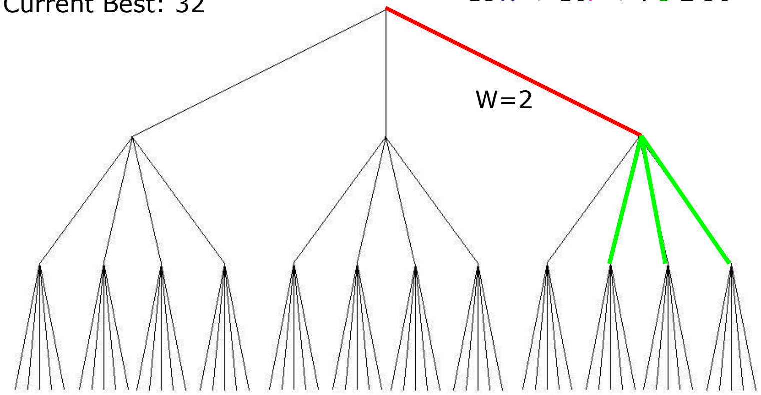
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 32

$4W + 3P + 2C \leq 9$

$15W + 10P + 7C \geq 30$



$15*2 + 10*max(P) + 7*max(C) = 30 + 30 + 28 = 88$

The subtree $P \geq 1$ is pruned, since $4*2 + 3*1 > 9$. Now $max(P)=0$.

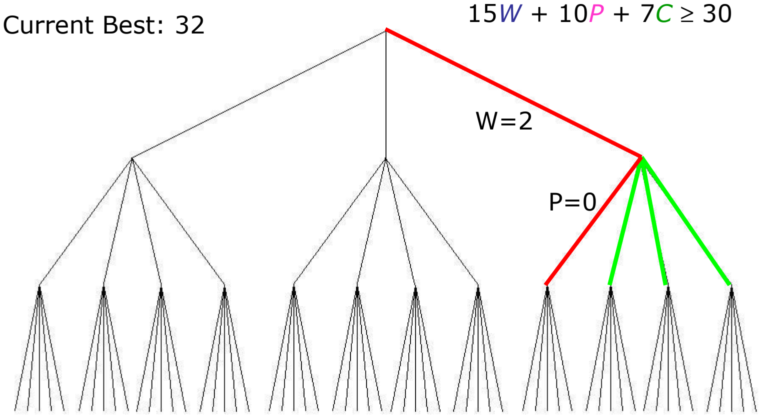
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 32

$4W + 3P + 2C \leq 9$

$15W + 10P + 7C \geq 30$



$15*2 + 10*0 + 7*max(C) = 30 + 0 + 28 = 58$

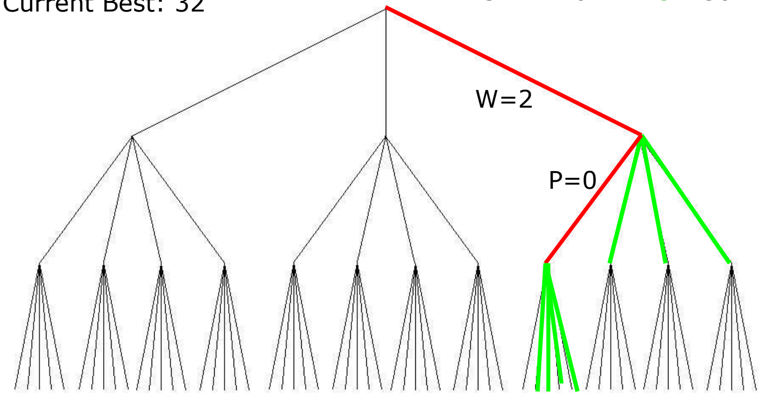
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 32

$\boxed{4W + 3P + 2C \le 9}$

$15W + 10P + 7C \ge 30$



$15*2 + 10*0 + 7*max(C) = 30 + 0 + 28 = 58$

The subtree $C \ge 1$ is pruned, since $4*2 + 3*0 + 2*1 > 9$. Now $max(C)=0$.

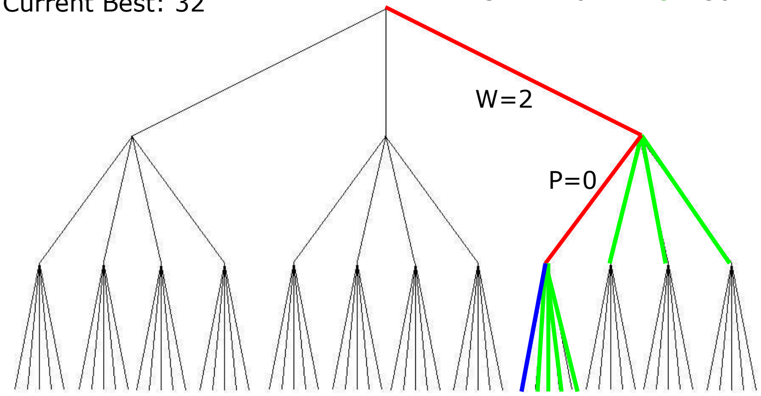
Basic Branch & Bound

Obj: $15W + 10P + 7C$

Current Best: 32

$4W + 3P + 2C \le 9$

$15W + 10P + 7C \ge 30$



$15*2 + 10*0 + 7*max(\text{C}) = 30 + 0 + 0 = 30$

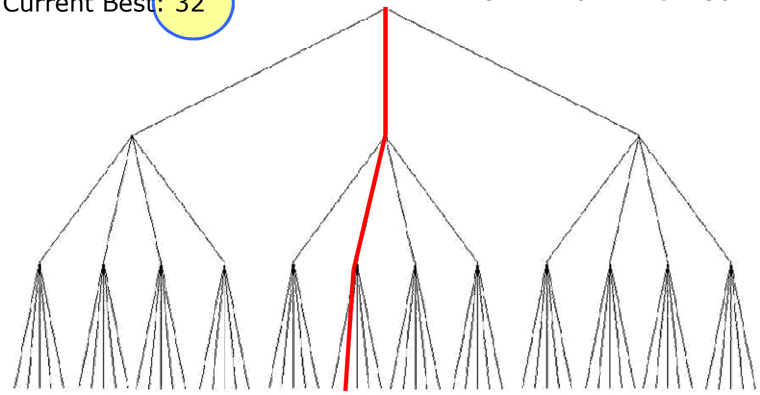
Since obj < current best value, subtree C=0 is also pruned.

Basic Branch & Bound

Obj: 15*W* + 10*P* + 7*C*

Current Best: 32

4*W* + 3*P* + 2*C* ≤ 9
15*W* + 10*P* + 7*C* ≥ 30



Search completed. Optimal solution is {*W*=1, *P*=1, *C*=1}.

Search Efficiency

- Variable Ordering
  - Affects the shape and size of the search tree
  - Is first-fail principle all??

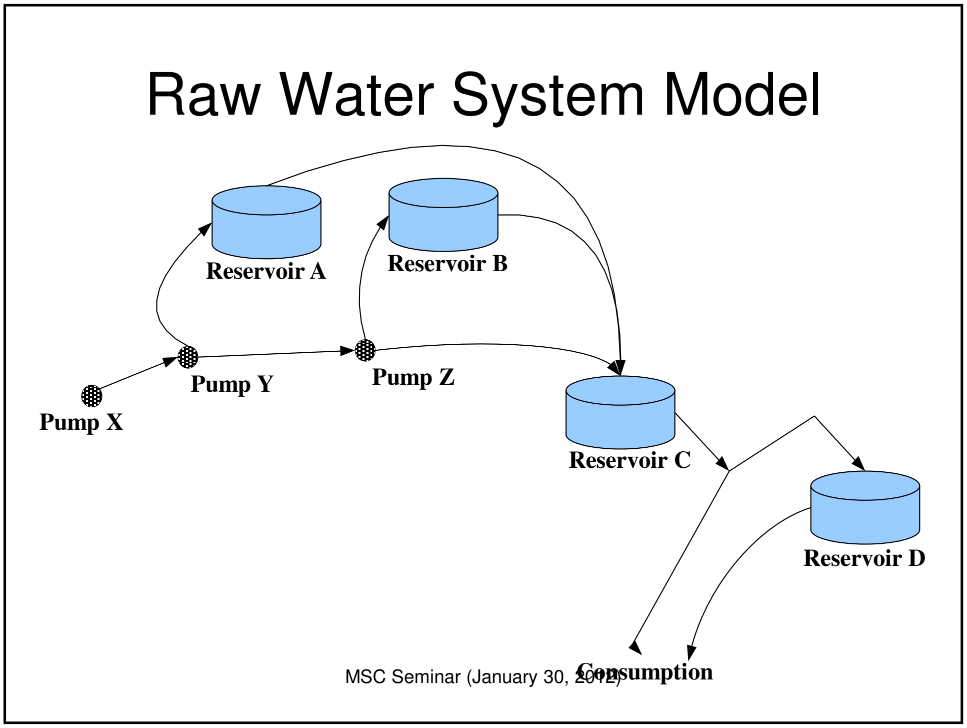
- Value Ordering
  - Affects the ordering of the branches
  - Move (good) solution branches as far to the left as possible (depth-first search from left to right)

- Constraint propagation: the more the better!

Outline

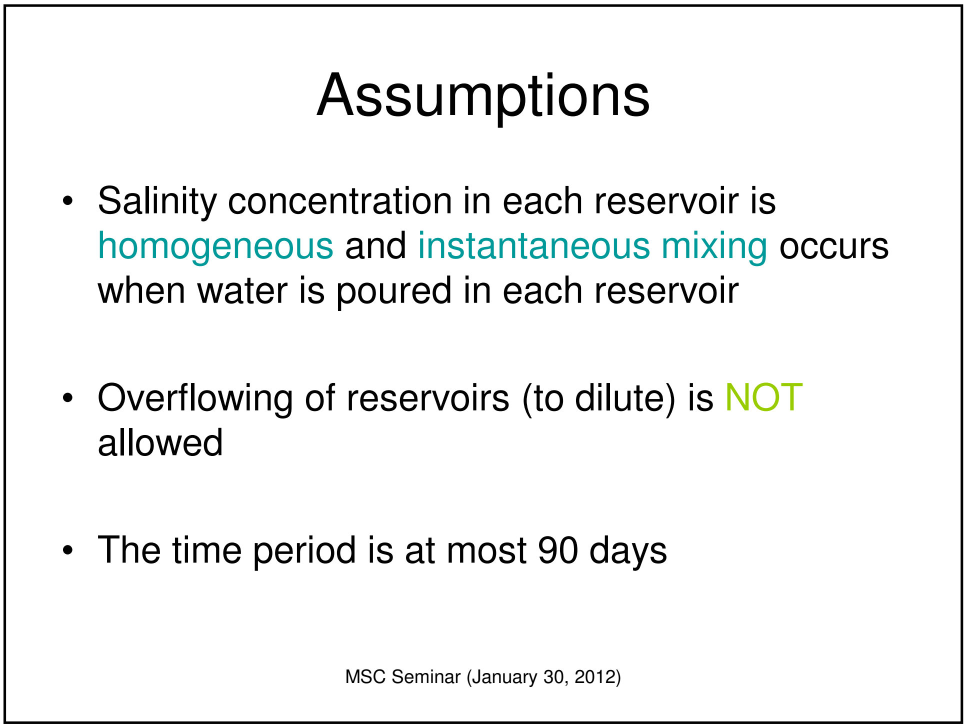
- Domain Description
- Constraint Programming (CP)
- Problem Modelling
- Improvements
- Concluding Remarks

MSC Seminar (January 30, 2012)



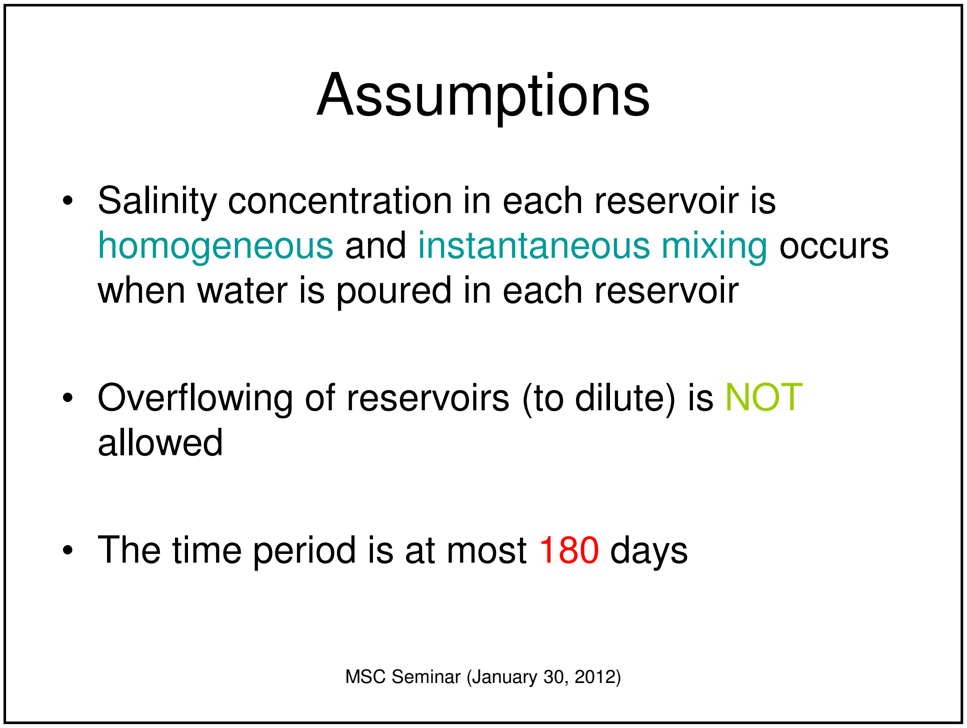
Raw Water System Model

MSC Seminar (January 30, 2012)



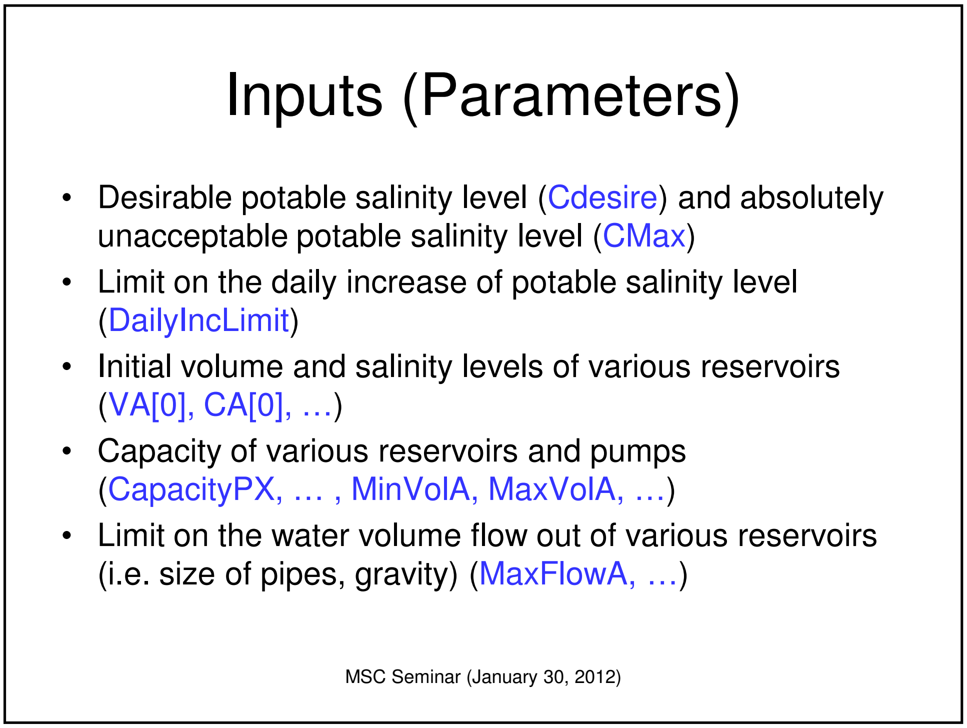
Assumptions

- Salinity concentration in each reservoir is homogeneous and instantaneous mixing occurs when water is poured in each reservoir
- Overflowing of reservoirs (to dilute) is NOT allowed
- The time period is at most 90 days



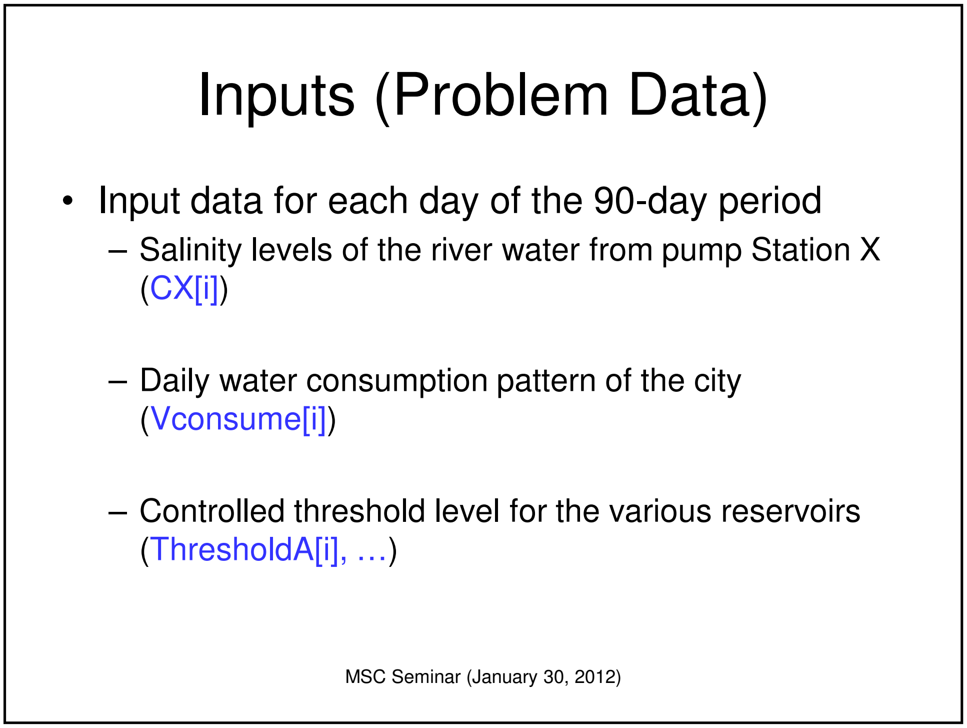
Assumptions

- Salinity concentration in each reservoir is homogeneous and instantaneous mixing occurs when water is poured in each reservoir
- Overflowing of reservoirs (to dilute) is NOT allowed
- The time period is at most 180 days



Inputs (Parameters)

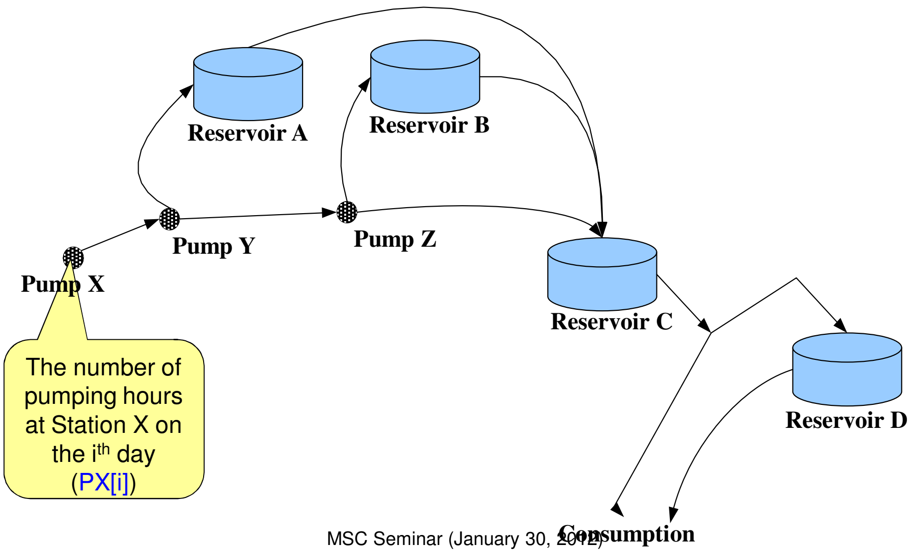
- Desirable potable salinity level (Cdesire) and absolutely unacceptable potable salinity level (CMax)
- Limit on the daily increase of potable salinity level (DailyIncLimit)
- Initial volume and salinity levels of various reservoirs (VA[0], CA[0], …)
- Capacity of various reservoirs and pumps (CapacityPX, … , MinVolA, MaxVolA, …)
- Limit on the water volume flow out of various reservoirs (i.e. size of pipes, gravity) (MaxFlowA, …)



Inputs (Problem Data)

- Input data for each day of the 90-day period
  - Salinity levels of the river water from pump Station X (CX[i])
  - Daily water consumption pattern of the city (Vconsume[i])
  - Controlled threshold level for the various reservoirs (ThresholdA[i], …)

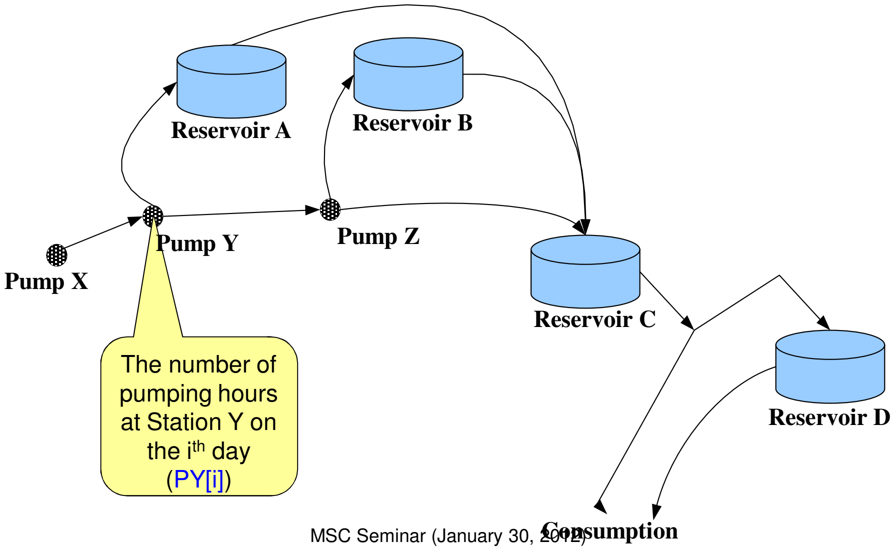
Decision Variables



The number of pumping hours at Station X on the i^{th} day (PX[i])

MSC Seminar (January 30, 2012)

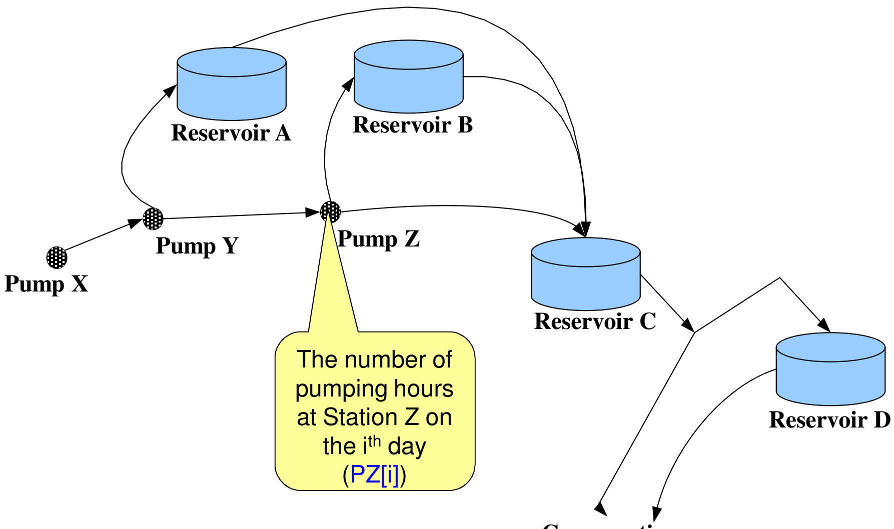
Decision Variables



The number of pumping hours at Station Y on the i^{th} day (PY[i])

MSC Seminar (January 30, 2012)

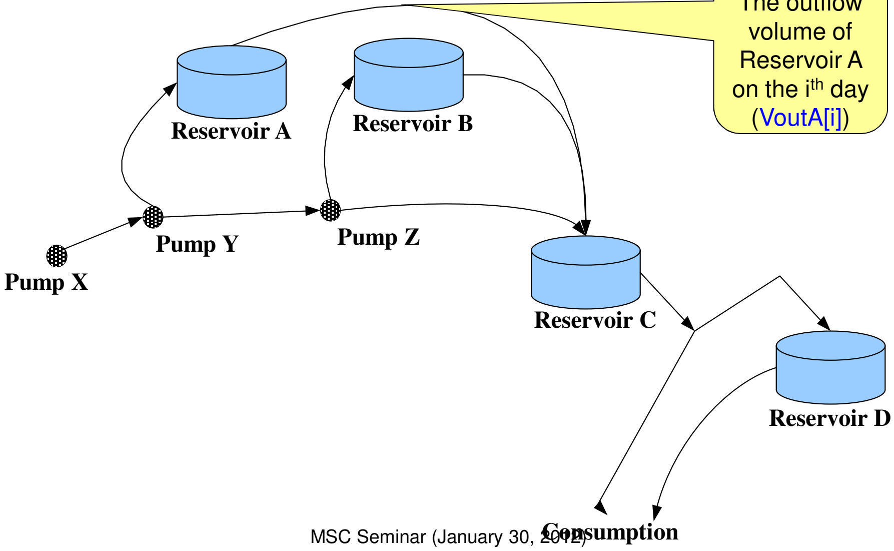
Decision Variables



The number of pumping hours at Station Z on the i^{th} day (PZ[i])

MSC Seminar (January 30, 2012)

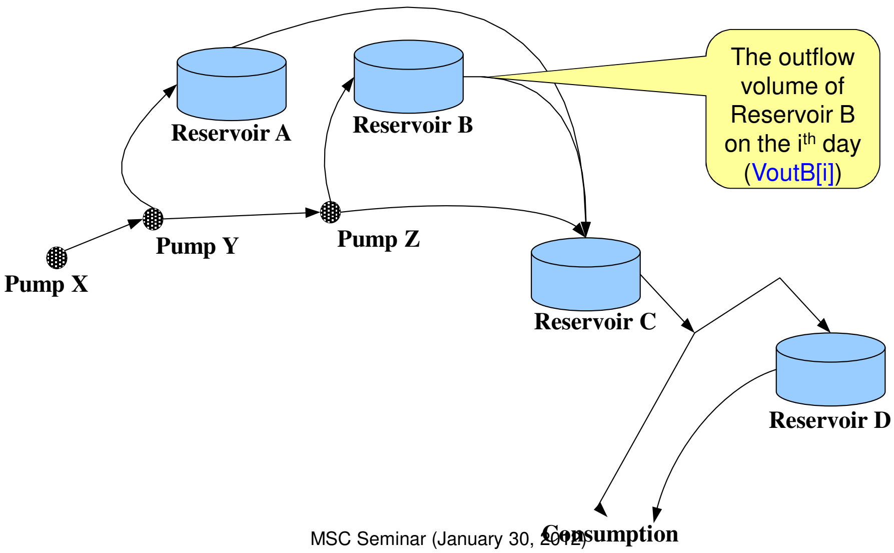
Decision Variables



The outflow volume of Reservoir A on the i^{th} day (VoutA[i])

MSC Seminar (January 30, 2012)

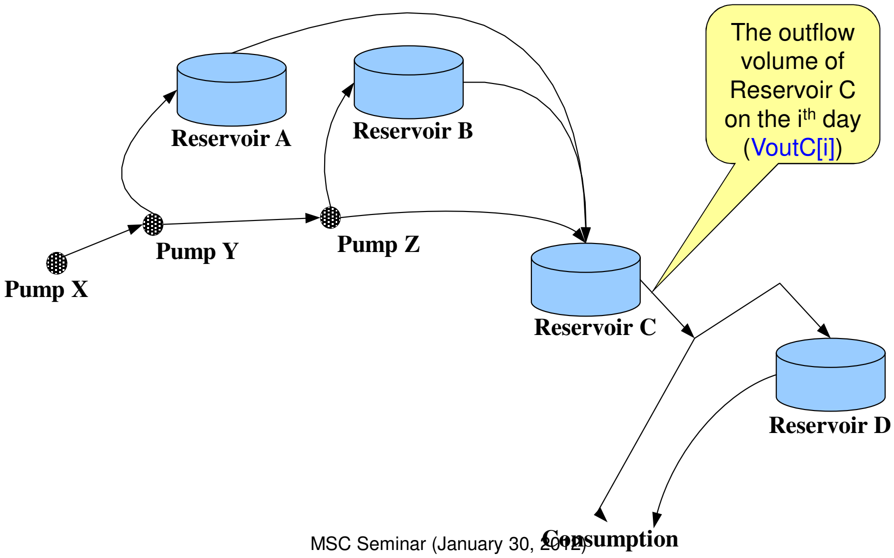
Decision Variables



The outflow volume of Reservoir B on the i^{th} day (VoutB[i])

MSC Seminar (January 30, 2012)

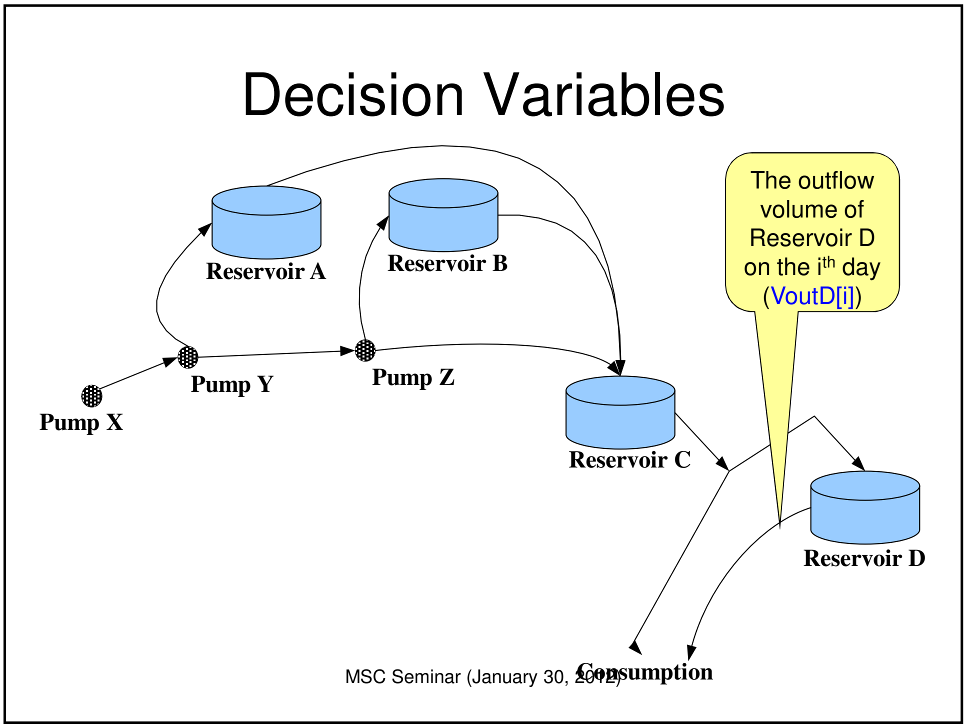
Decision Variables



The outflow volume of Reservoir C on the i^{th} day (VoutC[i])

MSC Seminar (January 30, 2012)

Decision Variables



The outflow volume of Reservoir D on the i^{th} day (VoutD[i])

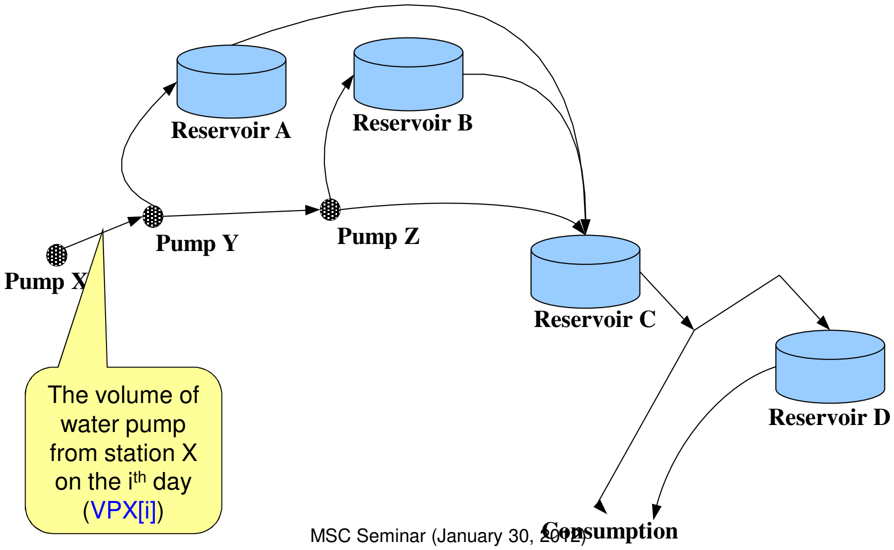
Domain Discretization

- Domain of the Decision variables (pumping hours and water transfer) are continuous in nature
- Discretize and quantize the domain to reduce the search space
  - PumpQuanta
    - Pump usage quanta/unit (e.g. 3 hrs or 6 hrs)
  - TransferQuanta
    - Water transfer quanta/unit (e.g. 5,000m^3 or 10,000m^3)

Domains of Decision Variables

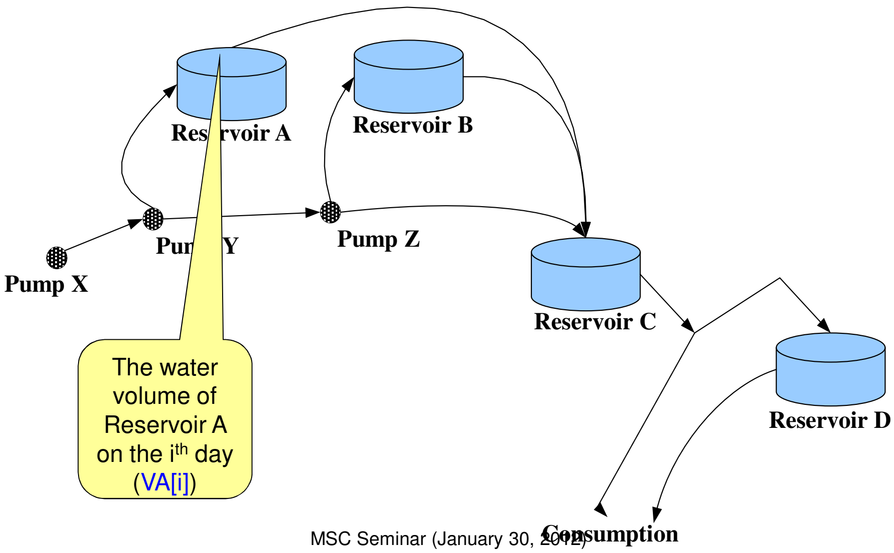
PX[i]	[0 .. (NumOfPumpX * 24 / PumpQuanta)]
PY[i]	[0 .. (NumOfPumpY * 24 / PumpQuanta)]
PZ[i]	[0 .. (NumOfPumpZ * 24 / PumpQuanta)]
VoutA[i]	[0 .. ⌊MaxFlowA / TransferQuanta⌋]
VoutB[i]	[0 .. ⌊MaxFlowB / TransferQuanta⌋]
VoutC[i]	[0 .. ⌊MaxFlowC / TransferQuanta⌋]
VoutD[i]	[0 .. ⌊MaxFlowD / TransferQuanta⌋]

Auxiliary Variables



The volume of water pump from station X on the i^{th} day (VPX[i])

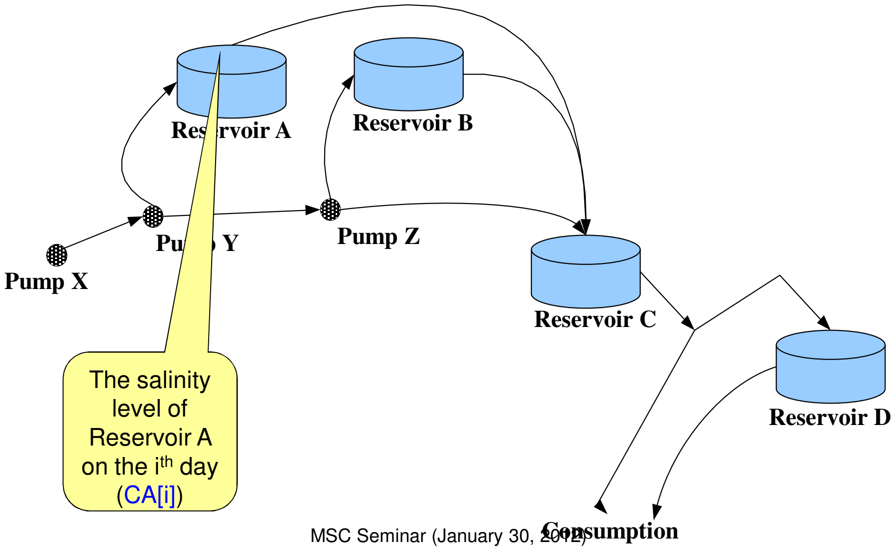
Auxiliary Variables



The water volume of Reservoir A on the i^{th} day (VA[i])

MSC Seminar (January 30, 2012)

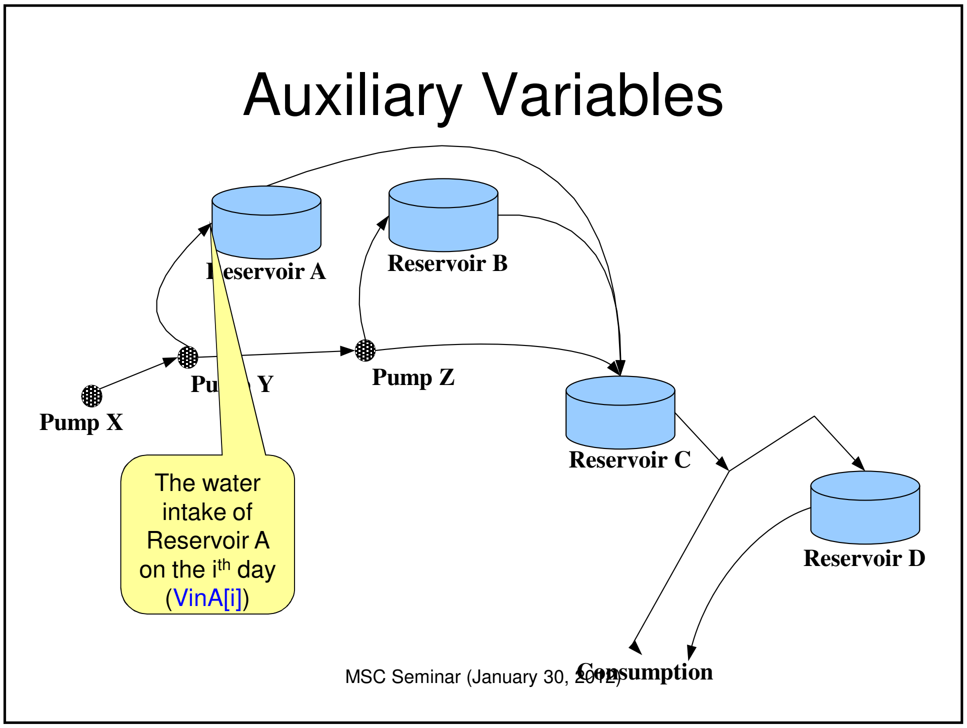
Auxiliary Variables



The salinity level of Reservoir A on the i^{th} day (CA[i])

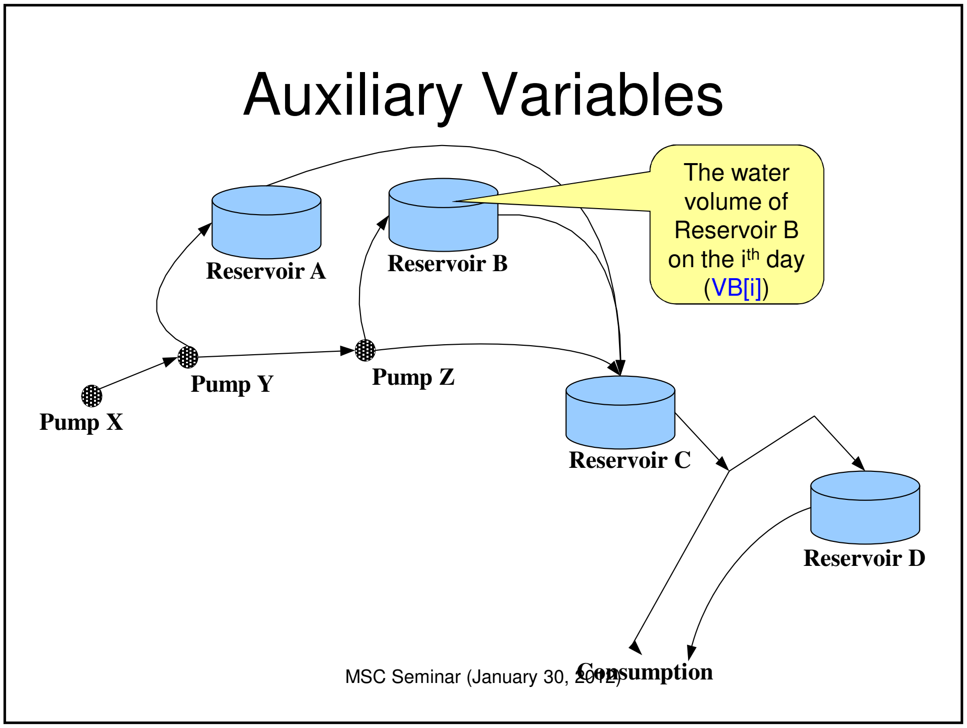
MSC Seminar (January 30, 2012)

Auxiliary Variables



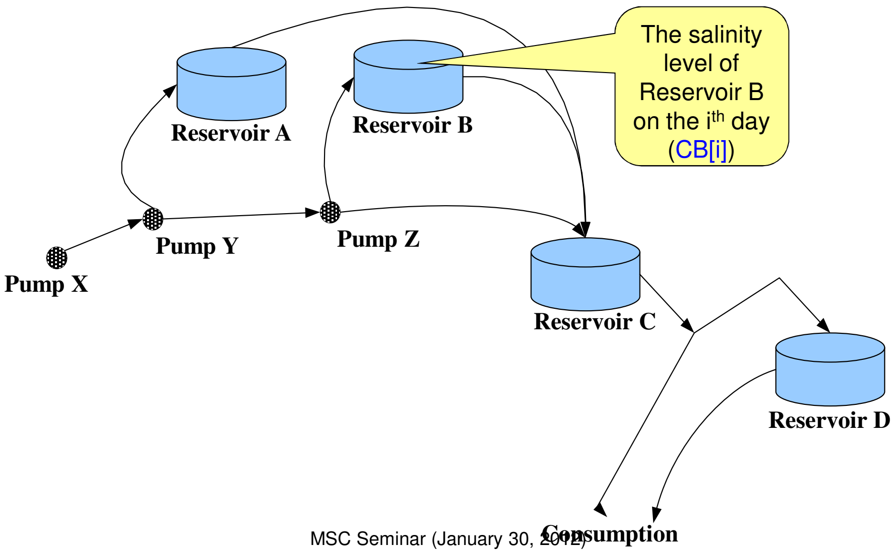
The water intake of Reservoir A on the i^{th} day (VinA[i])

Auxiliary Variables



The water volume of Reservoir B on the i^{th} day (VB[i])

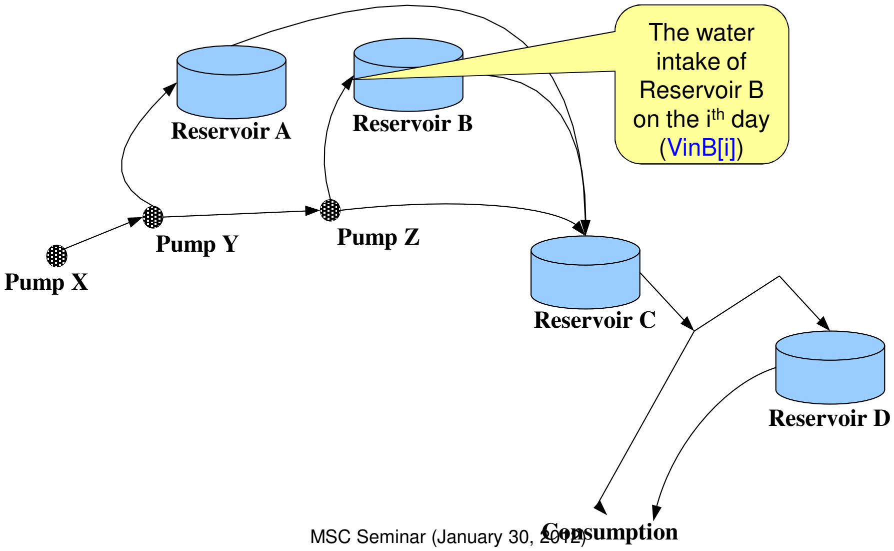
Auxiliary Variables



The salinity level of Reservoir B on the i^th^ day (CB[i])

MSC Seminar (January 30, 2012)

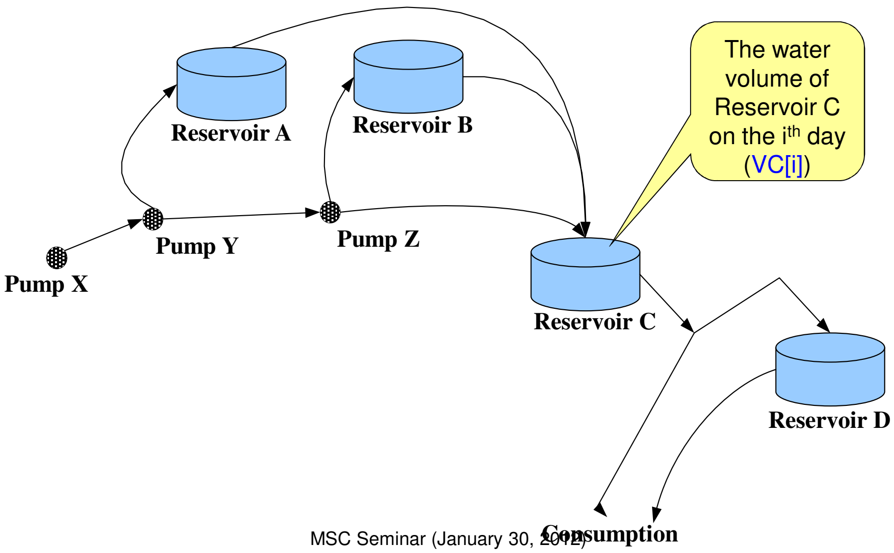
Auxiliary Variables



The water intake of Reservoir B on the i^{th} day (VinB[i])

MSC Seminar (January 30, 2012)

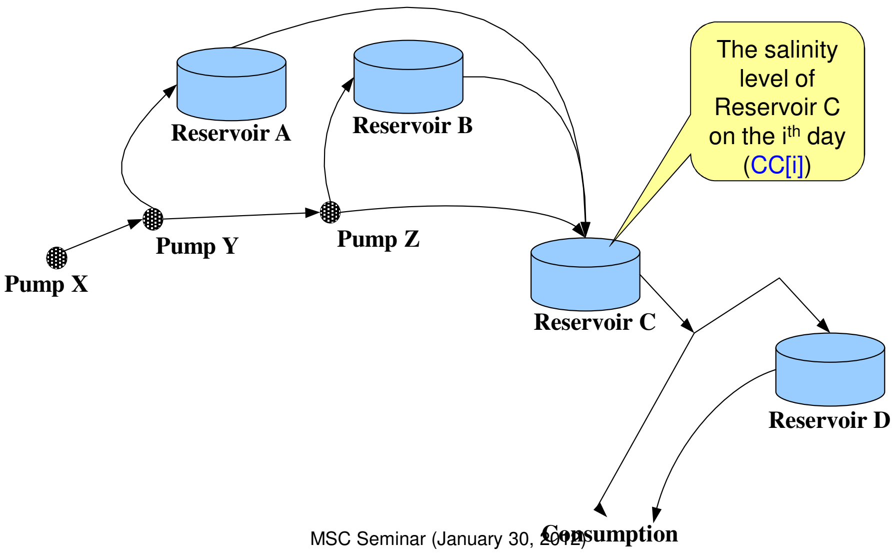
Auxiliary Variables



The water volume of Reservoir C on the i^{th} day (VC[i])

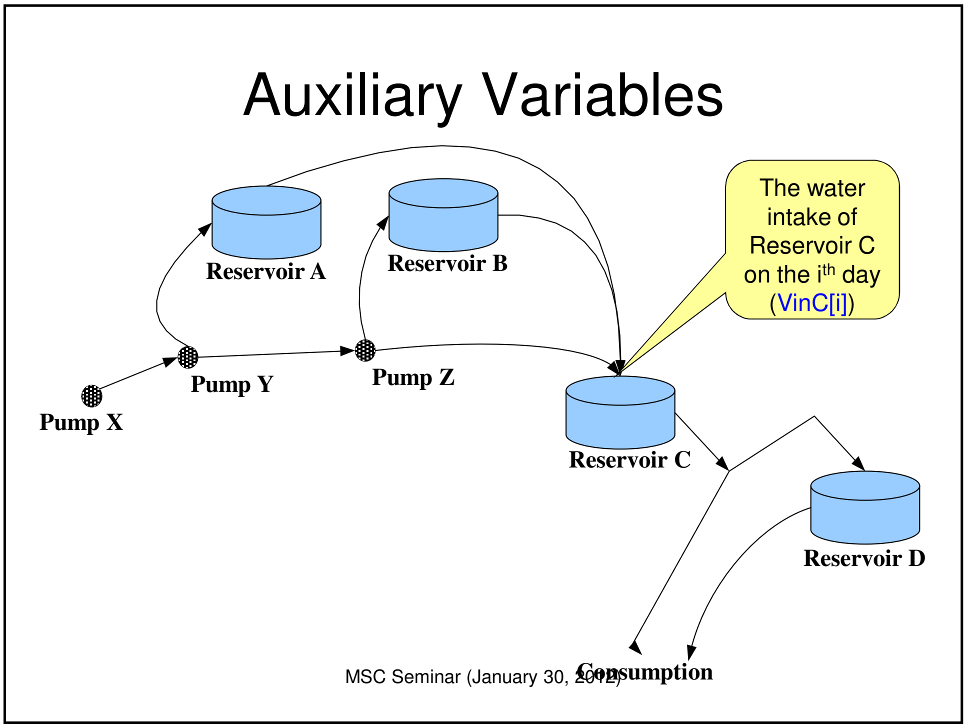
MSC Seminar (January 30, 2012)

Auxiliary Variables



The salinity level of Reservoir C on the i^{th} day (CC[i])

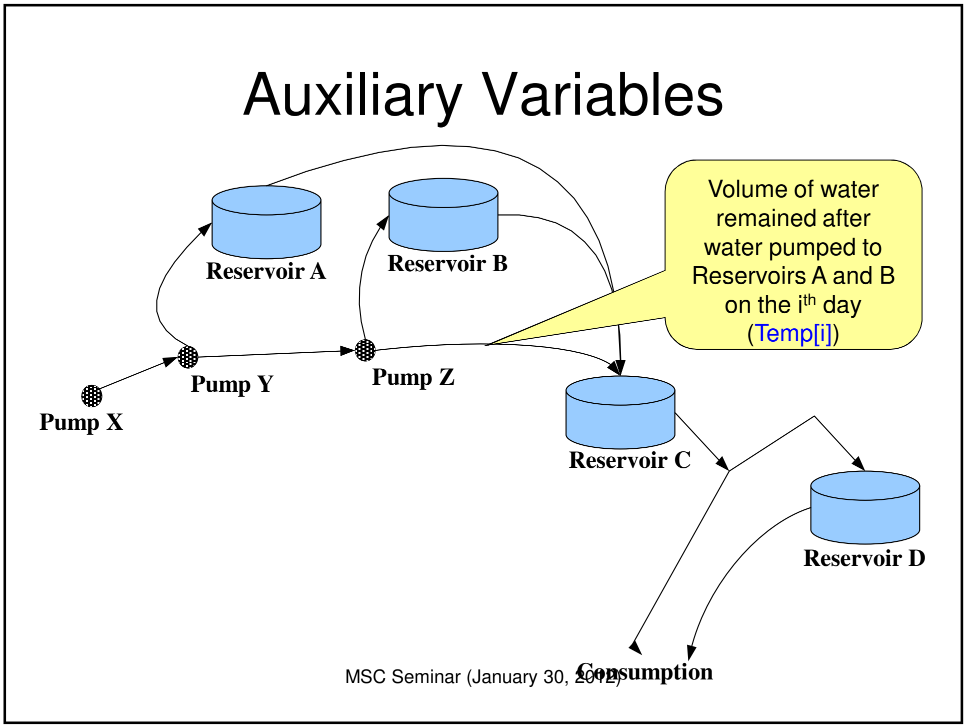
MSC Seminar (January 30, 2012)



Auxiliary Variables

The water intake of Reservoir C on the i^{th} day (VinC[i])

MSC Seminar (January 30, 2012)

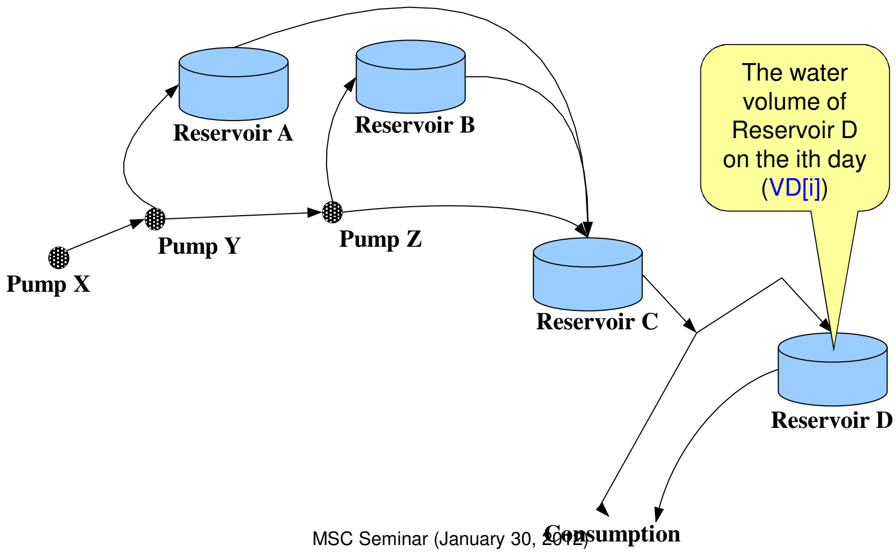


Auxiliary Variables

Volume of water remained after water pumped to Reservoirs A and B on the i^{th} day (Temp[i])

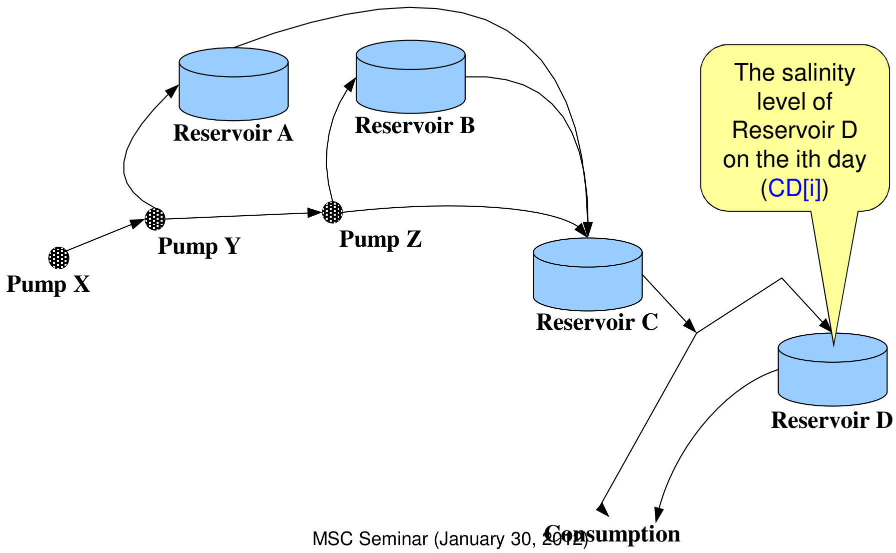
MSC Seminar (January 30, 2012)

Auxiliary Variables



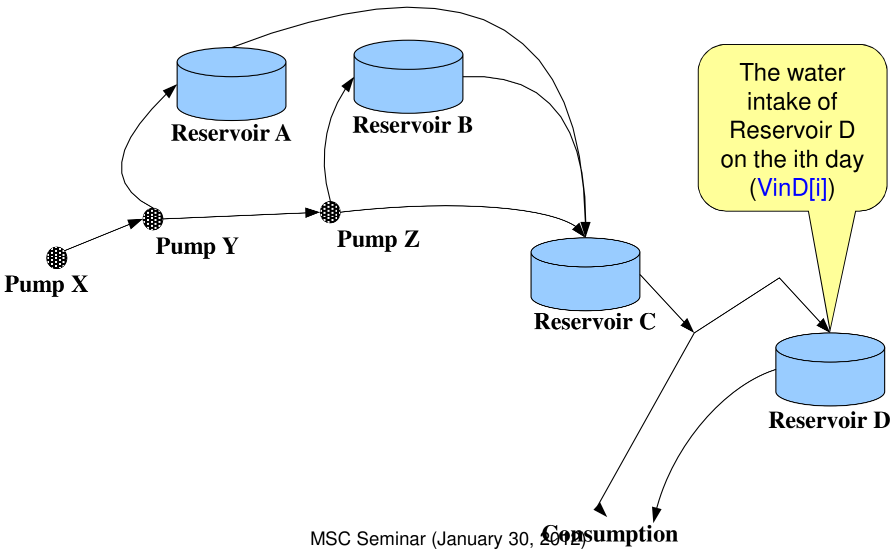
The water volume of Reservoir D on the ith day (VD[i])

Auxiliary Variables



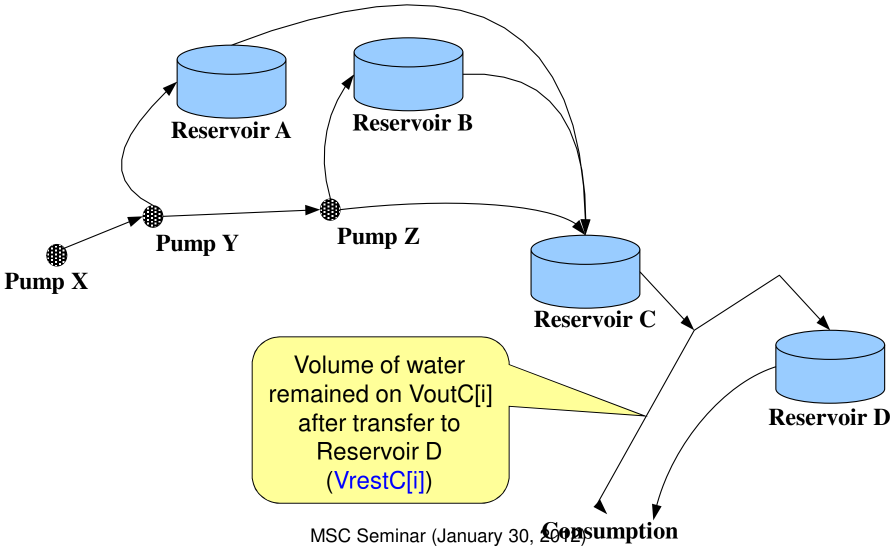
The salinity level of Reservoir D on the ith day (CD[i])

Auxiliary Variables



The water intake of Reservoir D on the ith day (VinD[i])

Auxiliary Variables



Volume of water remained on VoutC[i] after transfer to Reservoir D (VrestC[i])

Constraints

- Law of conservation of water and salts
- Physical Limitations
- Consumer Satisfaction

First Type of Constraints

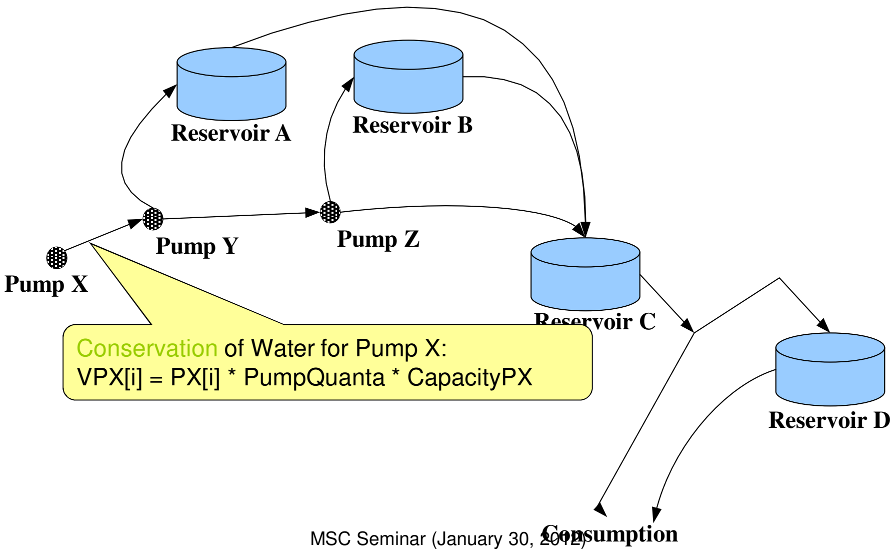
- Law of conservation of water of a reservoir (linear relation)

$$\text{Reservoir Volume day (i)} = \text{Reservoir Volume day (i-1)} - \text{Volume flow-out day (i)} + \text{Volume flow-in day (i)}$$

- Law of conservation of salts of a reservoir (nonlinear relation)

$$\text{Reservoir Volume day (i)} \times \text{Reservoir Salinity day (i)} = \text{Reservoir Volume day (i-1)} \times \text{Reservoir Salinity day (i-1)} - \text{Volume flow-out day (i)} \times \text{Reservoir Salinity day (i-1)} + \text{Volume flow-in day (i)} \times \text{Salinity flow-in}$$

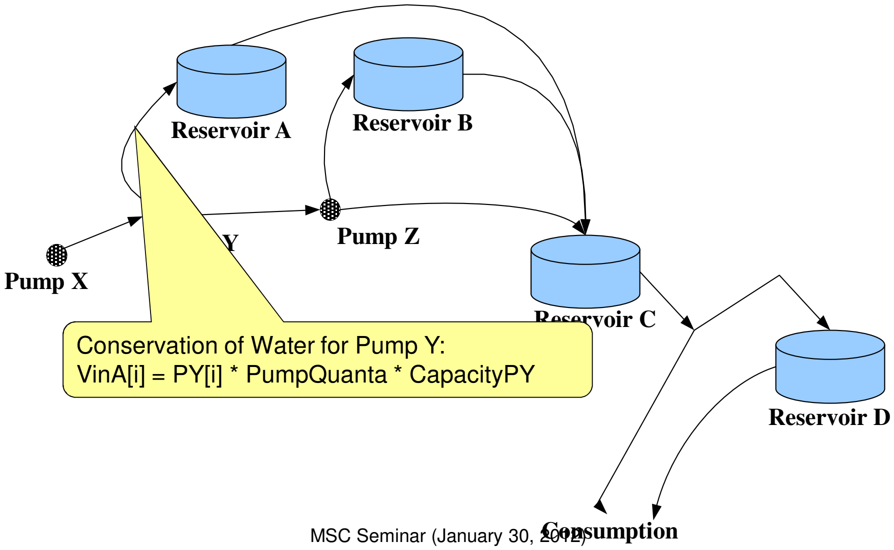
Constraints



Conservation of Water for Pump X:
VPX[i] = PX[i] * PumpQuanta * CapacityPX

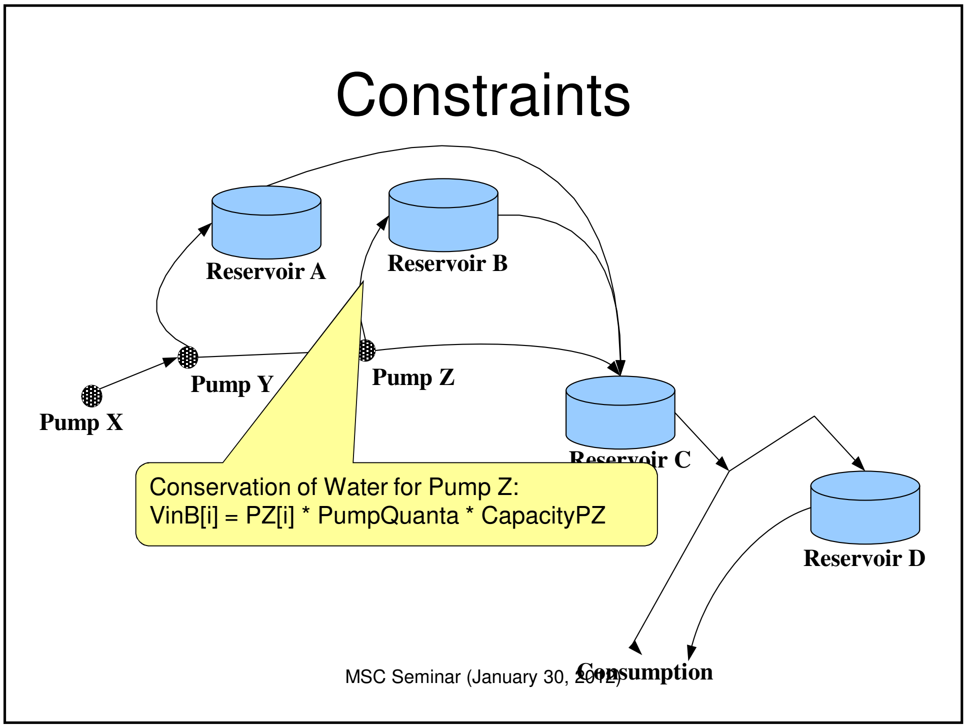
MSC Seminar (January 30, 2012)

Constraints



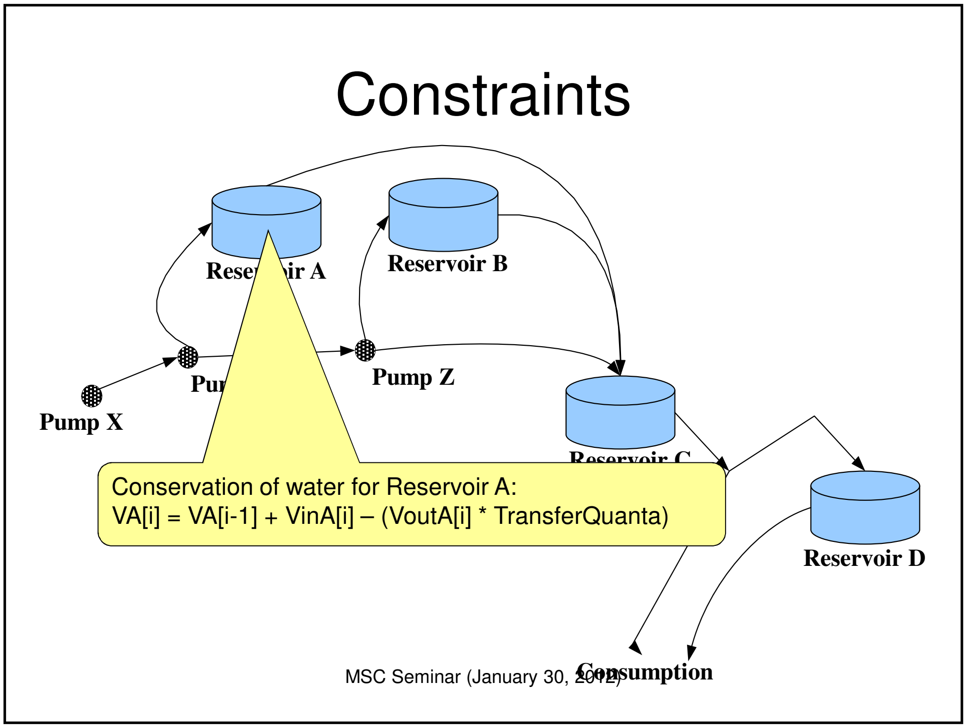
Conservation of Water for Pump Y:
VinA[i] = PY[i] * PumpQuanta * CapacityPY

MSC Seminar (January 30, 2012)



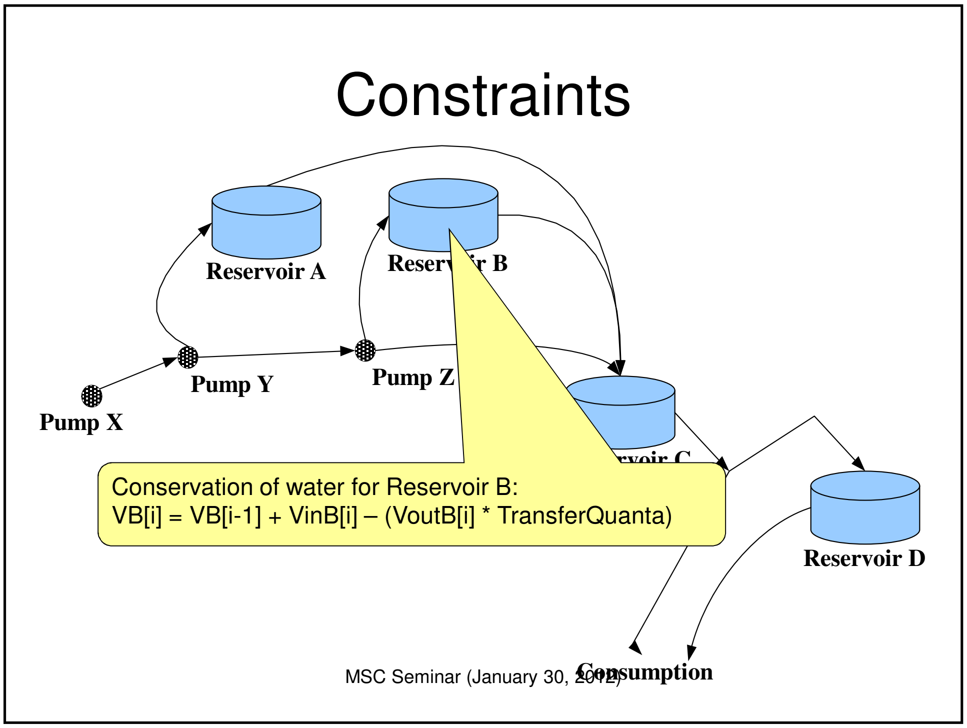
Constraints

Conservation of Water for Pump Z:
VinB[i] = PZ[i] * PumpQuanta * CapacityPZ



Constraints

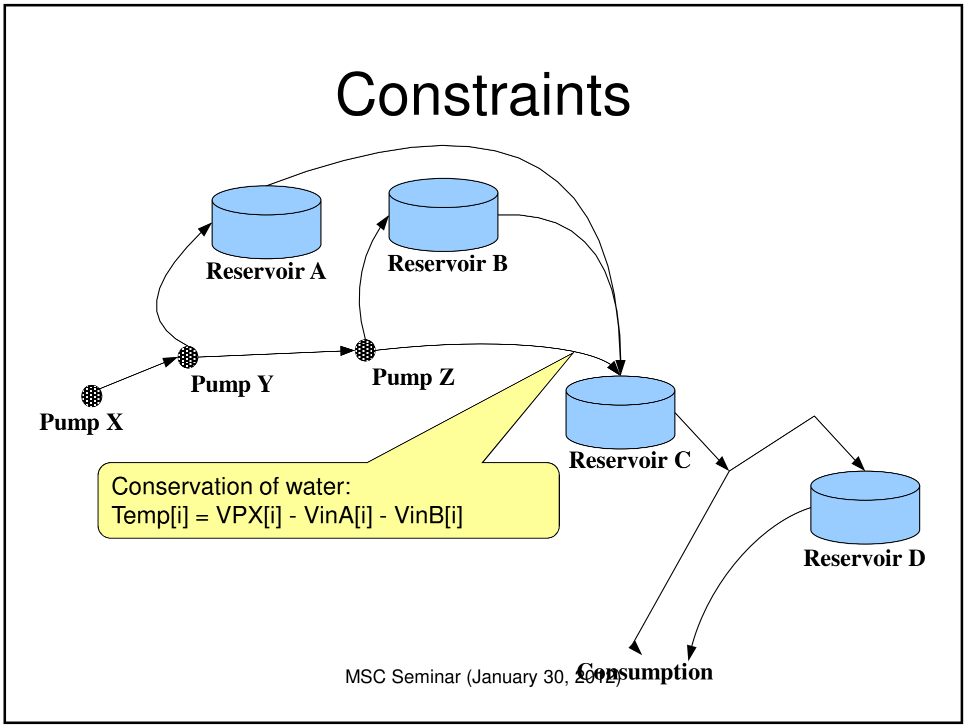
Conservation of water for Reservoir A:
VA[i] = VA[i-1] + VinA[i] – (VoutA[i] * TransferQuanta)



Constraints

Conservation of water for Reservoir B:
VB[i] = VB[i-1] + VinB[i] – (VoutB[i] * TransferQuanta)

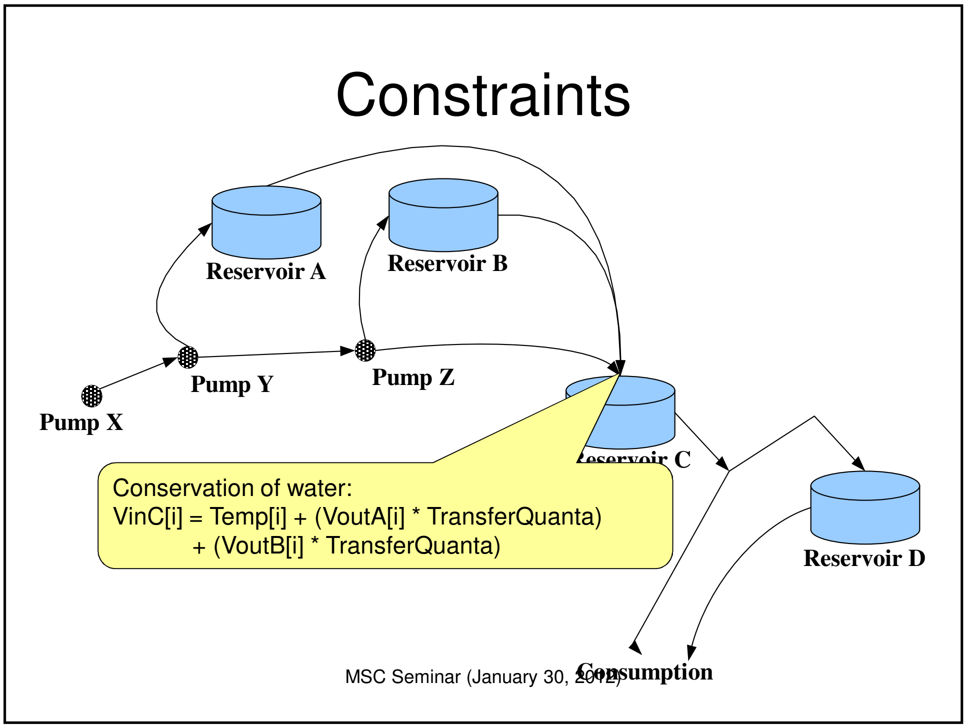
MSC Seminar (January 30, 2012)



Constraints

Conservation of water:
Temp[i] = VPX[i] - VinA[i] - VinB[i]

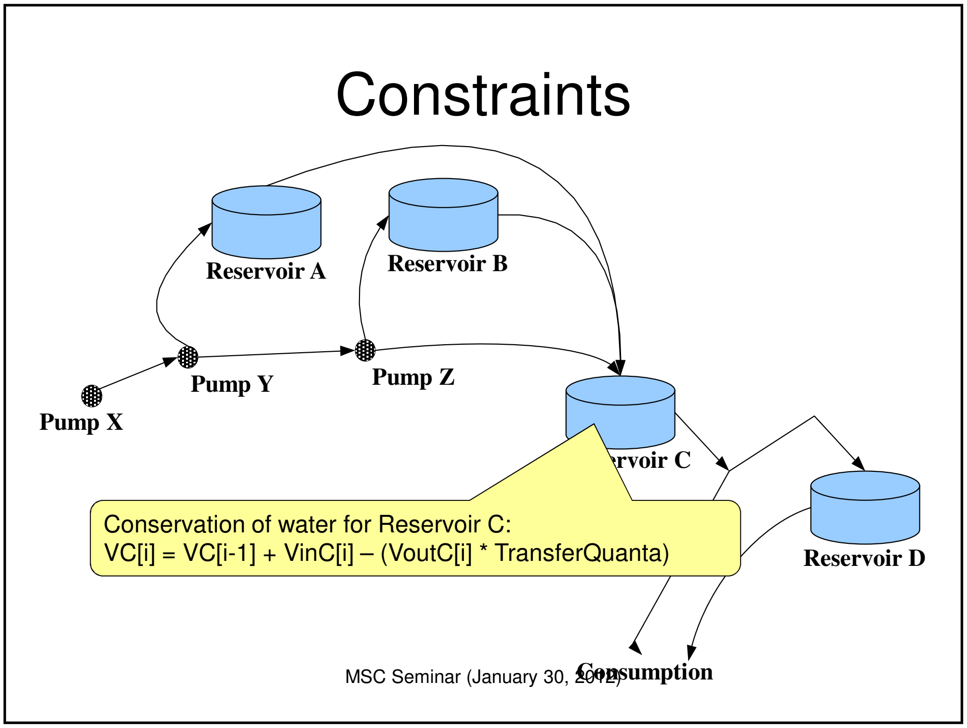
MSC Seminar (January 30, 2012)



Constraints

Conservation of water:
VinC[i] = Temp[i] + (VoutA[i] * TransferQuanta)
+ (VoutB[i] * TransferQuanta)

MSC Seminar (January 30, 2012)

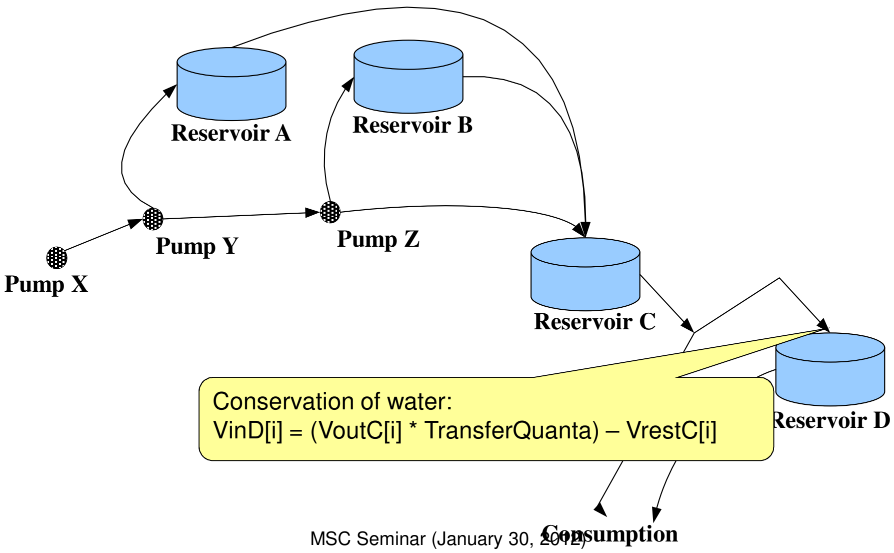


Constraints

Conservation of water for Reservoir C:
VC[i] = VC[i-1] + VinC[i] – (VoutC[i] * TransferQuanta)

MSC Seminar (January 30, 2012)

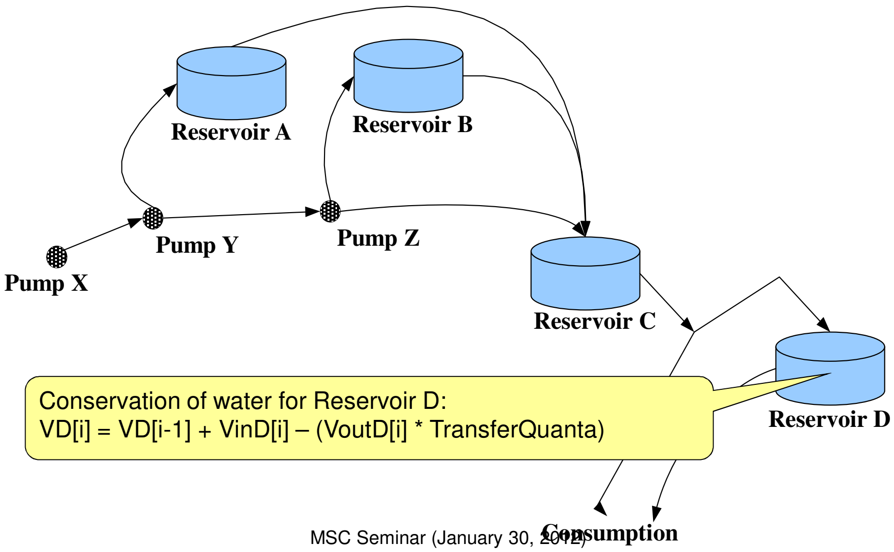
Constraints



Conservation of water:
VinD[i] = (VoutC[i] * TransferQuanta) – VrestC[i]

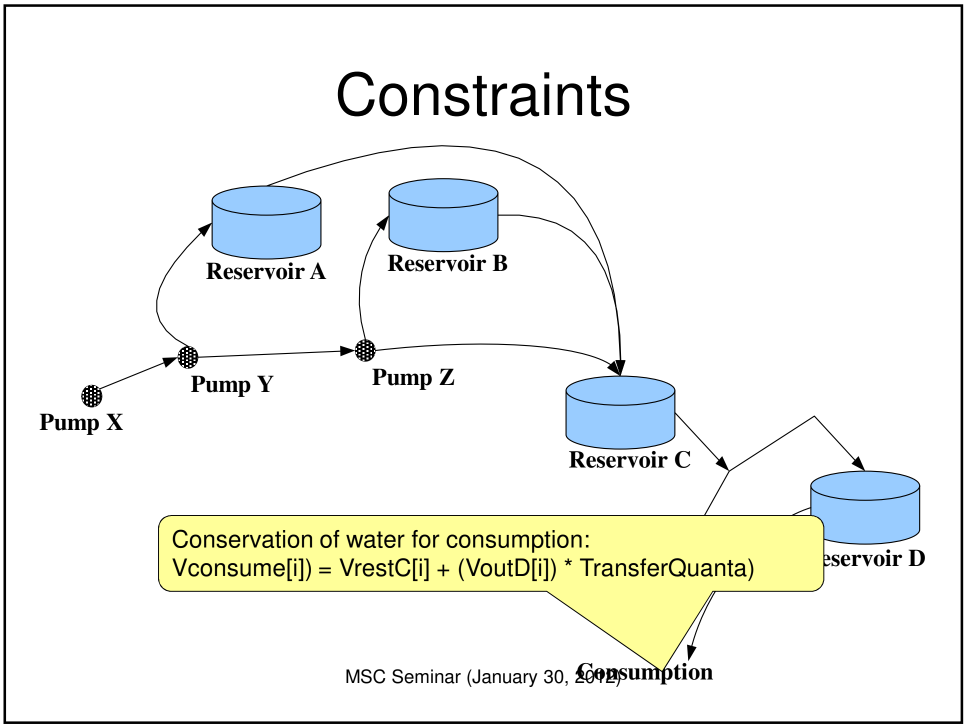
MSC Seminar (January 30, 2012)

Constraints



Conservation of water for Reservoir D:
VD[i] = VD[i-1] + VinD[i] – (VoutD[i] * TransferQuanta)

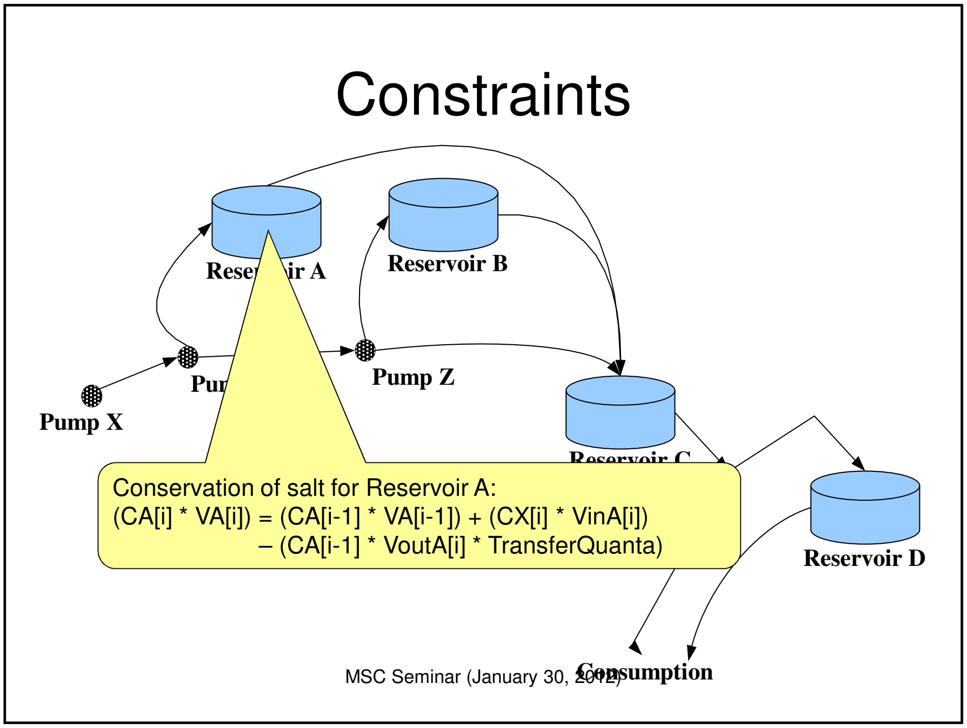
MSC Seminar (January 30, 2012)



Constraints

Conservation of water for consumption:
Vconsume[i]) = VrestC[i] + (VoutD[i]) * TransferQuanta)

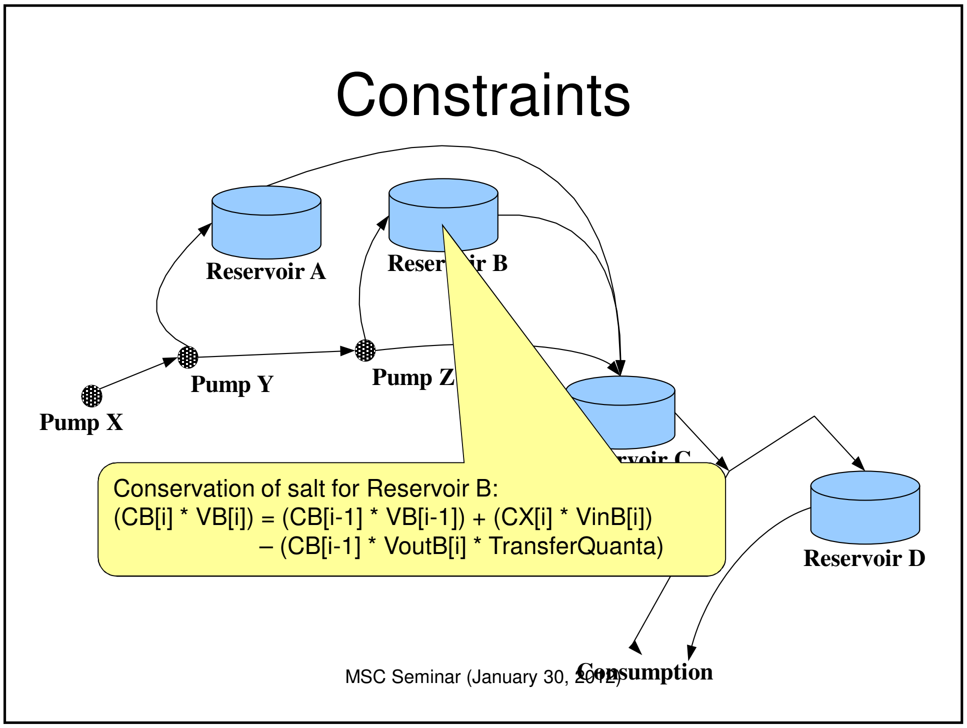
MSC Seminar (January 30, 2012)



Constraints

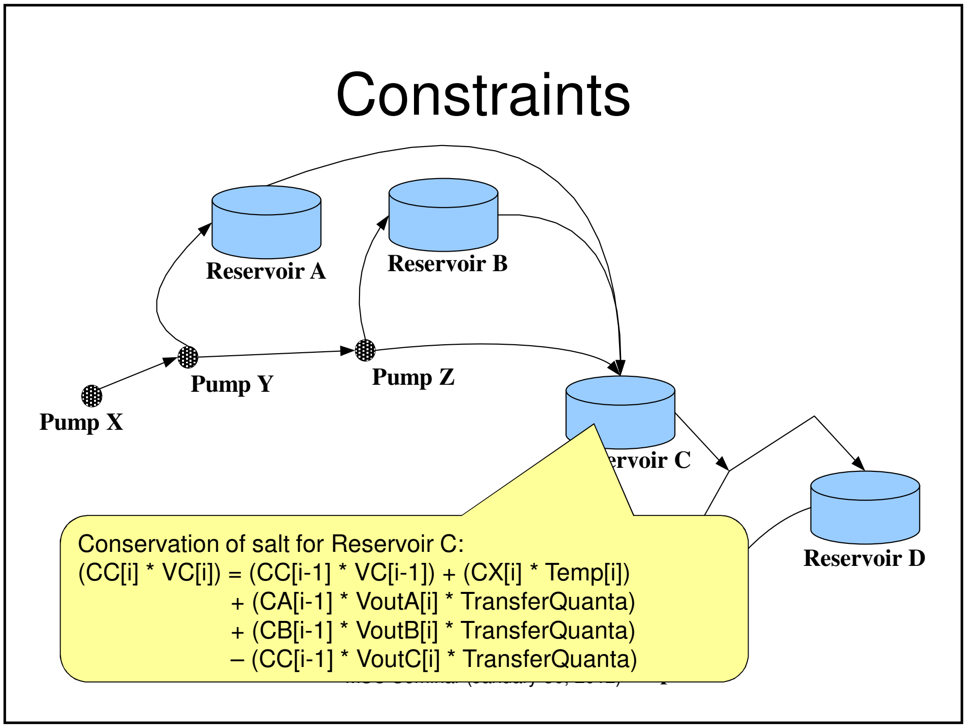
Conservation of salt for Reservoir A:
(CA[i] * VA[i]) = (CA[i-1] * VA[i-1]) + (CX[i] * VinA[i])
– (CA[i-1] * VoutA[i] * TransferQuanta)

MSC Seminar (January 30, 2012)



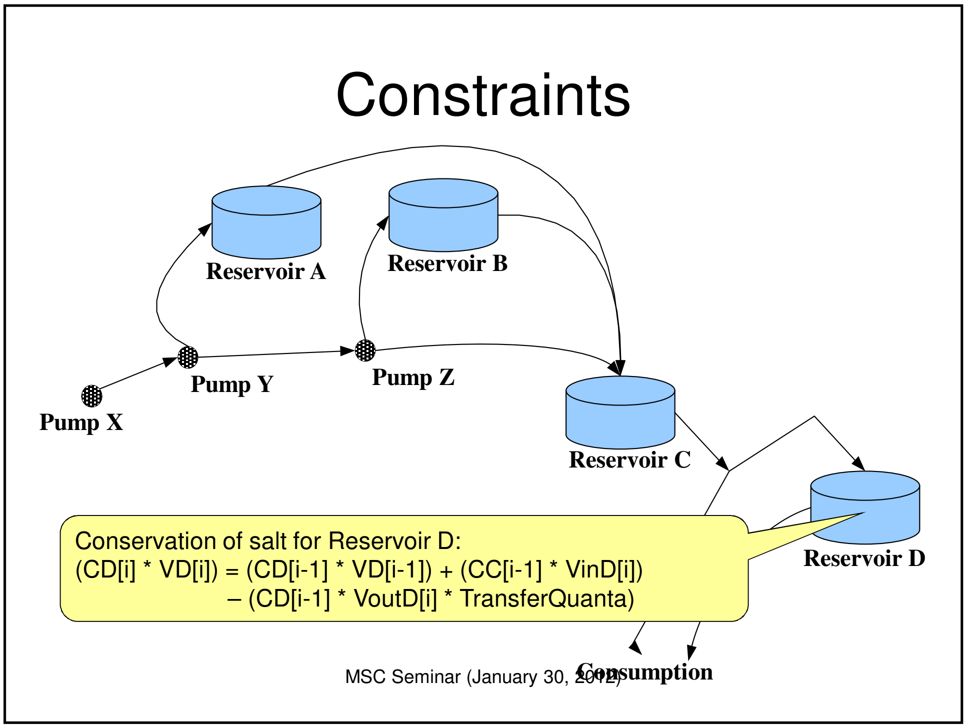
Constraints

Conservation of salt for Reservoir B:
(CB[i] * VB[i]) = (CB[i-1] * VB[i-1]) + (CX[i] * VinB[i])
– (CB[i-1] * VoutB[i] * TransferQuanta)



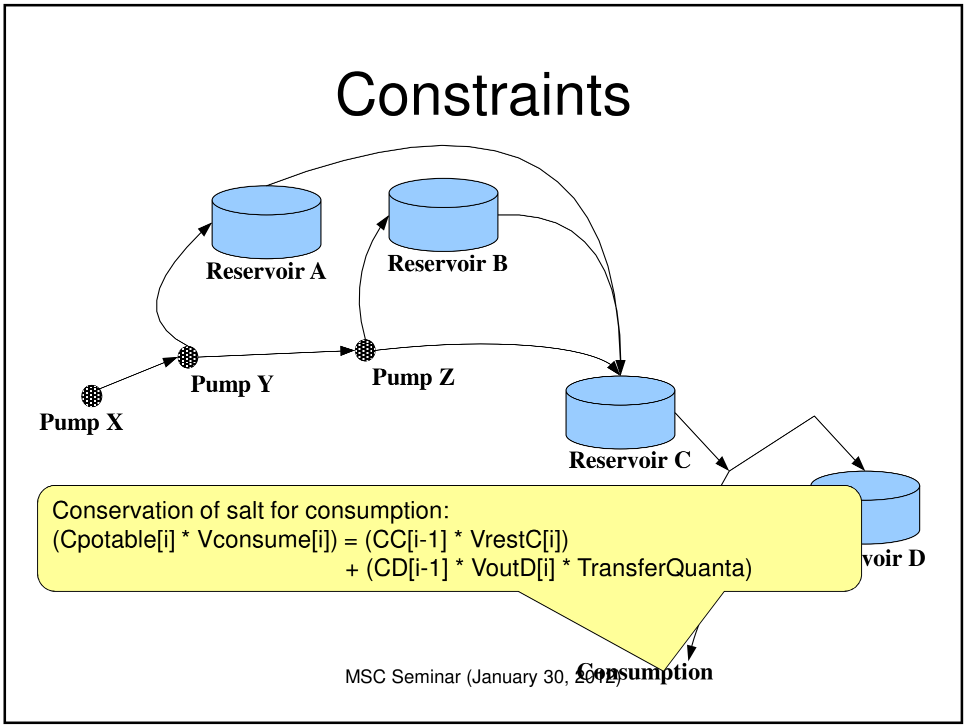
Constraints

Conservation of salt for Reservoir C:
(CC[i] * VC[i]) = (CC[i-1] * VC[i-1]) + (CX[i] * Temp[i])
+ (CA[i-1] * VoutA[i] * TransferQuanta)
+ (CB[i-1] * VoutB[i] * TransferQuanta)
– (CC[i-1] * VoutC[i] * TransferQuanta)



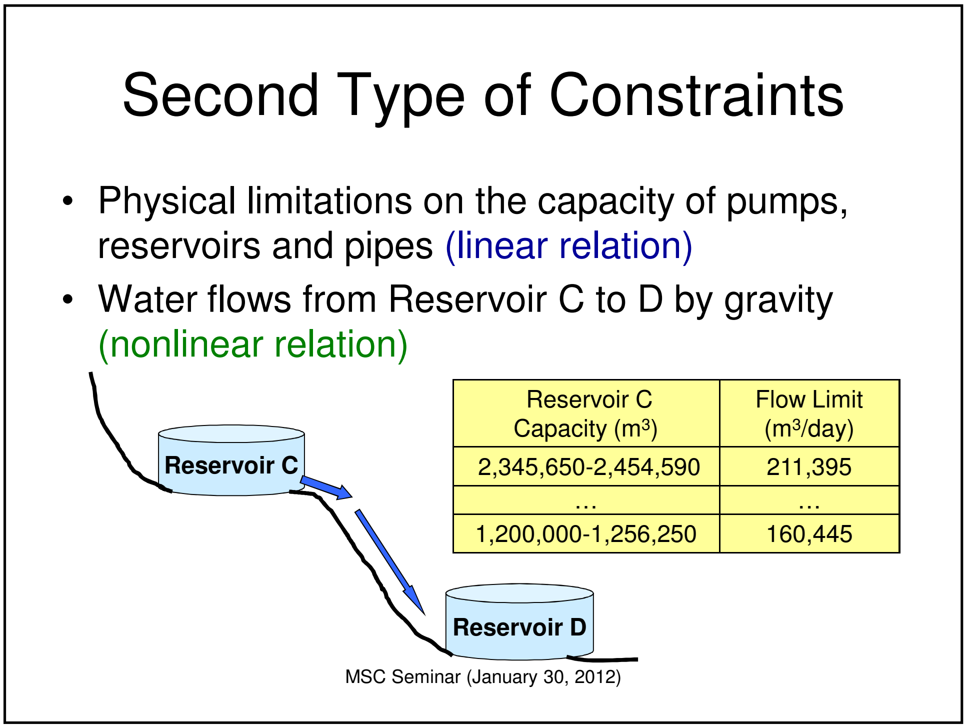
Constraints

Conservation of salt for Reservoir D:
(CD[i] * VD[i]) = (CD[i-1] * VD[i-1]) + (CC[i-1] * VinD[i])
– (CD[i-1] * VoutD[i] * TransferQuanta)



Constraints

Conservation of salt for consumption:
(Cpotable[i] * Vconsume[i]) = (CC[i-1] * VrestC[i])
+ (CD[i-1] * VoutD[i] * TransferQuanta)

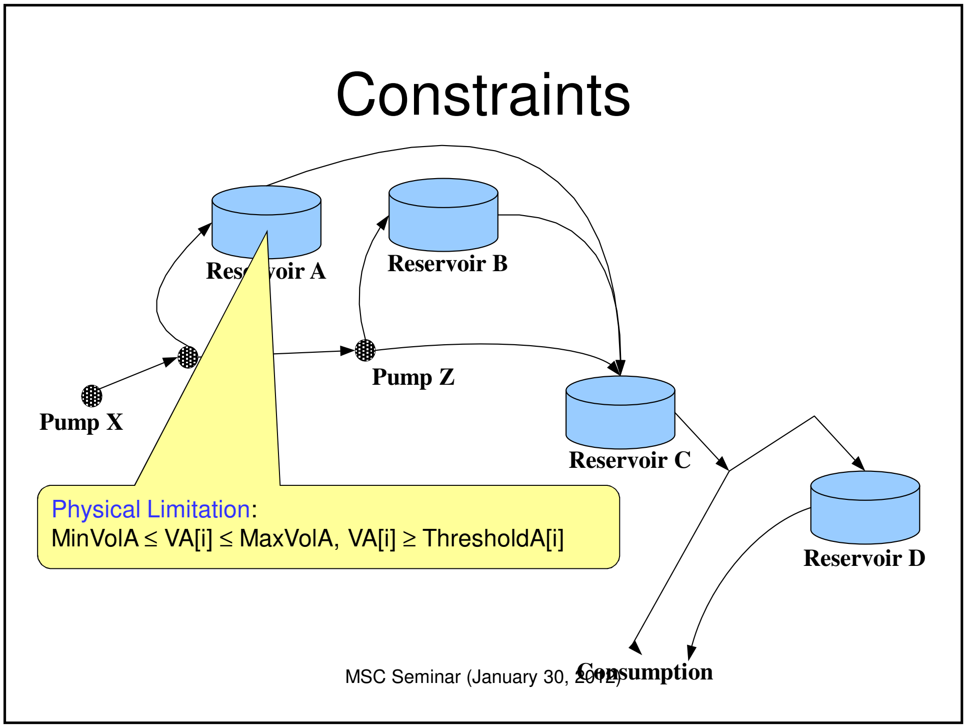


Second Type of Constraints

- Physical limitations on the capacity of pumps, reservoirs and pipes (linear relation)
- Water flows from Reservoir C to D by gravity (nonlinear relation)

Reservoir C Capacity (m³)	Flow Limit (m³/day)
2,345,650-2,454,590	211,395
…	…
1,200,000-1,256,250	160,445

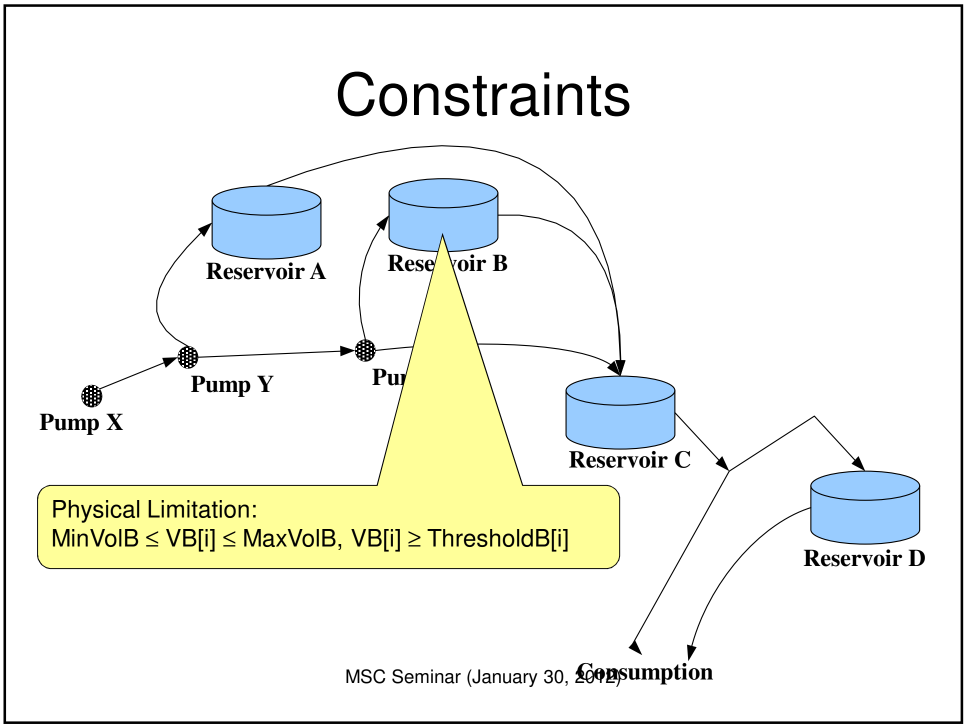
MSC Seminar (January 30, 2012)



Constraints

Physical Limitation:
MinVolA ≤ VA[i] ≤ MaxVolA, VA[i] ≥ ThresholdA[i]

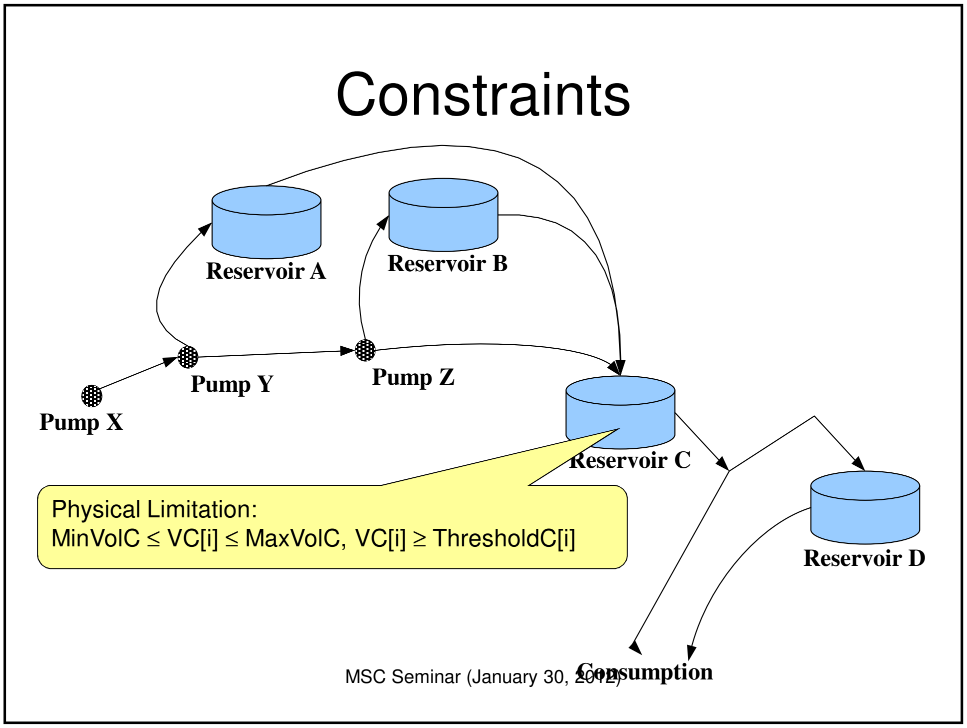
MSC Seminar (January 30, 2012)



Constraints

Physical Limitation:
MinVolB ≤ VB[i] ≤ MaxVolB, VB[i] ≥ ThresholdB[i]

MSC Seminar (January 30, 2012)

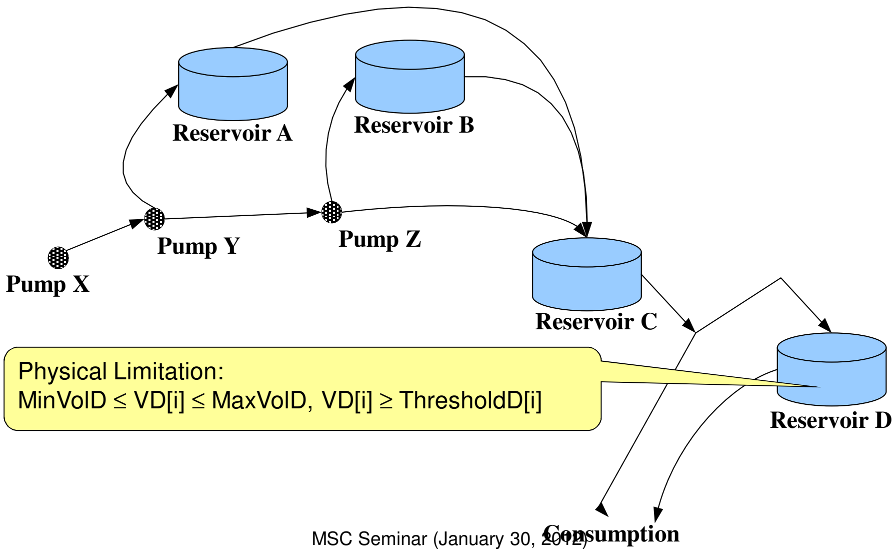


Constraints

Physical Limitation:
MinVolC ≤ VC[i] ≤ MaxVolC, VC[i] ≥ ThresholdC[i]

MSC Seminar (January 30, 2012)

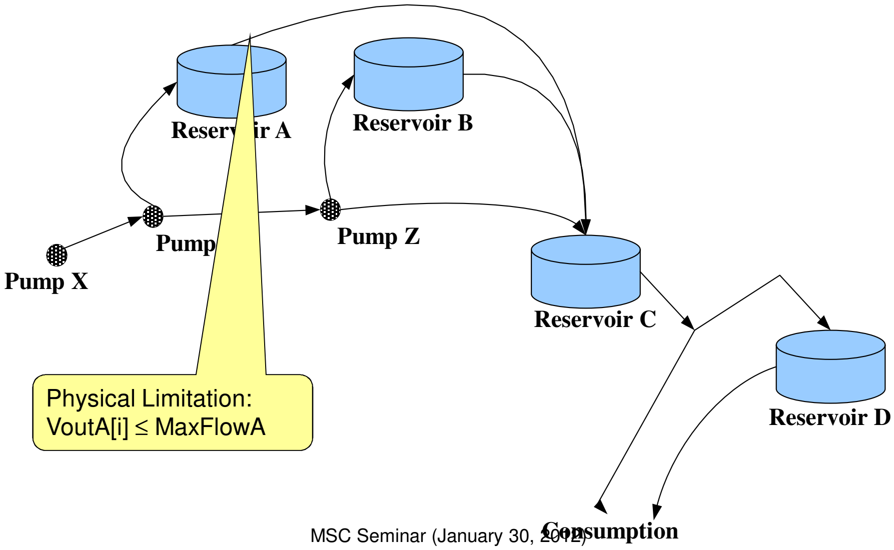
Constraints



Physical Limitation:
MinVolD ≤ VD[i] ≤ MaxVolD, VD[i] ≥ ThresholdD[i]

MSC Seminar (January 30, 2012)

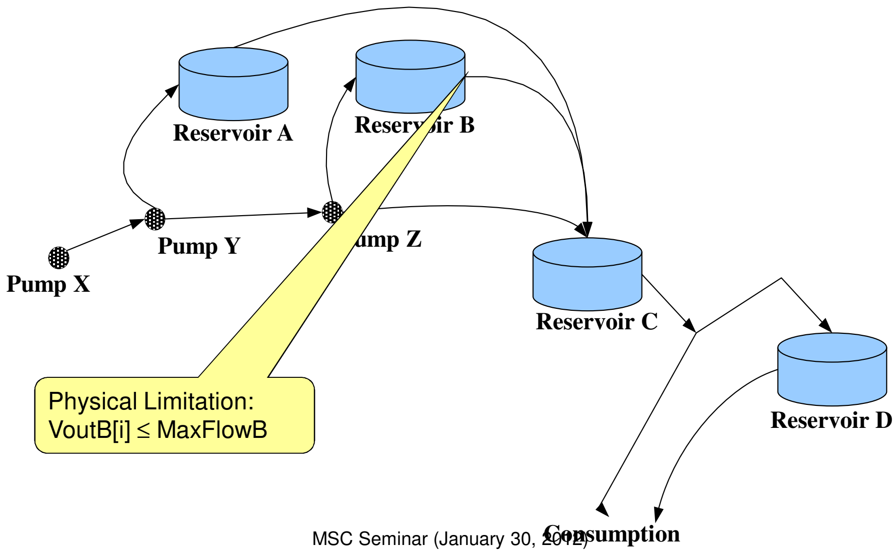
Constraints



Physical Limitation:
VoutA[i] ≤ MaxFlowA

MSC Seminar (January 30, 2012)

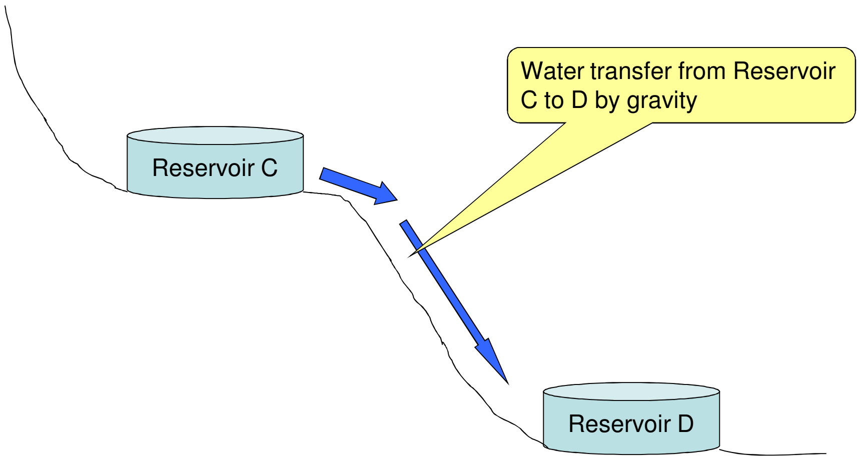
Constraints



Physical Limitation:
VoutB[i] ≤ MaxFlowB

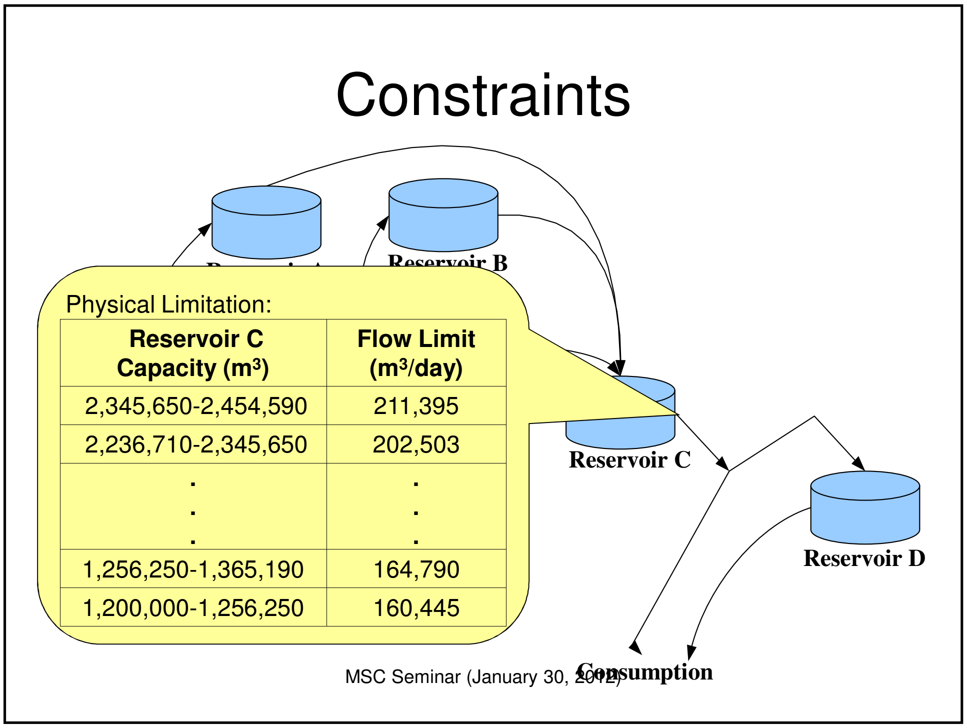
MSC Seminar (January 30, 2012)

Constraints



Water transfer from Reservoir C to D by gravity

MSC Seminar (January 30, 2012)

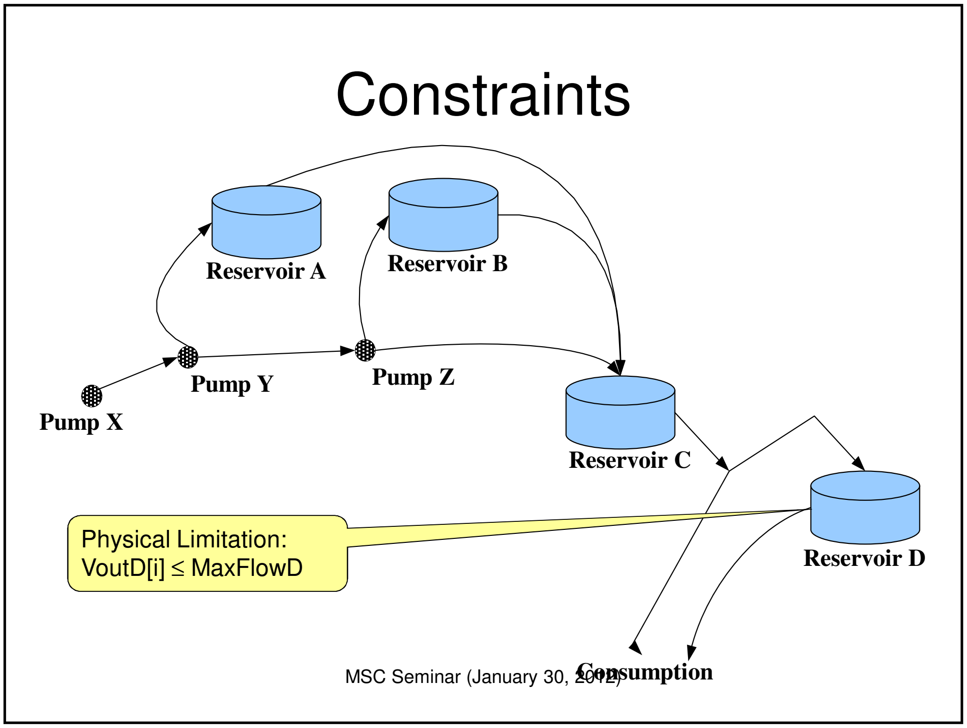


Constraints

Physical Limitation:

Reservoir C Capacity (m^3)	Flow Limit (m^3/day)
2,345,650-2,454,590	211,395
2,236,710-2,345,650	202,503
. . .	. . .
1,256,250-1,365,190	164,790
1,200,000-1,256,250	160,445

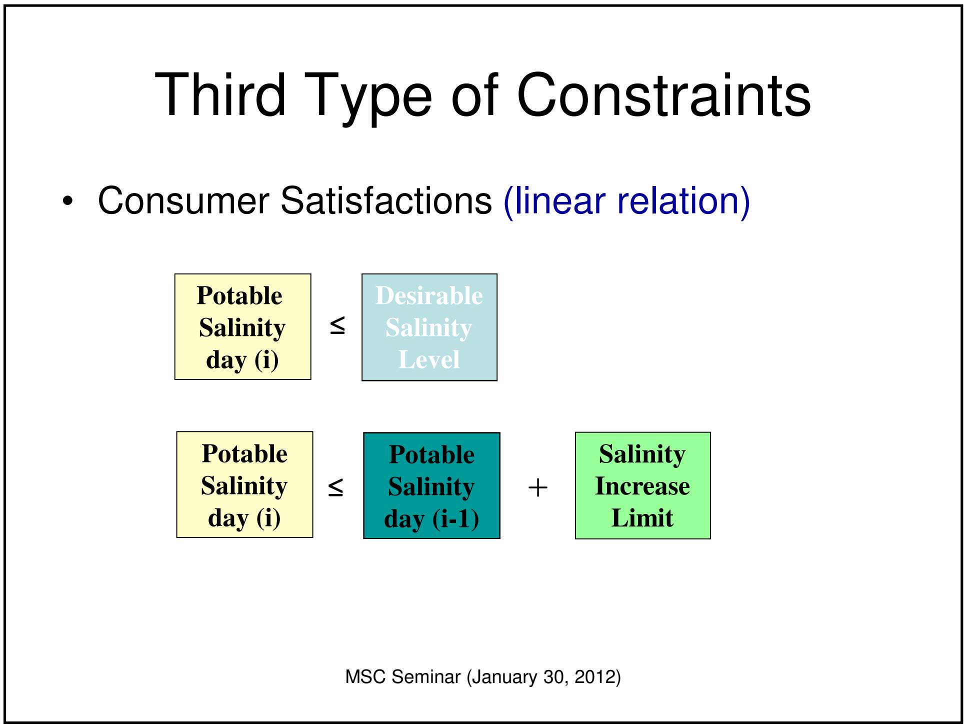
MSC Seminar (January 30, 2012)



Constraints

Physical Limitation:
VoutD[i] ≤ MaxFlowD

MSC Seminar (January 30, 2012)

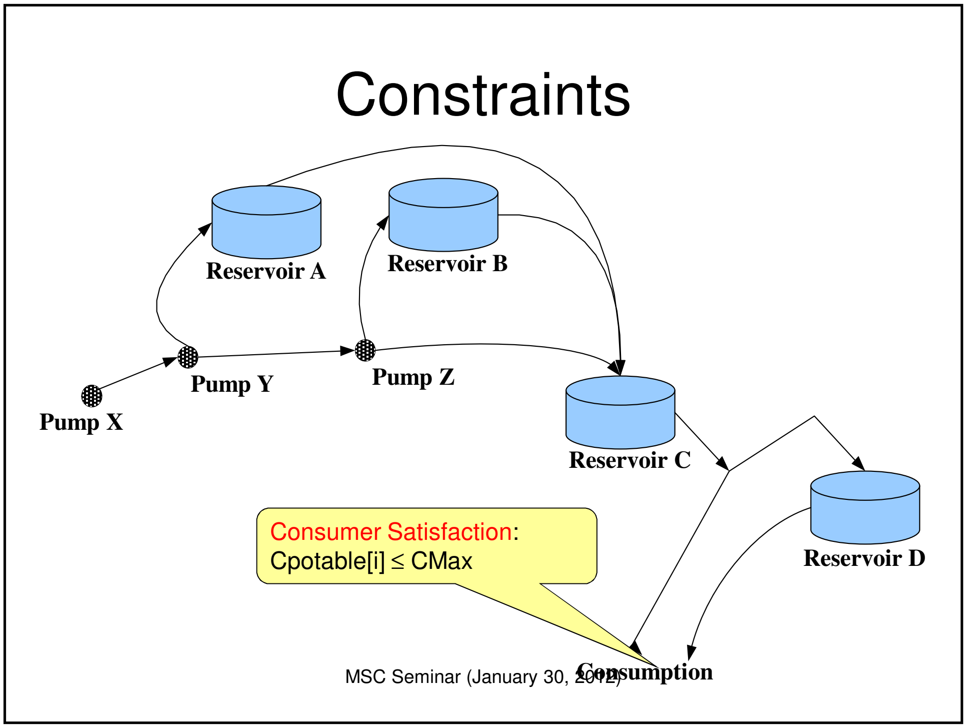


Third Type of Constraints

- Consumer Satisfactions (linear relation)

Potable Salinity day (i) ≤ Desirable Salinity Level

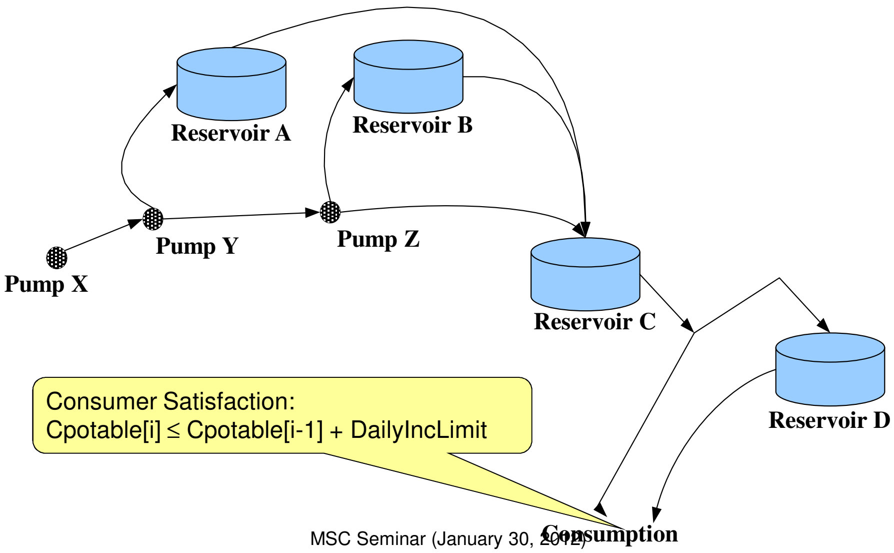
Potable Salinity day (i) ≤ Potable Salinity day (i-1) + Salinity Increase Limit



Constraints

Consumer Satisfaction:
Cpotable[i] ≤ CMax

Constraints

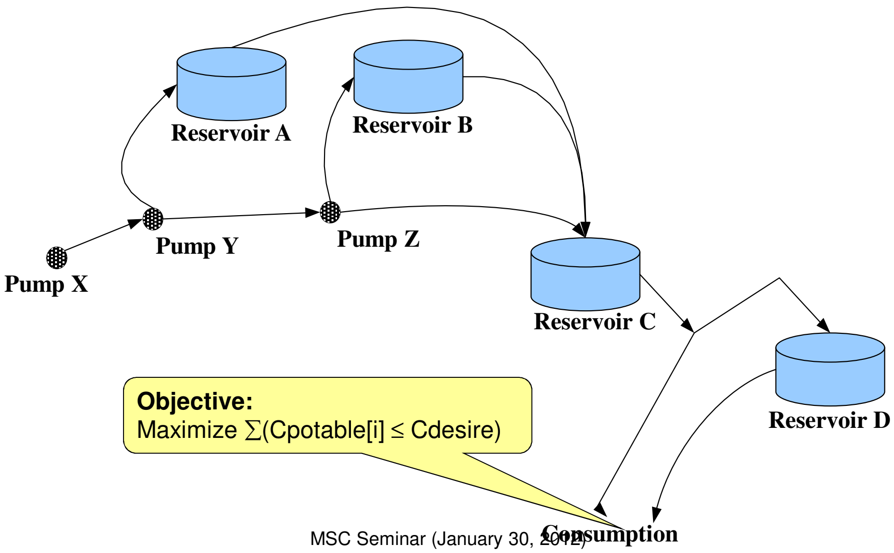


Consumer Satisfaction:
Cpotable[i] ≤ Cpotable[i-1] + DailyIncLimit

Objective

To maximize the number of days in which the potable salinity level is below the desirable level

Objective



Objective:
Maximize ∑(Cpotable[i] ≤ Cdesire)

Implementation Platform

- State-of-the-art constraint programming system
  - ILOG Solver 6.0 (C++ library)
  - GNU Compiler Collection (GCC 3.2)
  - PC running GNU/Linux (Linux kernel 2.4)

Difficulties

- The problem size is large
  - Salinity periods of 180 days
  - About 4,500 variables and 9,000 constraints
  - About $(3{,}612{,}000)^{180}$ possible search states

- The problem involves both linear and non-linear constraints

Difficulties

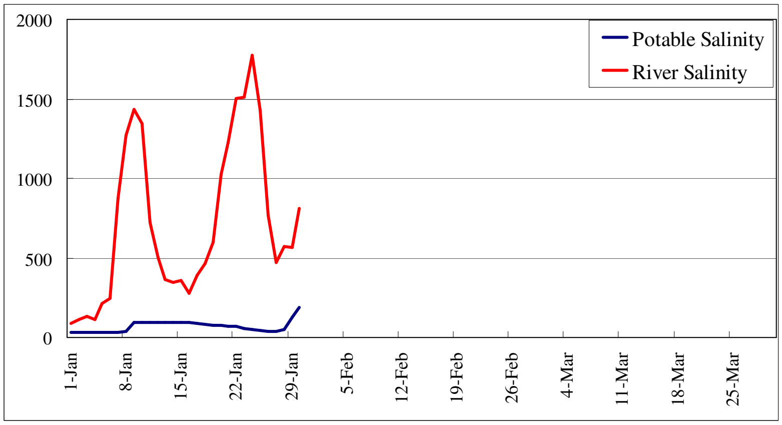
- The different components of the raw water system interact with each other temporally and geographically (topologically)

- Although efficient commercial optimization tools are available, out-of-the-box execution strategies fail to handle even small test cases

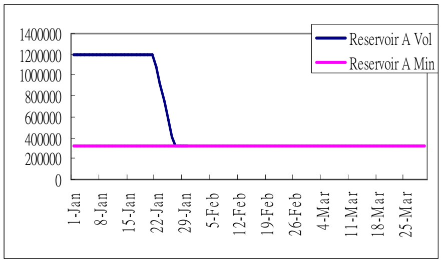
Scenario A

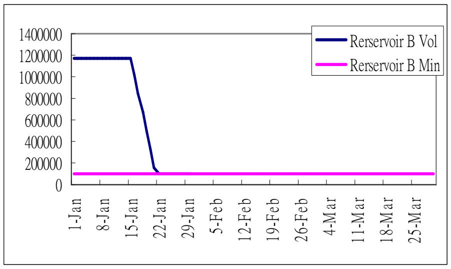
Default Search Heuristic

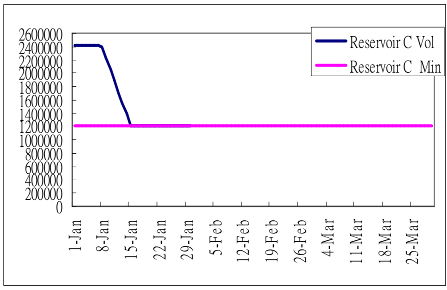
Result: Can handle only a 30-day period!!



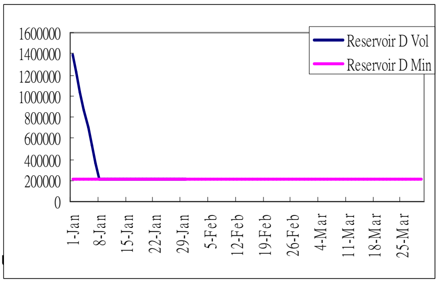
All Reservoirs Drained!!!!!!!!!!!

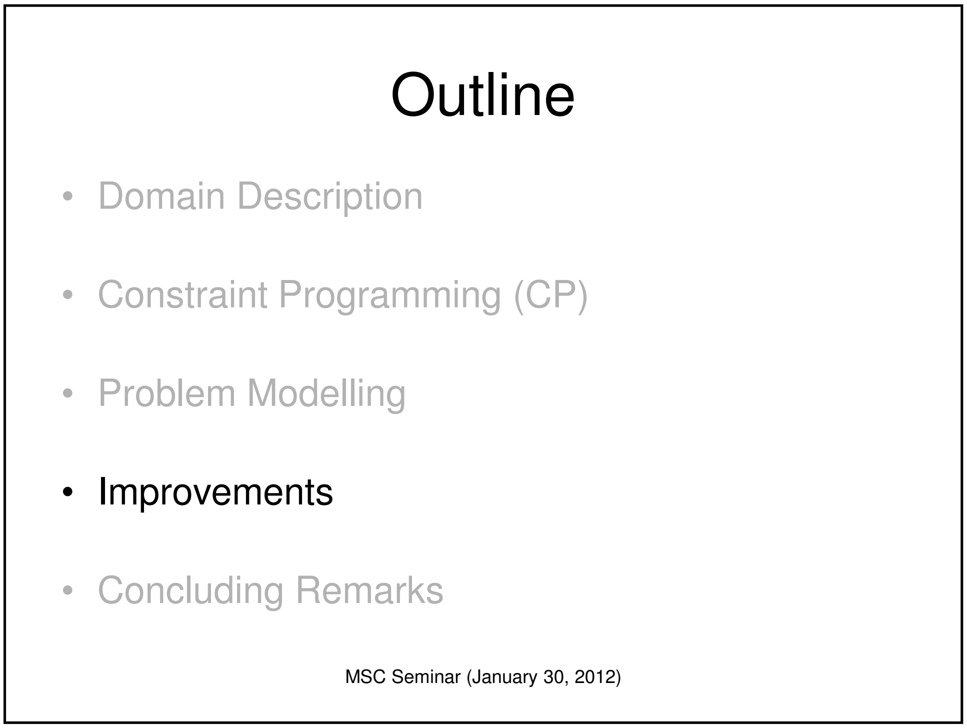






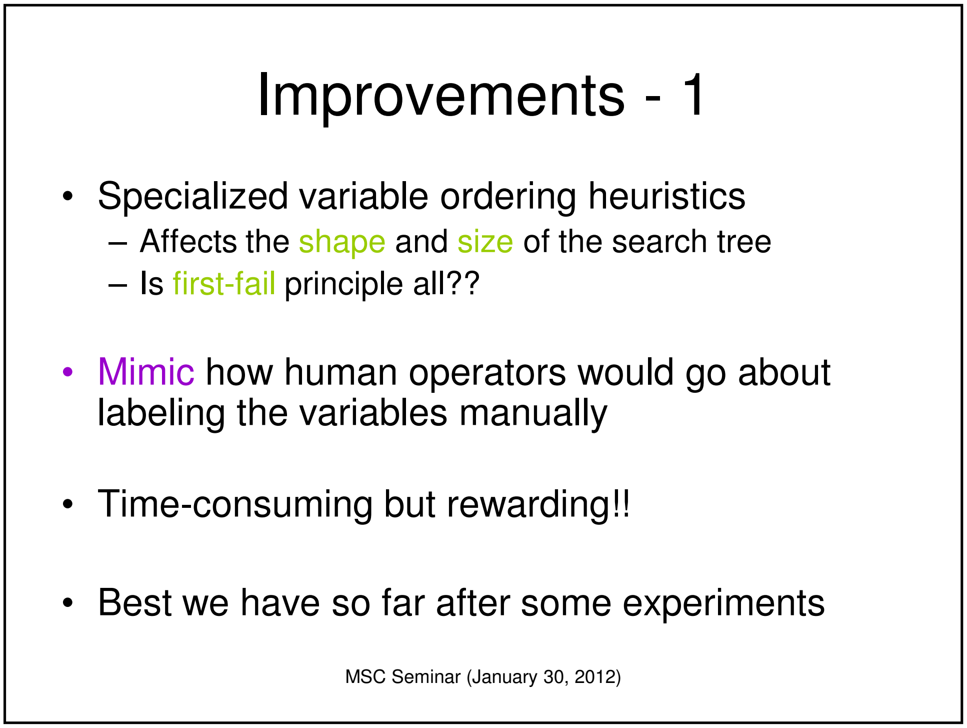
an





Outline

- Domain Description
- Constraint Programming (CP)
- Problem Modelling
- Improvements
- Concluding Remarks



Improvements - 1

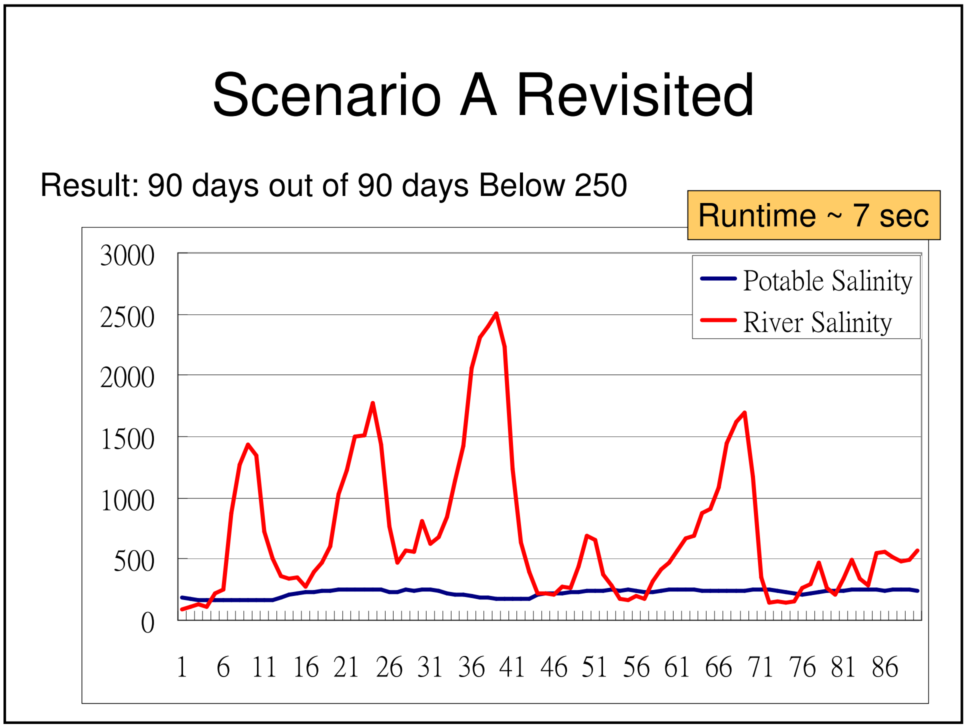
- Specialized variable ordering heuristics
  - Affects the shape and size of the search tree
  - Is first-fail principle all??
- Mimic how human operators would go about labeling the variables manually
- Time-consuming but rewarding!!
- Best we have so far after some experiments

Improvements - 2

- Specialized value ordering heuristics
  - Affects the ordering of the branches
  - Move (good) solution branches as far to the left as possible (depth-first search from left to right)
- Control how much to pump from the river
- Control how much to transfer between reservoirs (which reservoir’s water to use first)

Improvements - 3

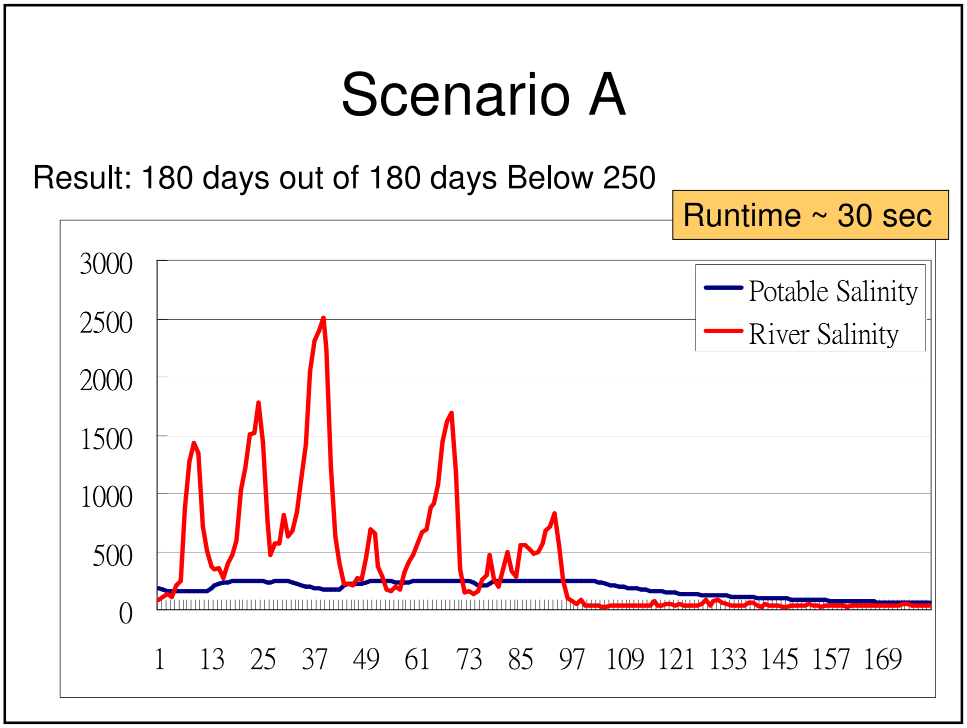
- Changing constraint representations by adding auxiliary variables and rewriting the constraints
- Induce extra constraint propagation and thus also pruning: the more the better!!



Scenario A Revisited

Result: 90 days out of 90 days Below 250

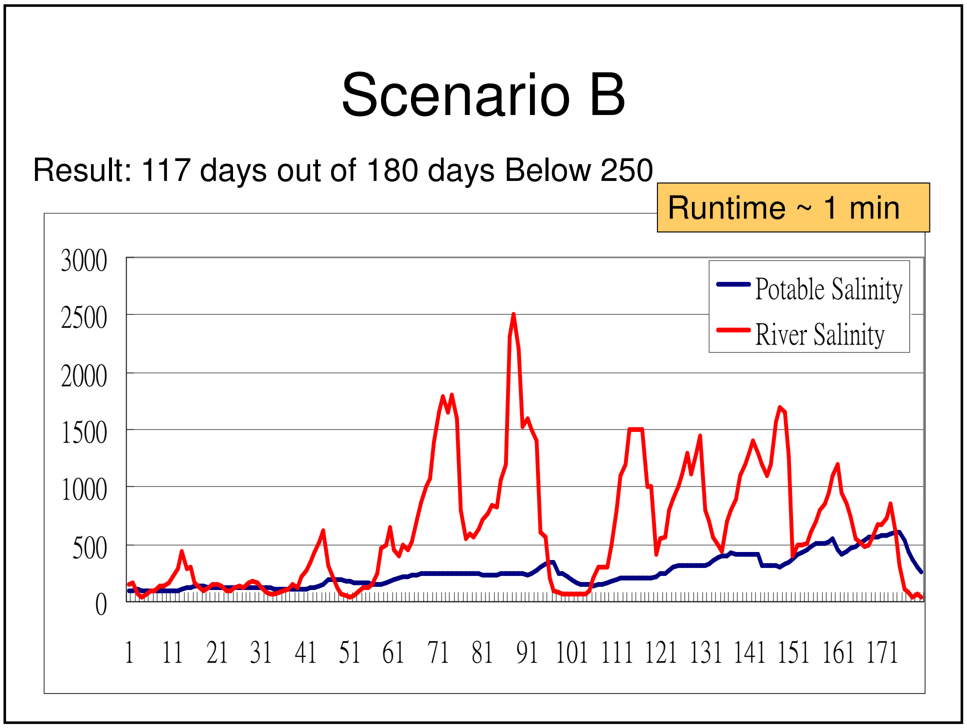
Runtime ~ 7 sec



Scenario A

Result: 180 days out of 180 days Below 250

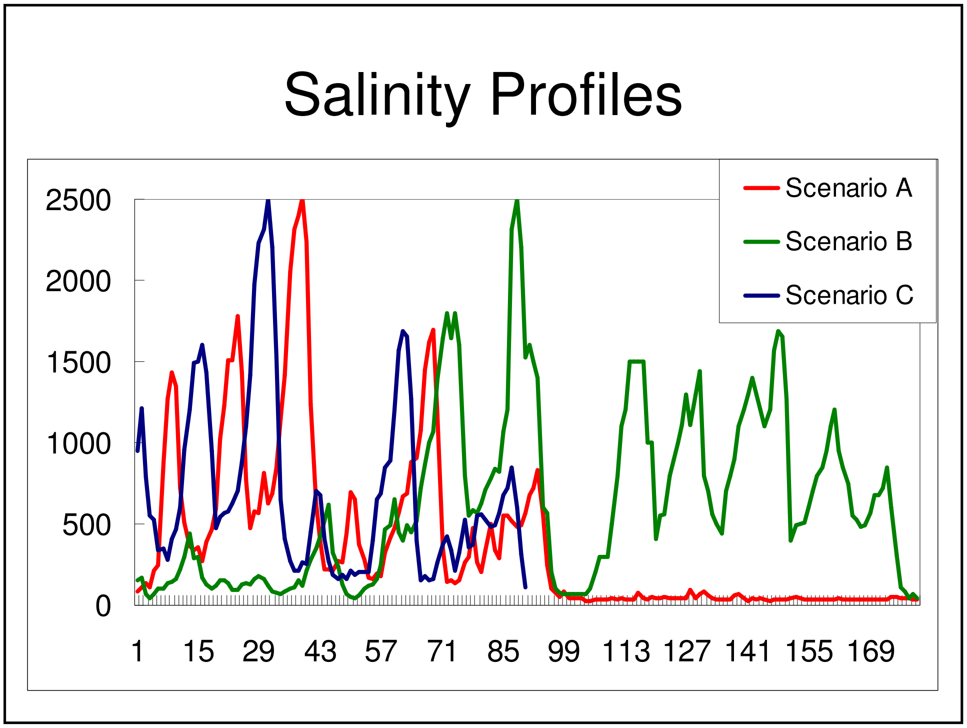
Runtime ~ 30 sec



Scenario B

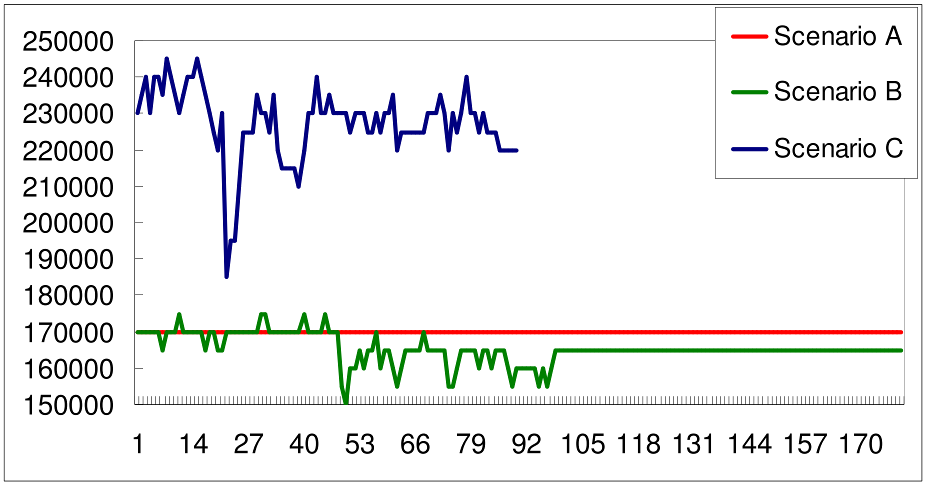
Result: 117 days out of 180 days Below 250

Runtime ~ 1 min



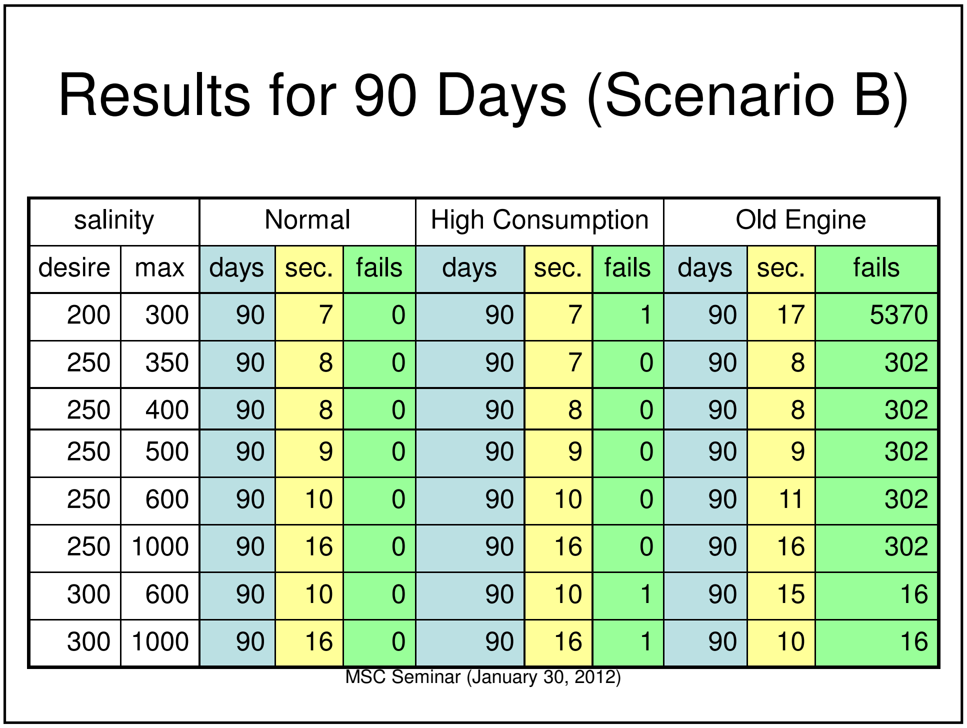
Salinity Profiles

Consumption Profiles



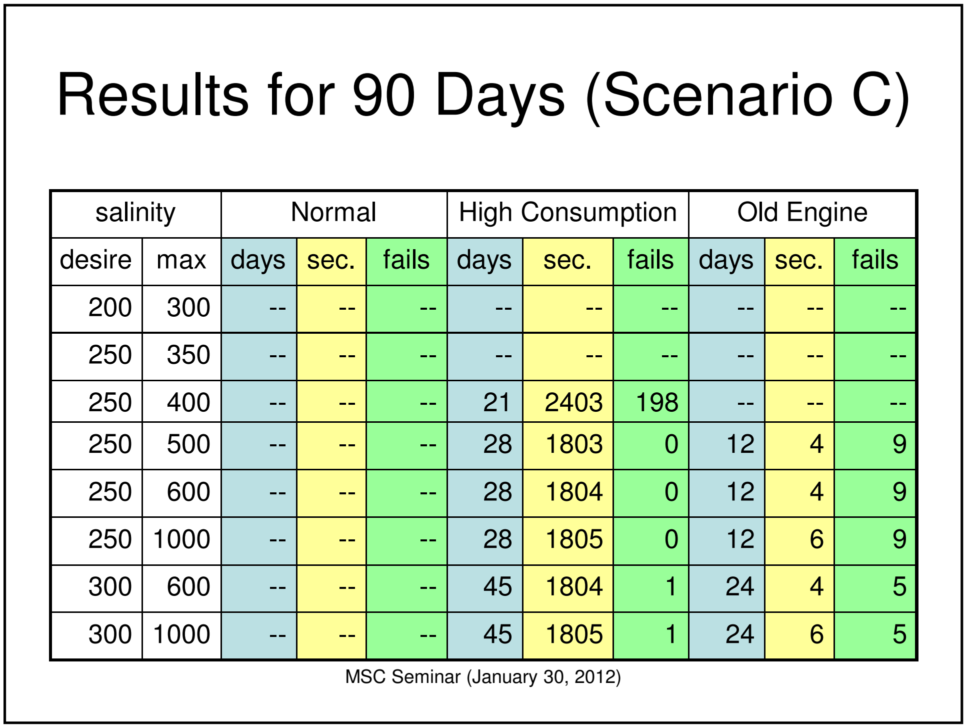
Results for 90 Days (Scenario A)

salinity		Normal			High Consumption			Old Engine		
desire	max	days	sec.	fails	days	sec.	fails	days	sec.	fails
200	300	74	5	0	51	425	1	36	300	115497
250	350	90	7	0	87	366	0	90	78	39758
250	400	90	8	0	87	367	0	90	78	39758
250	500	90	10	0	87	369	0	90	80	39758
250	600	90	12	0	87	370	0	90	81	39758
250	1000	90	19	0	87	377	0	90	87	39758
300	600	90	12	0	90	11	1	90	15	2391
300	1000	90	19	0	90	17	1	90	15	2391



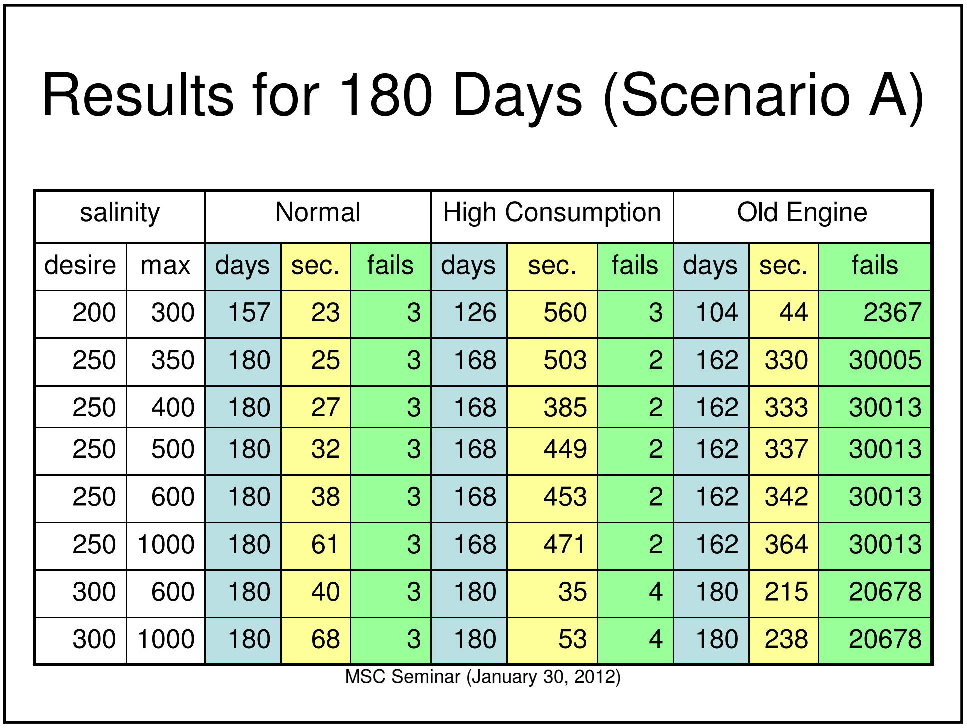
Results for 90 Days (Scenario B)

salinity		Normal			High Consumption			Old Engine		
desire	max	days	sec.	fails	days	sec.	fails	days	sec.	fails
200	300	90	7	0	90	7	1	90	17	5370
250	350	90	8	0	90	7	0	90	8	302
250	400	90	8	0	90	8	0	90	8	302
250	500	90	9	0	90	9	0	90	9	302
250	600	90	10	0	90	10	0	90	11	302
250	1000	90	16	0	90	16	0	90	16	302
300	600	90	10	0	90	10	1	90	15	16
300	1000	90	16	0	90	16	1	90	10	16



Results for 90 Days (Scenario C)

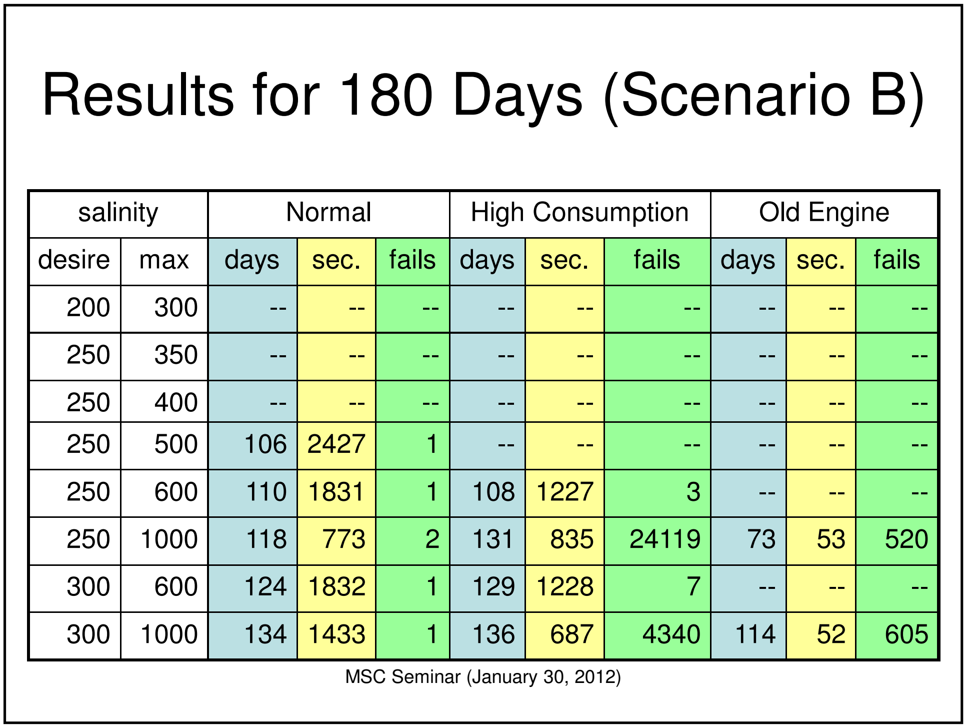
salinity		Normal			High Consumption			Old Engine		
desire	max	days	sec.	fails	days	sec.	fails	days	sec.	fails
200	300	--	--	--	--	--	--	--	--	--
250	350	--	--	--	--	--	--	--	--	--
250	400	--	--	--	21	2403	198	--	--	--
250	500	--	--	--	28	1803	0	12	4	9
250	600	--	--	--	28	1804	0	12	4	9
250	1000	--	--	--	28	1805	0	12	6	9
300	600	--	--	--	45	1804	1	24	4	5
300	1000	--	--	--	45	1805	1	24	6	5



Results for 180 Days (Scenario A)

salinity		Normal			High Consumption			Old Engine		
desire	max	days	sec.	fails	days	sec.	fails	days	sec.	fails
200	300	157	23	3	126	560	3	104	44	2367
250	350	180	25	3	168	503	2	162	330	30005
250	400	180	27	3	168	385	2	162	333	30013
250	500	180	32	3	168	449	2	162	337	30013
250	600	180	38	3	168	453	2	162	342	30013
250	1000	180	61	3	168	471	2	162	364	30013
300	600	180	40	3	180	35	4	180	215	20678
300	1000	180	68	3	180	53	4	180	238	20678

MSC Seminar (January 30, 2012)



Results for 180 Days (Scenario B)

salinity		Normal			High Consumption			Old Engine		
desire	max	days	sec.	fails	days	sec.	fails	days	sec.	fails
200	300	--	--	--	--	--	--	--	--	--
250	350	--	--	--	--	--	--	--	--	--
250	400	--	--	--	--	--	--	--	--	--
250	500	106	2427	1	--	--	--	--	--	--
250	600	110	1831	1	108	1227	3	--	--	--
250	1000	118	773	2	131	835	24119	73	53	520
300	600	124	1832	1	129	1228	7	--	--	--
300	1000	134	1433	1	136	687	4340	114	52	605

MSC Seminar (January 30, 2012)

Advantages

- Formulate a model relatively close to the original problem, making the model easy to verify and maintain
- Design domain specific search heuristic to reduce the time of searching for solutions
- Find better quality solutions in a much shorter time than human operators
- Can be used by novice operators

Outline

- Domain Description
- Constraint Programming (CP)
- Problem Modelling
- Improvements
- Concluding Remarks

Genetic Algorithms

- UNU-IIST tried EVOLVER, which is a GA-based optimization engine for MS Excel
- Less efficient and lower quality solution
- Semi-automatic: requires expert human guidance during search
- Unstable and unpredictable with regard to convergence

Mathematical Programming

- Collaboration with our OR experts
- Advantage: the domain of the problem is continuous in nature (i.e. real numbers)
- Major difficulties: non-linear constraints
  - Law of conservation of salts
  - Table constraints

Concluding Remarks

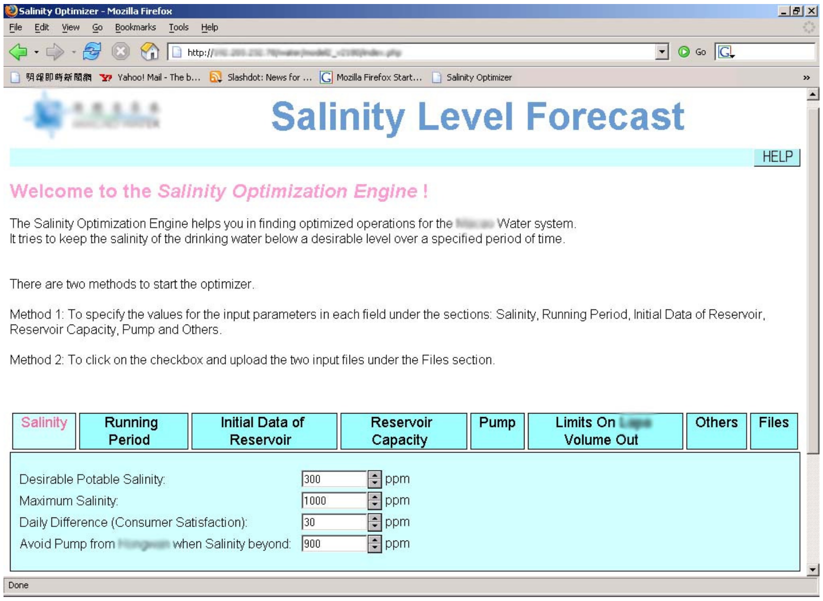
- Introduced to the problem in October, 2004
- First prototype early December, 2004
- Version 3.x now
- 1 PhD student and 2 final year undergraduates (+ Me)
- User acceptance passed

Concluding Remarks

- We apply CP to solve the optimization of logistical operations during the salinity periods (to the best of our knowledge, the first application of CP to water resource management)
- The system is expected to benefit some 450,000 residents during the upcoming salinity periods

Still, CP cannot combat with nature!

The Web Interface



Salinity Level Forecast

Welcome to the *Salinity Optimization Engine* !

The Salinity Optimization Engine helps you in finding optimized operations for the [redacted] Water system.
It tries to keep the salinity of the drinking water below a desirable level over a specified period of time.

There are two methods to start the optimizer.

Method 1: To specify the values for the input parameters in each field under the sections: Salinity, Running Period, Initial Data of Reservoir, Reservoir Capacity, Pump and Others.

Method 2: To click on the checkbox and upload the two input files under the Files section.

Salinity	Running Period	Initial Data of Reservoir	Reservoir Capacity	Pump	Limits On [redacted] Volume Out	Others	Files

Desirable Potable Salinity:	300	ppm
Maximum Salinity:	1000	ppm
Daily Difference (Consumer Satisfaction):	30	ppm
Avoid Pump from [redacted] when Salinity beyond:	900	ppm

Collaboration with UNU-IIST

- Core optimization engine developed by CUHK

- Web interface developed by UNU-IIST

United Nations
University

UNU-IIST
International Institute for
Software Technology