

System Reliability & Multi-Objective Optimization Design: Multiple Objective Optimization



Prof. David W. Coit & Assoc. Prof. Naruemon Wattanapongsakorn

King Mongkut's University Technology of Thonburi (KMUTT)

Term 1/2016

Developed new MOEA for design allocation problems:

MOEA-DAP characteristics:

- MOEA for solving reliability design allocation problems.
- It differs from other from other MOEAs in:
 - ✓ **Fitness assignment:** two types of fitness metrics are separately evaluated and later aggregated to obtain an overall aggregated fitness value
 - ✓ **Crossover step:** several offspring are created through multi-parent recombination – the EA uses a problem dependent structure

The two main goals in the development of MOEAs

1. **Proximity** : close to the Pareto front
2. **Diversity** : well distributed Pareto set

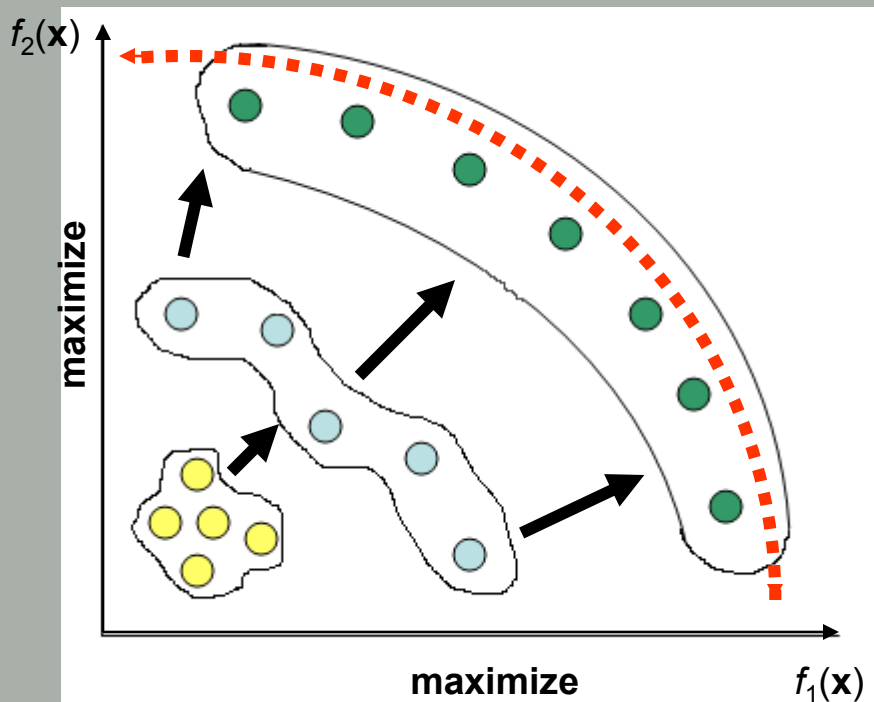


Figure 1. Achieving proximity in MOEAs

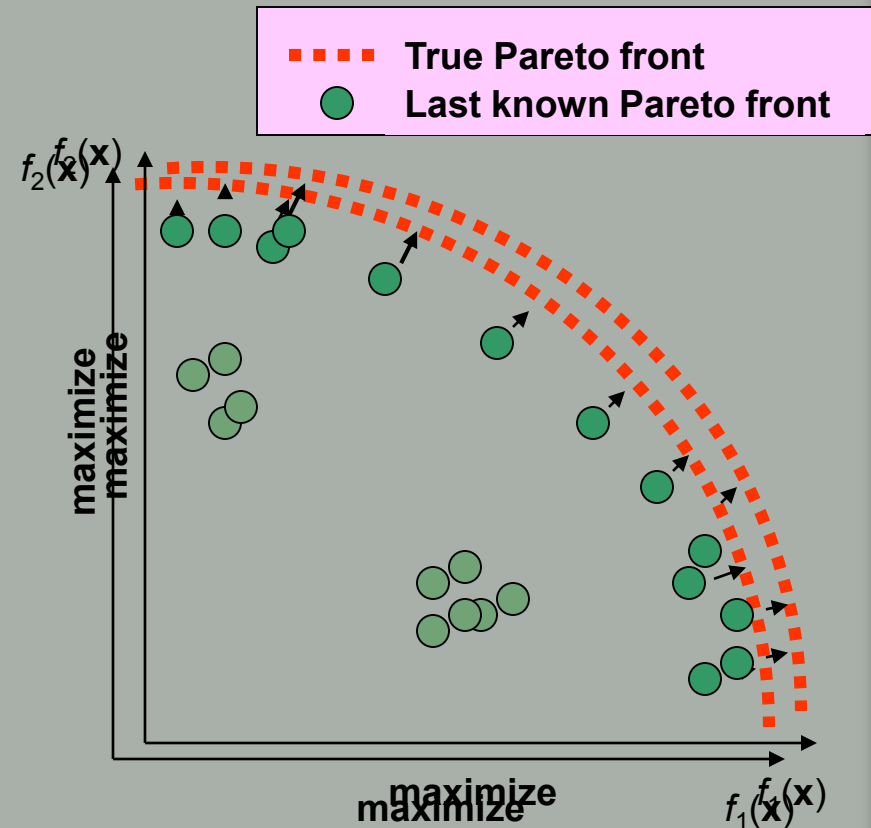


Figure 2. Diversity is not achieved

MOEA-DAP: design characteristics

1. Generation of the initial population

- Integer chromosomal representation

2. Pareto dominance criterion

3. Fitness assignment

Two different fitness functions:

- Fitness metric 1: Distance-based
- Fitness metric 2: Dominance count-based

4. Selection: Rank selection

5. Crossover: A customized “subsystem rotation crossover”

6. Single-point mutation

7. Elitist Reinsertion



Step 1. Generation of the initial population

Chromosome encoding

It is simply the form of the basic chromosome

- Binary
- Integer
- Value

1	1	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---

1	3	4	9	2	7	5	8	6
---	---	---	---	---	---	---	---	---



Chromosome 1:	1.23	2.15	4.92	2.20	3.41	5.68
---------------	------	------	------	------	------	------

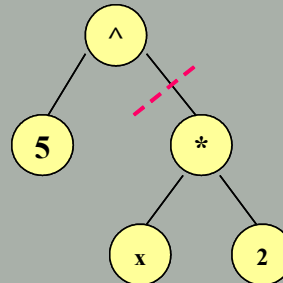


Chromosome 2:	A	B	E	D	F	C
---------------	---	---	---	---	---	---



Chromosome3:	N	W	S	N	E	N
--------------	---	---	---	---	---	---

- Tree encoding



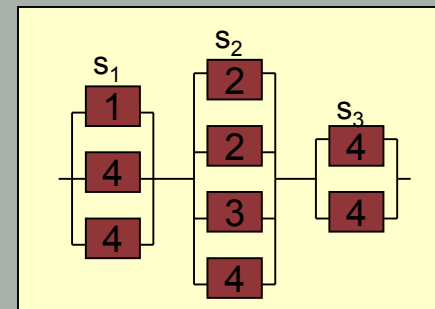
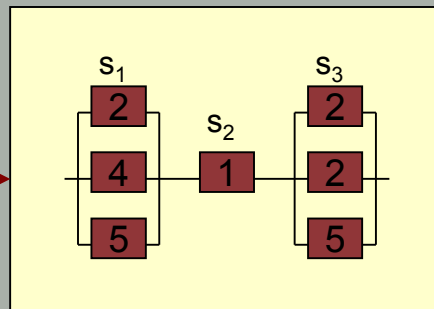
The chromosome implementation is driven by the type of problem to be solved

Step 1. Generation of the initial population (cont'd...)

Step 1.1 Evaluate objective function values

- 1 chromosome = 1 system configuration

Reliability	Cost	Weight	Subsystem 1					Subsystem 2				Subsystem 3				
0.960040	37	49	0	1	0	1	1	1	0	0	0	0	2	0	0	1
0.999974	101	98	1	2	0	3	0	3	0	1	1	1	3	0	1	0
0.910641	31	52	1	0	0	2	0	0	2	1	1	0	0	0	2	0
0.999188	100	98	3	0	2	1	1	1	0	3	0	2	2	1	0	1



- Length of the string

L = length of string

$m(i)$ = number of available components for subsystem i

$$L = m(1) + m(2) + \dots + m(i) = \sum_i m(i)$$

Step 1. Generation of the initial population (cont'd...)

- Integer chromosomal representation
- Generate a random initial population, ***Npop***, of chromosomes

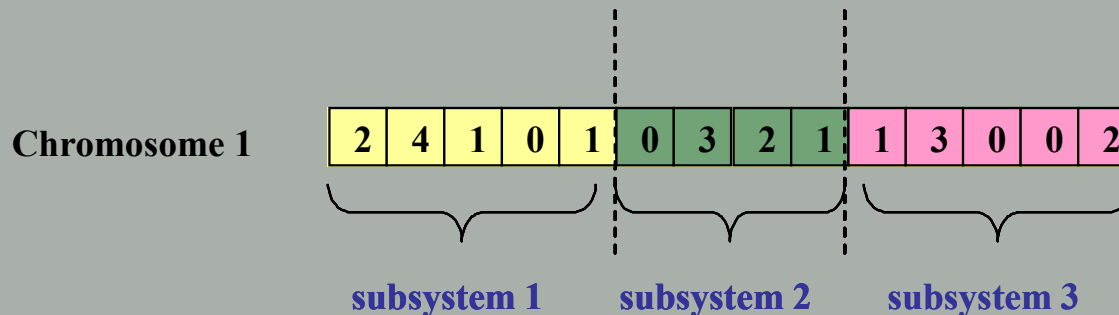
EXAMPLE:

- 3 subsystems
- Option of 5, 4 and 5 types of components in each subsystem

	R	C	W	R	C	W	R	C	W
1	0.94	9	9	0.97	12	5	0.96	10	6
2	0.91	6	6	0.86	3	7	0.89	6	8
3	0.89	6	4	0.70	2	3	0.72	4	2
4	0.75	3	7	0.66	2	4	0.71	3	4
5	0.72	2	8				0.67	2	4

Subsystem 1 Subsystem 2 Subsystem 3

- Each integer corresponds to the number of redundant components of that type.



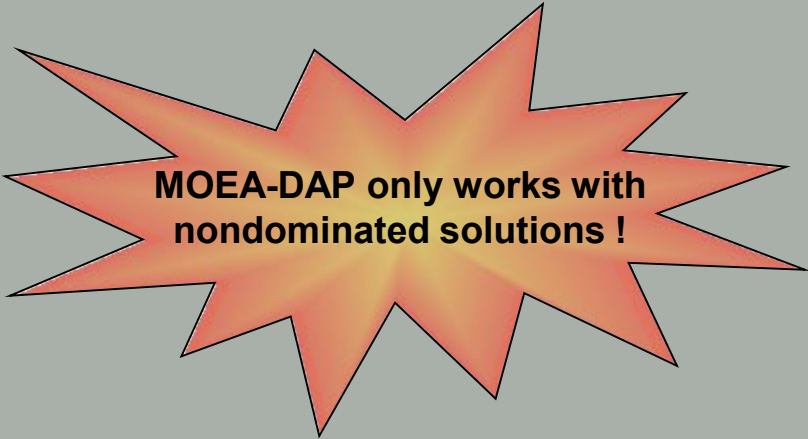
Step 2. Dominance count & Pareto dominance

EXAMPLE:

12 chromosomes are generated as initial population ---- $N_{pop} = 12$

Step 2.1 Perform Dominance count

Step 2.2 Check Pareto Dominance



**MOEA-DAP only works with
nondominated solutions !**

Nondominated set	Reliability	Cost	Weight	Dominance count
1	-0.98734	60	75	3
2	-0.94814	52	75	3
3	-0.96703	55	97	0
4	-0.98699	68	62	3
5	-0.95267	52	80	1
6	-0.79624	29	41	1
7	-0.89678	42	56	2
8	-0.92673	46	68	4

3. Fitness assignment

Step 3. Assign fitness to each solution

- Two different metrics to assign fitness to the solutions.

Fitness Metric 1: Distance-based, $f_1(i)$

- It is intended for maintaining population diversity.

Fitness Metric 2: Dominance count-based, $f_2(i)$

- It aims to select those individuals which are more dominating (intended to achieve proximity).

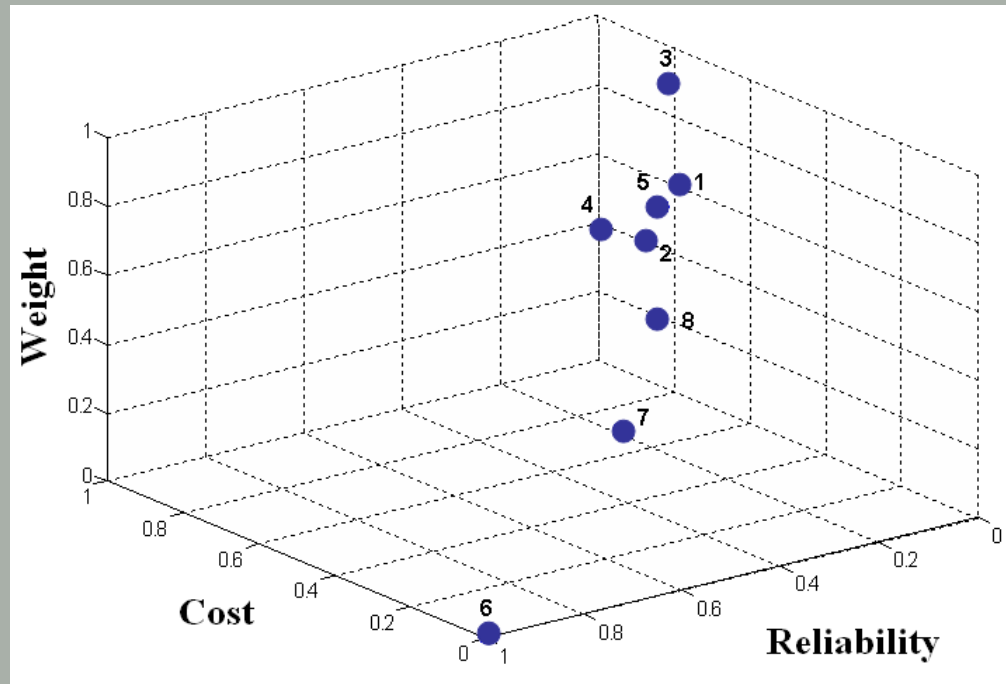
Aggregated Fitness Metric, $f_a(i)$: Fitness Metric 1 + Fitness Metric 2

- It aims to weight both metrics equally



Fitness Metric 1: Distance-based, $f_1(i)$, (cont'd...)

For this example, solution number 6 would receive the highest fitness value since it is the farthest solution with respect to the rest of the solutions.



First nondominated set - in a normalized space

Fitness metric 2: dominance count-based, $f_2(i)$

- Steps for the determination of the fitness value 2 are similar to those used in fitness value 1
- It is based on the dominance count concept

Nondominated set	Reliability	Cost	Weight	Dominance count	Fitness value 2
1	1	0.79487	0.60714	3	4
2	0.79487	0.58974	0.60714	3	4
3	0.89369	0.66667	1	0	1
4	0.99813	1	0.375	3	4
5	0.81856	0.58974	0.69643	1	2
6	0	0	0	1	2
7	0.52609	0.33333	0.26786	2	3
8	0.68281	0.4359	0.48214	4	5

Step 4. Selection

-Rank selection was used

-Rank-based fitness assignment behaves in a more robust manner than other fitness assignments, Whitley (1989) and Bäch & Hoffmeister (1991)

-Every nondominated individual receives an aggregated fitness value, $f_a(i)$, determined by their ranking from fitness metric 1 and 2

Solution	Fitness metric 1 $f_1(i)$	Fitness metric 2 $f_2(i)$	Aggregated Fitness value $f_a(i) = f_1(i) + f_2(i)$
1	1	4	5
2	1	4	5
3	2	1	3
4	2	4	6
5	1	2	3
6	5	2	7
7	2	3	5
8	1	5	6

Ranked Solutions	Aggregated Fitness value $f_a(i)$
6	7
4	6
8	6
1	5
2	5
7	5
3	3
5	3



Aggregated Fitness value

=

Fitness value 1

+

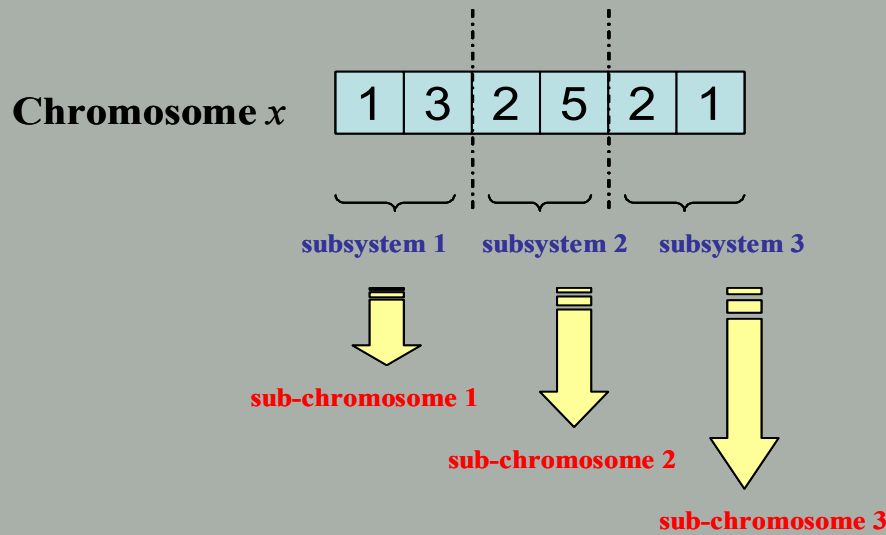
Fitness value 2

Step 5. Crossover

- Theorems exist (Fogel & Ghoseil, 1997) showing no intrinsic advantage in any given genetic representation
- Certain gene representations may be more effective and efficient in certain situations
- In MOEA-DAP, each solution, represented as a chromosome, has **s** number of sub-chromosomes

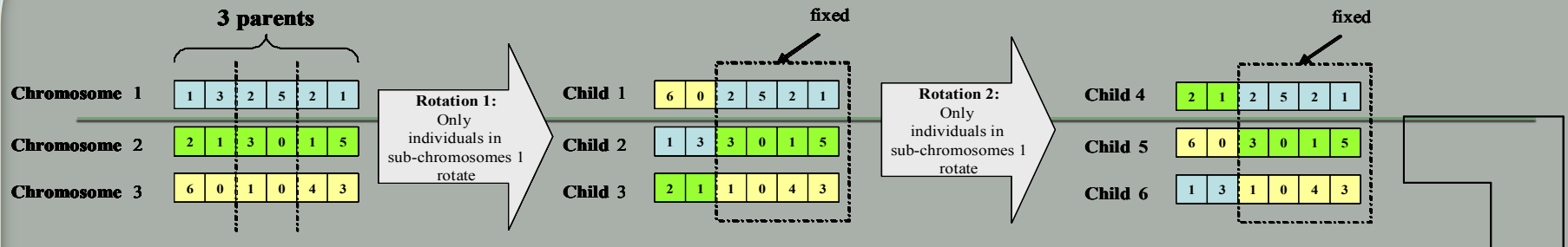
EXAMPLE: 3 subsystems

s = # of subsystems

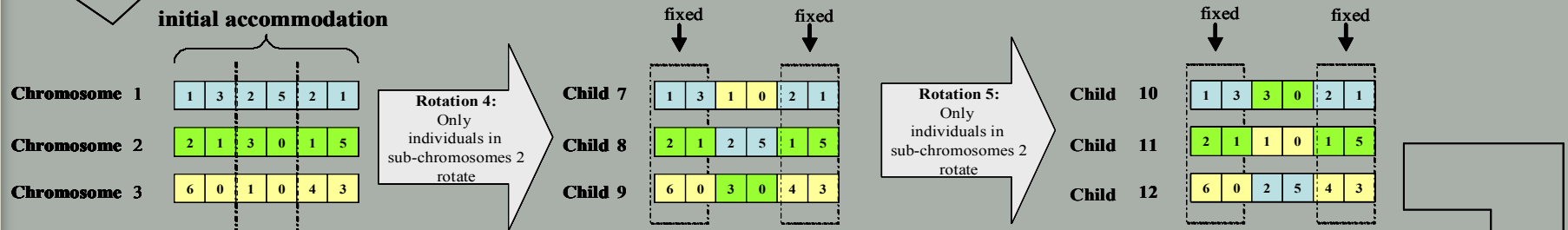


Each individual that can be selected for recombination has three sub-chromosomes

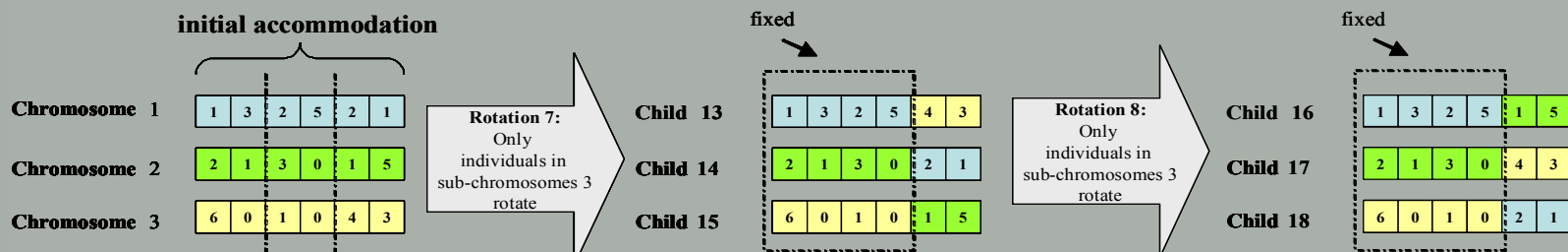
Crossover operation example (SURC)



In the third rotation of individuals in sub-chromosome 1, chromosomes return to the initial accommodation



In the sixth rotation of individuals in sub-chromosome 2, chromosomes return to the initial accommodation



To promote diversity, the greater the variety of genes available in the mating pool, the greater the likelihood of the system to identify alternate

MOEA-DAP: the whole picture

Input

parameters:

$N_{pop} = 20$

$P_{cross} = 0.75$

$P_{mut} = 0.1$

$s = 3$

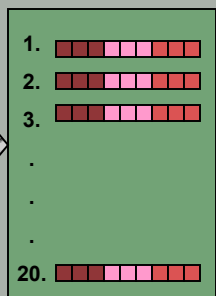
$\%elitism = 25\%$

$N_{gen} = 10$

$N_{elite} = \%elitism (N_{dom})$

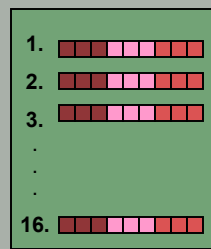
$N_{elite} = 0.25 (16) = 4$

Initial population



Check Pareto
Dominance

Nondominated
individuals



$f_1(i)$

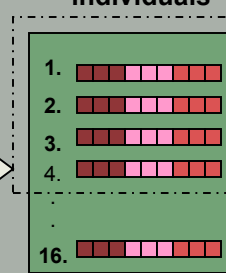
+

$f_2(i)$

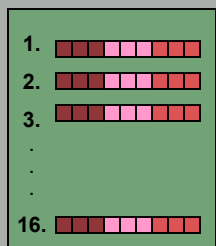
=

$f_a(i)$

Ranked
individuals



Rank selection



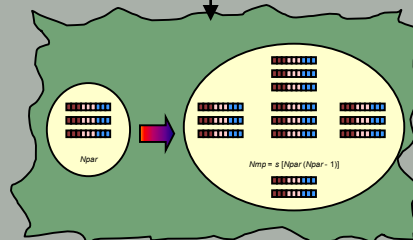
$N_{par} = P_{cross} (N_{dom})$

$N_{par} = 0.75 (16) = 12$

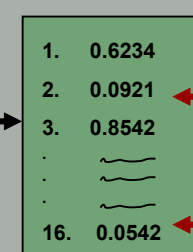
Selection of parents
to perform crossover

$N_{mp} = s [N_{par} (N_{par}-1)]$

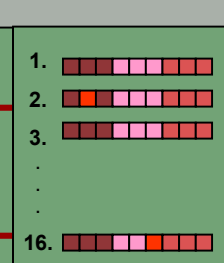
$N_{mp} = 3 [12(11)] = 396$



Random
numbers



Mutated list



$P_{mut} = 0.1$

Next population

$N_{elite} + \text{Mutated list}$

$4 + 16 = 20$

Next population

Performance comparison: MOEA-DAP vs. NSGA-II

system reliability

system cost

system weight

$$\max \left[\prod_{i=1}^s R_i(\mathbf{x}_i) \right], \min \left[\sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij} \right], \min \left[\sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \right]$$

subject to:

$$1 \leq \sum_{j=1}^{m_i} x_{ij} \leq n_{\max,i} \quad \text{for } \forall i = 1, 2, \dots, s$$

$$x_{ij} \in \{0, 1, 2, \dots\}$$

EXAMPLE:

- 3 subsystems

- Option of 5, 4 and 5 types of components in each subsystem

-10 runs (NSGA-II)

-10 runs (MOEA-DAP)

Input parameters: $N_{pop}=50$ $N_{gen}=100$
 $P_{cross}=0.8$ $P_{mut}=0.08$

	R	C	W	R	C	W	R	C	W
1	0.94	9	9	0.97	12	5	0.96	10	6
2	0.91	6	6	0.86	3	7	0.89	6	8
3	0.89	6	4	0.70	2	3	0.72	4	2
4	0.75	3	7	0.66	2	4	0.71	3	4
5	0.72	2	8				0.67	2	4
	Subsystem 1			Subsystem 2			Subsystem 3		

NSGA and NSGA-II (Deb *et al*)

- Nondominated Sorting Genetic Algorithm –NSGA
- Population based search algorithm
- Find nondominated solution points simultaneously
 - Searches and identifies good solutions
 - Produces offsprings of the current generation
 - Performs crossover and mutation operations on decision variables or “genes”
- Assigns ranks to every solution to keep track of good solutions
 - Assigns a dummy fitness value that helps in ranking and reproduction
 - Works with any number of objective functions
- NSGA-II
 - Eliminates sharing parameter of NSGA by introducing a crowding distance parameter
 - Keeps the nondominated solutions found separately
 - Includes the nondominated solutions in the next generation
 - Faster algorithm than NSGA

Performance comparison (cont'd...)

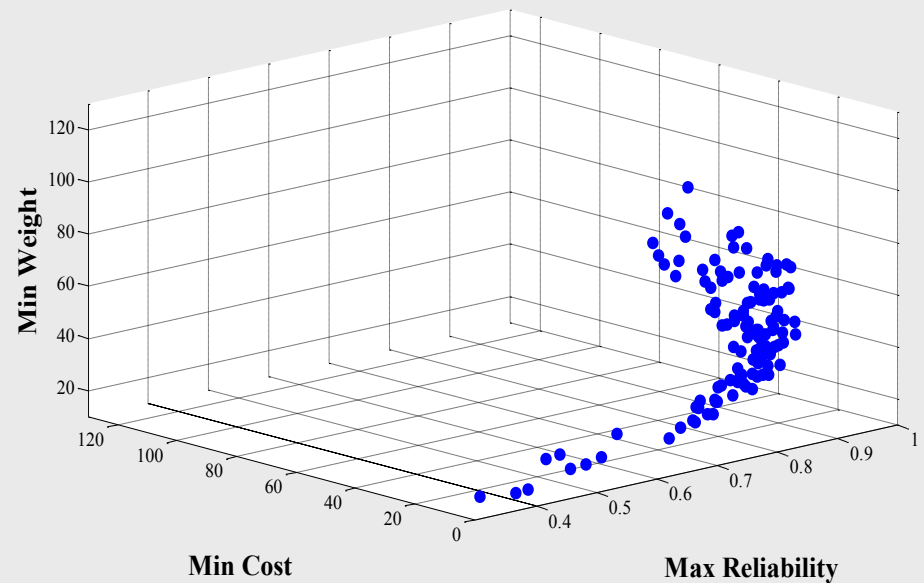
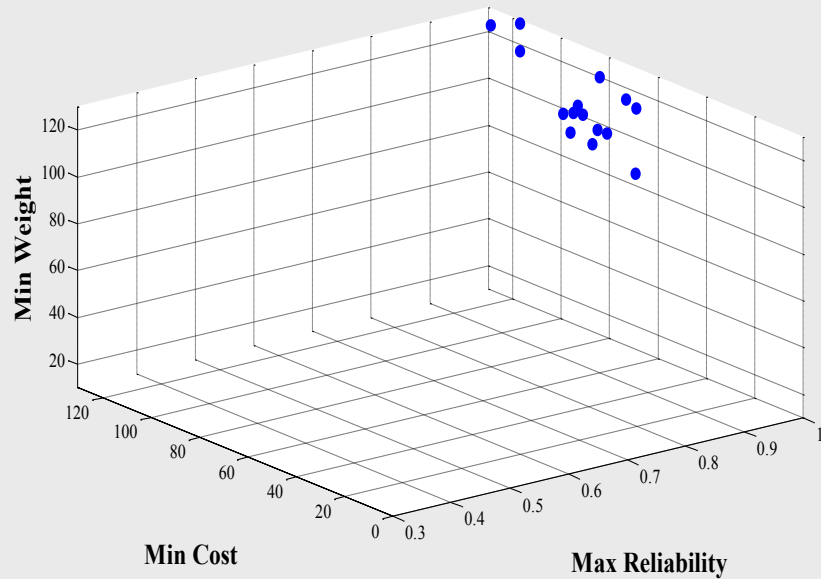
Table B.5 Nondominated solutions in Pareto front. Run 5, **NSGA-II**

NSGA-II: Original output				Non-duplicated nondominated solutions		
Solution number	Reliability	Cost	Weight	Reliability	Cost	Weight
1	0.96945	67	88	0.96945	67	88
2	0.99141	71	90	0.99141	71	90
3	0.96949	69	91	0.96949	69	91
4	0.99141	71	90			
5	0.99141	71	90			
6	0.96945	67	88			
.
.
.
50	0.96949	69	91			
Non-duplicated nondominated solutions				4		

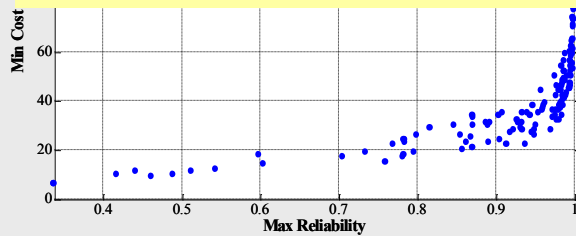
Table B.11 Nondominated solutions in Pareto front. Run 1, **MOEA-DAP**

Original output			
Solution number	Reliability	Cost	Weight
1	0.99981	77	68
2	0.90598	48	32
3	0.96026	40	37
4	0.87971	36	33
5	0.99819	74	54
6	0.99894	78	48
7	0.87959	36	27
8	0.99799	74	52
9	0.95929	42	35
.	.	.	.
.	.	.	.
.	.	.	.
25	0.99514	56	50
26	0.87106	34	21
27	0.97596	50	26
Non-duplicated nondominated solutions	27		

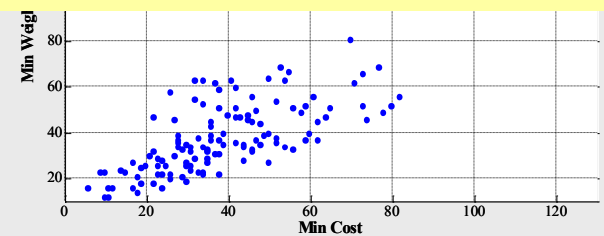
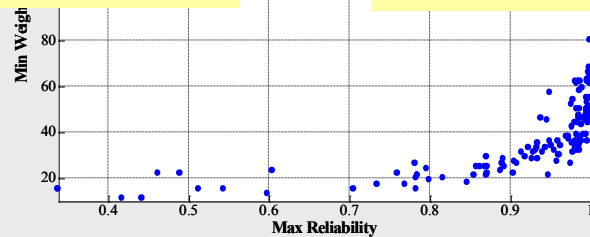
Performance comparison (cont'd...)



Nondominated solutions in Y_{true} obtained from NSGA-II algorithm



Nondominated solutions in Y_{true} obtained from MOEA-DAP algorithm



Nondominated solutions in Y_{true} obtained from MOEA-DAP algorithm

Conclusion

- The solutions obtained by the NSGA-II algorithm are not uniformly distributed along the Pareto front and mostly are extreme solutions, with very high reliability but also with large cost and weight.
- In contrast, we can observe that MOGA-DAP can obtain a set of non-dominated solutions fairly uniformly spreading along the Pareto frontier.
- **NSGA-II**
 - 1. Solutions not uniformly distributed along the Pareto front
 - 2. Extreme solutions
 - 3. General algorithm
- **MOEA-DAP**
 - 1. Uniform spread along the Pareto front
 - 2. Diverse solutions
 - 3. Problem-specific algorithm
- NSGA-II tends to spread quickly and appropriately when a certain nondominated region is found. However, it seems to have difficulties to generate nondominated solution vectors that lie in certain regions of the search space, Coello Coello & Toscano Pulido (2001).