제품의 다중 요구사항을 다루기 위한 충돌해결 방법론

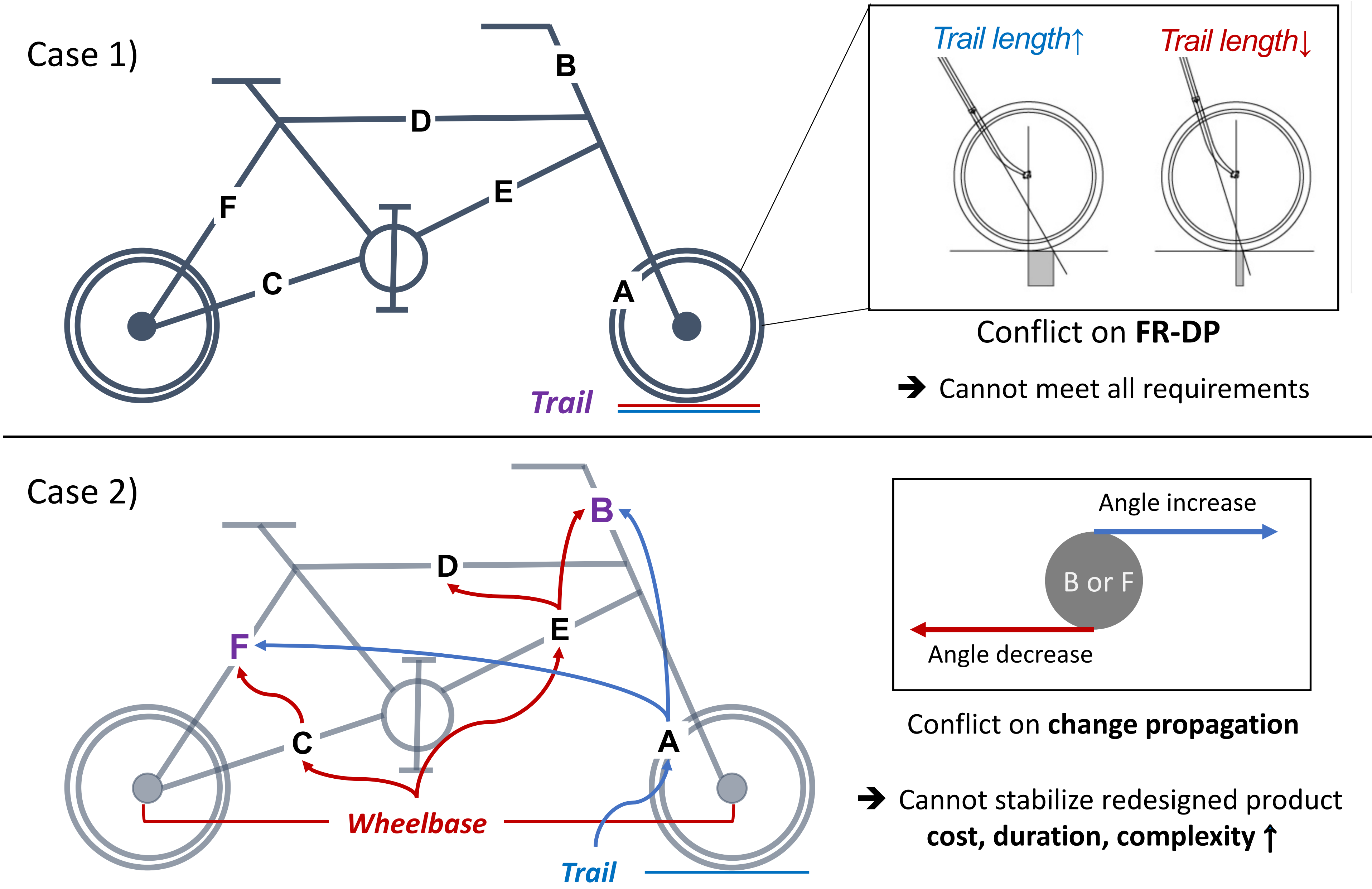
23.05.01 (월)

유 재 상

<서울대학교 제품서비스공학 연구실>

Direct and indirect conflicts when handling multi-functional requirements

FR$_1$) 조향성 **FR$_2$) 안전성**



문헌연구 (및 연구의 필요성)

- ❑ 신제품보다 기존제품의 요구사항을 반영하여 제품을 출시가 더 효과적.
  (Tavčar et al. 2005; Ercket. et al. 2004 cohen; et al.2000; Ahmad. et al. 2009)

- ❑ 제품의 기능 다양화에 따른 제품의 복잡성 증가(Cooper 1990; Meyer and Utterback 1995) 및 기업의 경쟁 가속화로 인한 다중 요구사항을 다룬 신제품 출시 필요성 증가 (Cooper 1990; Sbragia 2000)

- ❑ 그로 인해 여러 요구사항 사이에서의 충돌 발생 증가 그리고 설계 비용 및 기간 증가(Zhao et al. 2001)

기능 다양화 → 제품의 복잡성 증가

경쟁 가속화 → 여러 요구사항 반영

→ **충돌발생 증가** → 충돌로 인한 (cost, duration) 증가

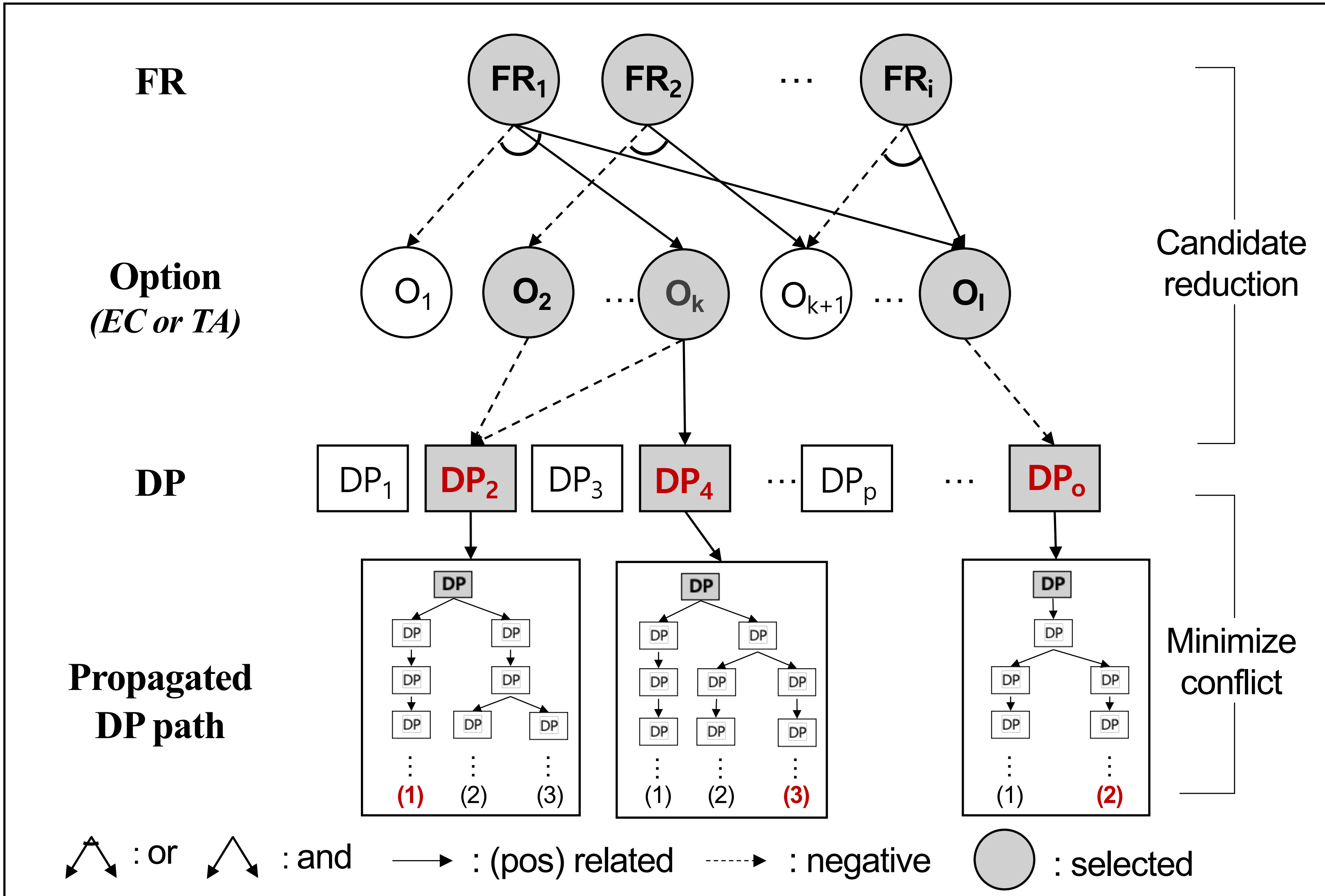
∴ 기업의 경쟁력 있는 설계변경관리를 위한 다중 요구사항 충돌해결 연구 필요

Comparing other researches in Engineering Change Management

	Ahmad. et al. (2009)	Jarratt (2011)	Ostrosi et al. (2012)	Koh et al. (2012)	Yang, et al. (2012)	Tang, et al. (2016)	Ullah et al. (2017)	Alireza et al. (2022)	This
Multi FRs			O	O				O	O
Multi candidates	O	O		O			O		O
Multi paths					O	O	O		O
Path Search					O		O		O
Conflict resolution			O					O	O

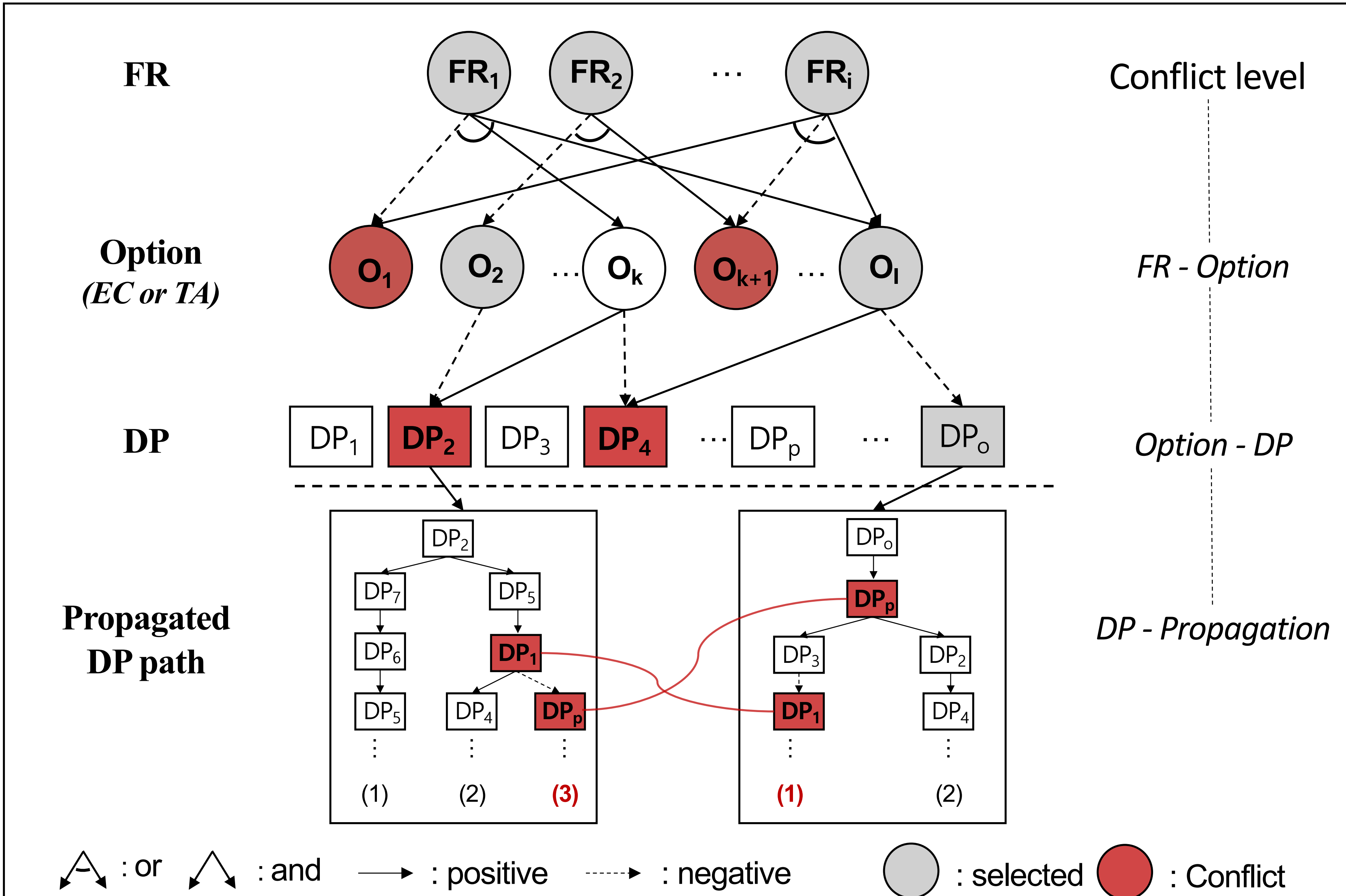
동일한 객체를 다른 방향으로의 조정을 요구하는 상황

연구목표 : 다중 요구 사항에서의 충돌을 최소화하기 위한 (1) **설계변수 조합 구성** 및 (2) **설계변수의 전파 경로 최적화**



TA : Technical Attribute **EC** : Engineering Characteristic

Before modeling) ‘Conflict’ on each level



TA : Technical Attribute **EC** : Engineering Characteristic

모델 수립을 위한 고려사항 (assumption)

❑ Functional requirement & Multiple options

- 기능적 요구사항을 충족하는 방법은 unique하지 않고 다양하다. (Jarratt, 2011; Ostrosi et al. 2012; Deutz et al. 2010)
- 각 재설계 방안은 **설계변수의 집합**으로 구성되며, 이는 요구사항을 충족하기 위한 최소한의 단위이다. (Suh, 1998; Marques et al. 2013)

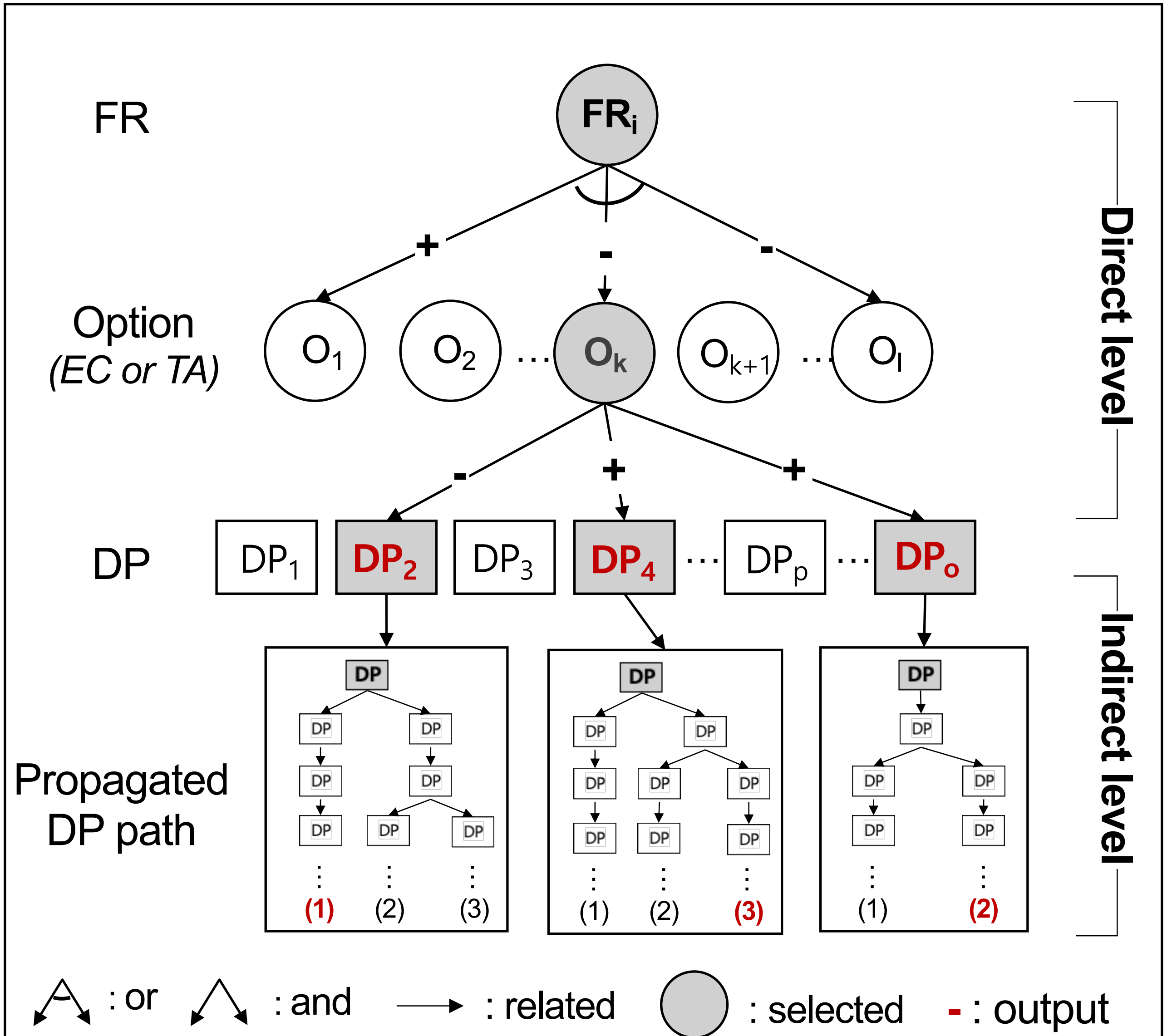
❑ Change propagation & Multiple path

- 변경전파의 핵심 목표는 요구사항이 아닌, **제품의 안정된 상태를 유지**하는 것이다.(Yang et al. 2012)
- 제품의 변경전파의 경로는 unique하지 않고, 어떤 경로를 선택하는지에 따라서 cost, duration, complexity가 달라진다. (Yang et al. 2012; Yin et al. 2022; Gan et al. 2021)
- **각 설계변수 및 부품의 논리관계**를 토대로 변경 경로를 도출할 수 있다. (Ma et al. 2016)

❑ Multiple functional requirement & Conflict

- 기능적 요구사항과 설계변수와의 관계는 1:1 대응이 아니기에 coupling이 존재한다. (Cohen, 2000)
- 특정 설계변수의 일정방향의 조정이 여러 요구사항을 모두 충족하지 못하는 상황을 발생할 수 있고, (Marques et al. 2013) 이를 **직접적 충돌**이라고 정의한다.
- change propagation path 상에서도, 설계변수가 **간접적 충돌**이 발생할 수 있다. (Haibing et al. 2021)

기능적 요구사항과 설계변수 간의 계층적 분해와 연결관계



FR – DP hierarchical decomposition

Given information

i) Possible *FR-Option*

- choose one **(OR)**
- with direction

ii) *Option-DP relation*

- DP sets **(AND)**
- with each direction

iii) *(Constraint, DP) linkage*

- Logical relation (**AND,OR**)
- Induce DP paths

EC : Engineering Characteristic
TA : Technical Attribute

Given information

Phase 1) Direct conflict	Phase 2) Indirect conflict

Given 1) **(PFO)** Possible *FR-Option* relation

$$\mathbf{PFO_{ij}} = \begin{cases} \mathbf{1} & (\text{If } Corr(Fr_i, Option_j) > 0) \\ \mathbf{0} & (\text{If } Corr(Fr_i, Option_j) = 0) \\ \mathbf{-1} & (\text{If } Corr(Fr_i, Option_j) < 0) \end{cases}$$

	Option 1	Option 2	Option 3	Option 4	…	Option m
FR 1	1	1	-1		…	1
FR 2		-1			…	
FR 3		1	1		…	
…	…	…	…	…	…	…
FR n	1			-1	…	-1

Given 2) **(OD)** *Option-DP relation*

$$\mathbf{OD_{kl}} = \begin{cases} \mathbf{1} & (\text{If } Corr(Option_k, DP_p) > 0) \\ \mathbf{0} & (\text{If } Corr(Option_k, DP_p) = 0) \\ \mathbf{-1} & (\text{If } Corr(Option_k, DP_p) < 0) \end{cases}$$

	DP1	DP2	DP3	DP4	…	DP l
Option 1	1	1			…	
Option 2			-1		…	
Option 3				1	…	
Option 4	-1				…	
…	…	…	…	…	…	…
Option m				-1	…	1

Given 3) key constraints and parameters linkage

	DP1	DP2	DP3	DP4	…	DP l
DP1		x			…	
DP 2			x		…	
DP 3				x	…	
DP 4	x				…	x
…	…	…	…	…	…	…
DP l				x	…	

Parameter linkage

$$f_1(\mathbf{DP}) \geq 0$$
$$f_2(\mathbf{DP}) \geq 0$$
$$\cdots$$
$$f_n(\mathbf{DP}) \geq 0$$

Key constraint

Induced) **(PM)** Path matrix

	DP1		DP2	DP3		…	DP l
Path	**(1)**	**(2)**	**(1)**	**(1)**	**(2)**	…	**(1)**
DP1	X	X		1	1	…	
DP2			x	-1		…	
DP3	1			X	X	…	1
…	…	…	…	…	…	…	…
DPl		-1	1		-1	…	X

N

$\mathbf{PM_N}$	DP 1	DP 2	DP 3	…	DP l
Path	$\mathbf{Path_{11}}$	$\mathbf{Path_{21}}$	$\mathbf{Path_{31}}$	…	$\mathbf{Path_{l1}}$
DP 1	X		1	…	
DP 2		X	-1	…	
DP 3	1		X	…	1
…	…	…	…	…	…
DP l		1		…	X

- Tensor to easily multiplication for path search

Full) conflict resolution for handling multi-FR with Change propagation path

$$min \quad \boldsymbol{CD} = \sum_{k(2)=1}^{l} \boldsymbol{I}\left(\sum_{k(1)=1}^{l} |TPk_{(1)k(2)|} \neq | \sum_{k(1)=1}^{l} TP_{k(1)k(2)} | \right)$$

$$where \quad DP_k = \operatorname{sgn}\left(\sum_{j=1}^{m} Opt_j * OD_{jk}\right) \qquad \textbf{(1)}$$

$$Opt_j = \operatorname{sgn}\left(\sum_{i=1}^{n} PFO_{ij} * x_{ij}\right)$$

$$TP_{kj} = \operatorname{sgn}\left(\sum_{t=1}^{T} Path^t{}_{kj}\right) \qquad \textbf{(2)}$$

$$path_k = \begin{cases} path^{t-1}{}_k * \left(\sum_{q=1}^{N} PM_q * y_q\right) & (if\ t \neq 0) \\ DP_k & (if\ t = 0) \end{cases}$$

$$s.t. \quad \sum_{j=1}^{m} PFO_{ij} * x_{ij} \neq 0 \qquad \textbf{(1-1)}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad x_{ij} \in [0,1] \qquad \textbf{(1-2)}$$

$$\sum_{i=1}^{n} |PFO_{ij} * x_{ij}| = |\sum_{i=1}^{n} PFO_{ij} * x_{ij}| \qquad \textbf{(1-3)}$$

$$\sum_{j=1}^{m} |Opt_j * OD_{jk}| = |\sum_{j=1}^{m} Opt_j * OD_{jk}| \qquad \textbf{(1-4)}$$

$$\sum_{v=1}^{N} y_q = 1 \qquad y_q \in [0,1] \qquad \textbf{(2-1)}$$

Nomenclature

- parameter

n : number of FR
m : number of option
l : number of DP
N : number of path matrix

i : index of FR
j : index of option
k : index of DP
q : index of path matrix

- Matrix

PFO : **P**ossible **FR**-**O**ption matrix
OD : **O**ption-**DP** relation matrix
PM : **P**ath **M**atrix

- Decision variable

x_{ij} : whether Option j is used to handle FR i
y_q : whether Path matrix q is used

- Output

Opt : selected Option list with sign
DP : selected DP list
$Path^t{}_k$: selected path matrix at t stage when DP_k is initiated
$TP_{k(1)k(2)}$: sign of $DP_{k(2)}$ when initiated k(1) is propagated

CD: conflict degree (or number)

Direct conflict) conflict resolution at FR-DP level (Choose initiating DPs)

FR-DP conflict resolution

$$find \quad DP\ list \quad (\boldsymbol{DP} = [1,0,-1,\dots,-1])$$

$$where \ DP_p = \operatorname{sgn}\left(\sum_{k=1}^{l} Opt_k * OD_{kp}\right)$$

$$Opt_j = \operatorname{sgn}\left(\sum_{j=1}^{m} PFO_{ij} * x_{ij}\right)$$

$$s.t. \quad \sum_{j=1}^{m} PFO_{ij} * x_{ij} \neq 0 \qquad \textbf{(1)}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad x_{ij} \in [0,1] \qquad \textbf{(2)}$$

$$\sum_{i=1}^{n} |PFO_{ij} * xi_j| = \left|\sum_{i=1}^{n} PFO_{ij} * x_{ij}\right| \qquad \textbf{(3)}$$

$$\sum_{j=1}^{m} |Opt_j * OD_{jk}| = \left|\sum_{j=1}^{m} Opt_j * OD_{jk}\right| \qquad \textbf{(4)}$$

First constraint) Choose one option per FR which is related

FR-DP conflict resolution

$$find \quad DP\ list \quad (\boldsymbol{DP} = [1,0,-1,\dots,-1])$$

$$where \quad DP_p = \mathrm{sgn}(\sum_{k=1}^{l} Opt_k * OD_{kp})$$

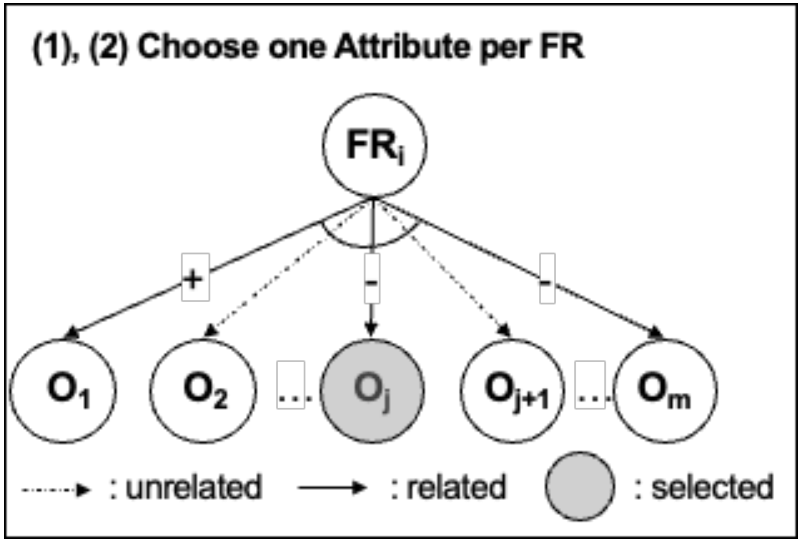
$$Opt_j = \mathrm{sgn}(\sum_{j=1}^{m} PFO_{ij} * x_{ij})$$

$$s.t. \quad \sum_{j=1}^{m} PFO_{ij} * x_{ij} \neq 0 \qquad \textbf{(1)}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad x_{ij} \in [0,1] \qquad \textbf{(2)}$$

$$\sum_{i=1}^{n} |PFO_{ij} * x_{ij}| = |\sum_{i=1}^{n} PFO_{ij} * x_{ij}| \qquad \textbf{(3)}$$

$$\sum_{j=1}^{m} |Opt_j * OD_{jk}| = |\sum_{j=1}^{m} Opt_j * OD_{jk}| \qquad \textbf{(4)}$$



	Option 1	Option 2	Option 3	Option 4	…	Option m
FR 1	1	1	-1		…	1
FR 2		-1			…	
FR 3		1	1		…	
…	…	…	…	…	…	…
FR n	1			-1	…	-1

(PFO) Possible *FR-Option* relation

Second constraint) OR gate logic to detect conflict on Option or DP level

FR-DP conflict resolution

$$find \quad DP\ list \quad (\boldsymbol{DP} = [1,0,-1,\ldots,-1])$$

$$where \quad DP_p = \operatorname{sgn}(\sum_{k=1}^{l} Opt_k * OD_{kp})$$

$$Opt_j = \operatorname{sgn}(\sum_{j=1}^{m} PFO_{ij} * x_{ij})$$

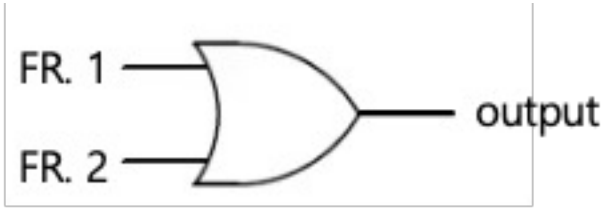
$$s.t. \quad \sum_{j=1}^{m} PFO_{ij} * x_{ij} \neq 0 \qquad \textbf{(1)}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad x_{ij} \in [0,1] \qquad \textbf{(2)}$$

$$\sum_{i=1}^{n} |PFO_{ij} * x_{ij}| = |\sum_{i=1}^{n} PFO_{ij} * x_{ij}| \qquad \textbf{(3)}$$

$$\sum_{j=1}^{m} |Opt_j * OD_{jk}| = |\sum_{j=1}^{m} Opt_j * OD_{jk}| \qquad \textbf{(4)}$$

(3, 4) extended OR gate logic



FR1	FR2	Output
-1	-1	-1
-1	0	-1
-1	**1**	**C**
0	-1	-1
0	0	0

FR1	FR2	Output
0	1	1
1	**-1**	**C**
1	0	1
1	1	1

$$\sum |x| = |(\sum x)|$$

→ 부호가 다른 경우는 **infeasible**

Review) conflict resolution at FR-DP level (Choose initiating DPs)

FR-DP conflict resolution

$$find \quad DP\ list \quad (\boldsymbol{DP} = [1,0,-1,\ldots,-1])$$

$$where \quad DP_p = \mathrm{sgn}\left(\sum_{k=1}^{l} Opt_k * OD_{kp}\right)$$

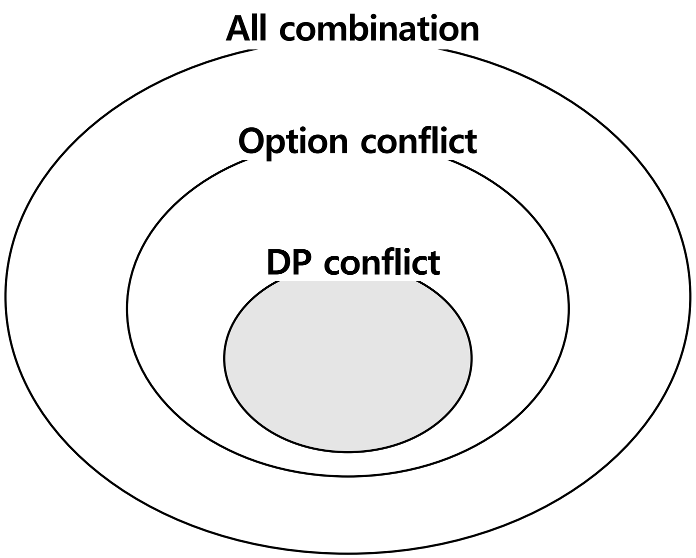
$$Opt_j = \mathrm{sgn}\left(\sum_{j=1}^{m} PFO_{ij} * x_{ij}\right)$$

$$s.t. \quad \sum_{j=1}^{m} PFO_{ij} * x_{ij} \neq 0 \qquad \textbf{(1)}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad x_{ij} \in [0,1] \qquad \textbf{(2)}$$

$$\sum_{i=1}^{n} |PFO_{ij} * x_{ij}| = |\sum_{i=1}^{n} PFO_{ij} * x_{ij}| \qquad \textbf{(3)}$$

$$\sum_{j=1}^{m} |Opt_j * OD_{jk}| = |\sum_{j=1}^{m} Opt_j * OD_{jk}| \qquad \textbf{(4)}$$



➔ Candidate reduction

Indirect conflict) detect **conflict** on each Propagated DP path (Choose path)

DP path conflict calculation

$$min \qquad \boldsymbol{CD} = \sum_{k(2)=1}^{l} \boldsymbol{I}\left(\sum_{k(1)=1}^{l} |TP_{k(1)k(2)|} \neq | \sum_{k(1)=1}^{l} TP_{k(1)k(2)}| \right)$$

$$where \qquad TP_{kr} = \operatorname{sgn}\left(\sum_{t=1}^{T} Path^{t}{}_{kr}\right)$$

Propagation을 모두 고려했을 시에 바뀌어야 할 DP의 전체 집합

$$path_k = \sum_{v=1}^{V(k)} Path\ List_{kv} * y_{kv})$$

$$Path\ List_k = \begin{cases} \mathbf{0} \quad [1 \times T \times l] & (if\ \boldsymbol{DP\ list_k} = 0) \\ Function\ \underline{path\ Searching}\ (\boldsymbol{DP\ list_k}, PM) & (if\ \boldsymbol{DP\ list_k} \neq 0) \end{cases}$$

$[V(k) \times T \times l]$

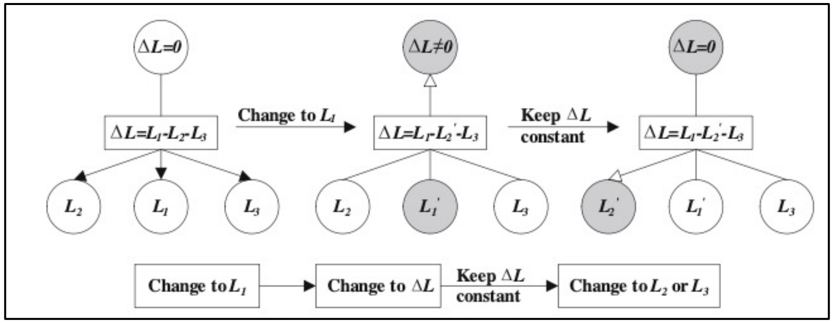
전파될 수 있는 경로들을 구하는 **재귀함수**

$$s.t. \qquad \sum_{v=1}^{V(k)} y_{kv} = 1 \qquad y_{kv} \in [0,1]$$

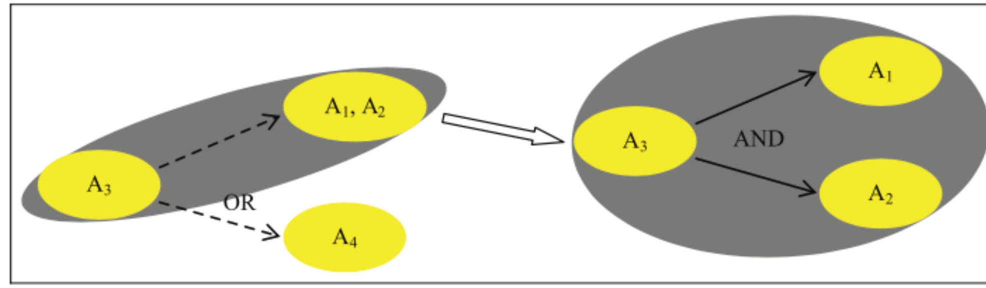
- PM : Path matrix
- V(k) : Total Propagation path
- T : Total propagation step
- l : Total Design parameter

Appendix i) the way to find change propagation Paths (path matrix)

Given 3) product's key constraints and parameters linkage



parameter linkage-based method(Yang , 2011)



Change propagation paths based on the logic relation (Tang, 2016)

1) Exist constraints to make product stable (**constraint linkage**)
2) Know the equation with parameter (**parameter linkage**)
3) Decompose equation with (**AND/OR)** logic structure

⇒ **Induce change propagation path depending on each DP**

1) Constraints

$$f_1(DP_1, DP_2, DP_3) \leq k_1$$

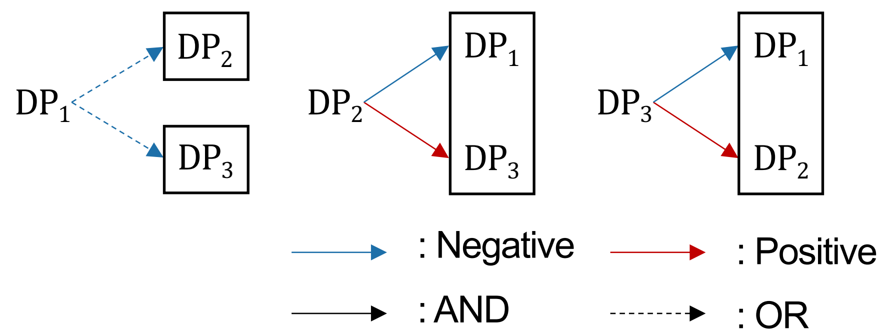
$$f_2(DP_1, DP_2, DP_3) \leq k_2$$

2) Parameter equation

$$DP_1 + DP_2 + DP_3 \leq k_1$$

$$DP_2 - DP_3 \leq k_2$$

3) Logical relation



4) Induce path matrix (PM)

PM_1	DP 1	DP 2	DP 3
Path	**$Path_{11}$**	**$Path_{21}$**	**$Path_{31}$**
DP 1	X	-1	-1
DP 2		X	1
DP 3	-1	1	X

PM_2	DP 1	DP 2	DP 3
path	**$Path_{12}$**	**$Path_{21}$**	**$Path_{31}$**
DP 1	X	-1	-1
DP 2	-1	X	1
DP 3		1	X

Appendix ii) Path search algorithm with recursive function

Induced 1) Path Matrix from linkage relation

N

PM_N	DP 1	DP 2	DP 3	…	DP l
Path	$Path_{11}$	$Path_{21}$	$Path_{31}$	…	$Path_{l1}$
DP 1	X		1	…	
DP 2		X	-1	…	
DP 3	1		X	…	1
…	…	…	…	…	…
DP l		1		…	X

1) $Path_{kq(k)}$ = q^{th} change path when DP_k changed

(# of path depends on DP)

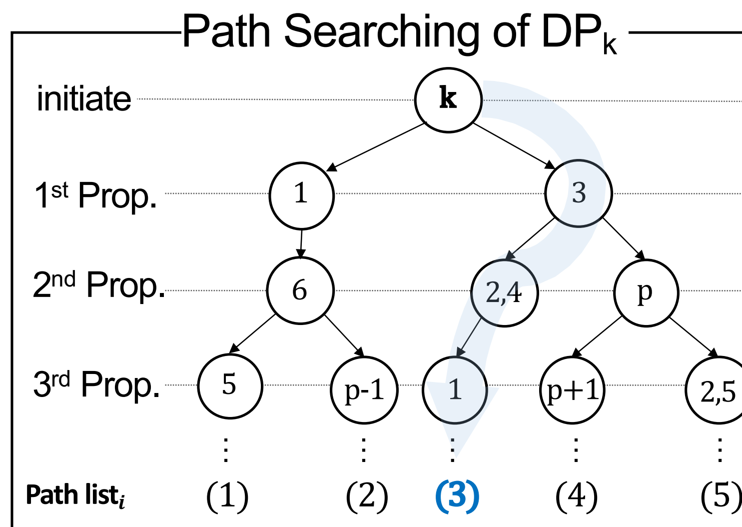
2) $Path_{kqr} = \begin{cases} 1 & (\text{If } Corr(DP_k, DP_r) > 0 \ \& \ DP_r \in Path_{pq}) \\ 0 & (\text{If } DP_r \notin Path_{pq}) \\ -1 & (\text{If } Corr(DP_k, DP_r) < 0 \ \& \ DP_r \in Path_{pq}) \end{cases}$

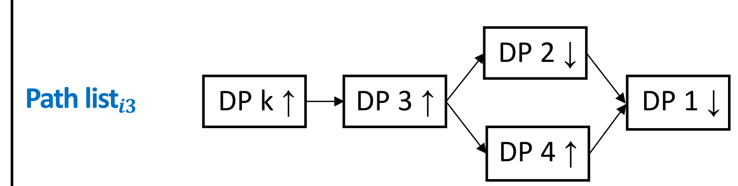
3) N (# of path matrix) = $\prod_{k=1} N(Path_k)$

Induced 2) DP list (from Direct conflict phase)

$DP\ list \quad (\boldsymbol{DP} = [1, 0, -1, \dots, -1])$

1	**0**	**-1**	**…**	**-1**





Function ***path Searching*** *(path, k)*

If k < K(propagation step) :

PM List = related PM candidates

For each path matrix list:

$Path^{k+1}{}_i = Path^k{}_i$ * PM

$Path^{k+1}{}_{ij} = \text{sign}(Path^k{}_{ij})$

Path List.append($Path^{k+1}{}_i$)

path Searching($Path^{k+1}{}_i$, PM, $k+1$)

Else:

Return Path List

Indirect conflict) choose each path to minimize conflict degree

DP path conflict calculation

$$min \quad \boldsymbol{CD} = \sum_{k(2)=1}^{l} \boldsymbol{I}\Big(\sum_{k(1)=1}^{l} |TPk_{{(1)}k{(2)}|} \neq | \sum_{k(1)=1}^{l} TP_{k{(1)}k{(2)}} | \Big)$$

$$where \quad TP_{k{(1)}k{(2)}} = \operatorname{sgn}\Big(\sum_{t=1}^{T} Path^{t}{}_{k{(1)}k{(2)}}\Big)$$

$$path_k = \sum_{v=1}^{V(k)} Path\ List_{kv} * y_{kv})$$

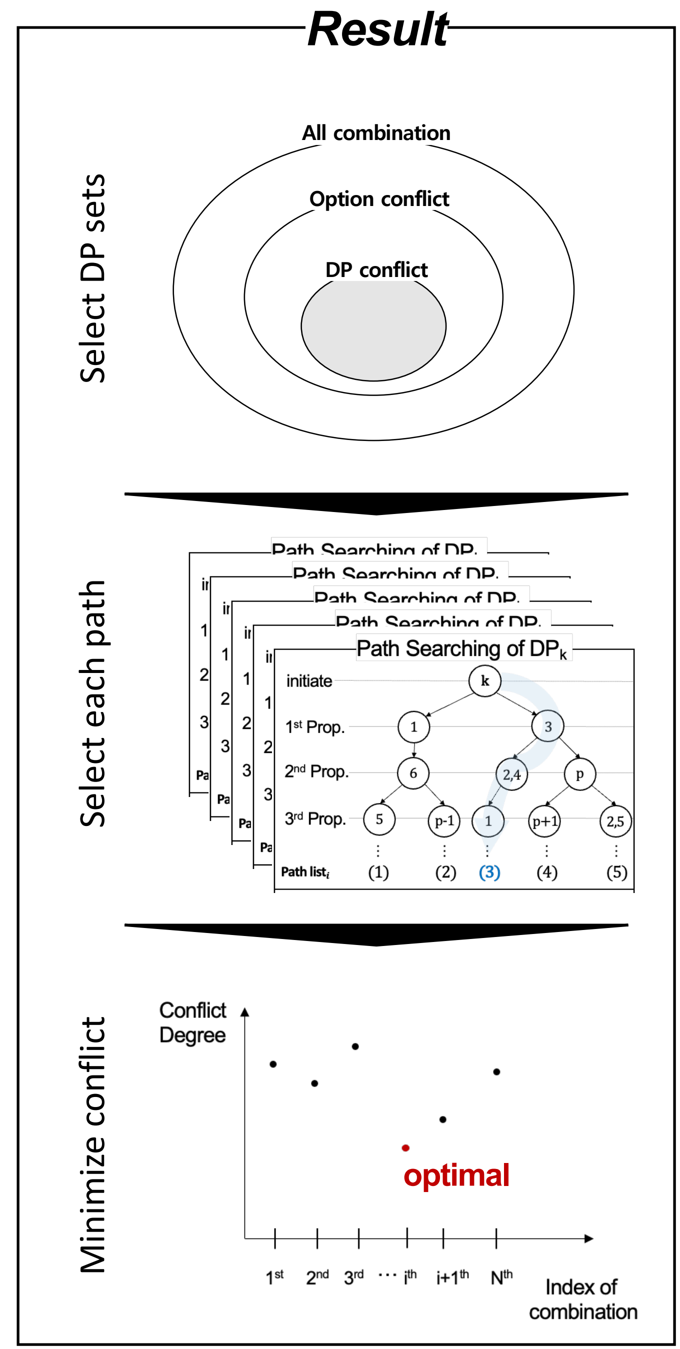
$$Path\ List_k = Function\ \textbf{path Searching}\ (\textbf{DP list}_k\ , PM)$$

$$s.t. \quad \sum_{v=1}^{V(k)} y_{kv} = 1 \qquad y_{\mathrm{kv}} \in [0,1]$$

---

*Function **path Searching** (path, k)*

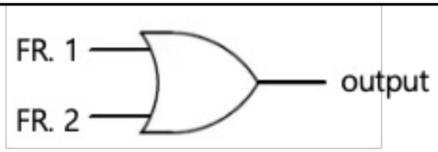
If k < K(propagation step) :
PM List = related PM candidates
For each path matrix list:
$Path^{k+1}{}_i = Path^{k}{}_i$ * PM
$Path^{k+1}{}_{ij} =$ sign($Path^{k}{}_{ij}$)
Path List.append($Path^{k+1}{}_i$)
path Searching($Path^{k+1}{}_i$,*PM, k+1*)
Else:
Return Path List



Case study) OR logic gate to detect whether existing conflict at direct conflict

Given 1) FR and each redesign options

		EC (or TA)			Design parameter					
FR	options	EC1	EC2	EC3	DP1	DP2	DP3	DP4	DP5	DP6
FR-1	1-1	1			1					
	1-2		1				1			
FR-2	2-1		-1				-1			
	2-2			1	-1					-1



FR1	FR2	Output
-1	-1	-1
-1	0	-1
-1	1	C
0	-1	-1
0	0	0

FR1	FR2	Output
0	1	1
1	-1	C
1	0	1
1	1	1

1-1	1	0	0	1	0	0	0	0	0
2-1	0	-1	0	0	0	-1	0	0	0

(1)	1	-1	0	1	0	-1	0	0	0

Feasible

1-1	1	0	0	1	0	0	0	0	0
2-2	0	0	1	-1	0	0	0	0	-1

(2)	1	0	1	C	0	0	0	0	-1

Infeasible on DP_1 (DP sets is smallest unit)

1-2	0	1	0	0	0	1	0	0	0
2-1	0	-1	0	0	0	-1	0	0	0

(3)	0	C	0	0	0	C	0	0	0

Infeasible on Option level

1-2	0	1	0	0	0	0	1	0	0
2-2	0	0	-1	-1	0	0	0	0	-1

(4)	0	1	1	-1	0	0	1	0	-1

Feasible

Indirect conflict) Path search and detect direction conflict on Propagated DP

Given 1) FR and each redesign options

		EC (or TA)			Design parameter					
FR	options	EC1	EC2	EC3	DP1	DP2	DP3	DP4	DP5	DP6
FR-1	1-1	1			1					
	1-2		1				1			
FR-2	2-1		-1				-1			
	2-2			1	-1					-1
	Output	1	-1	0	**1**	0	**-1**	0	0	0

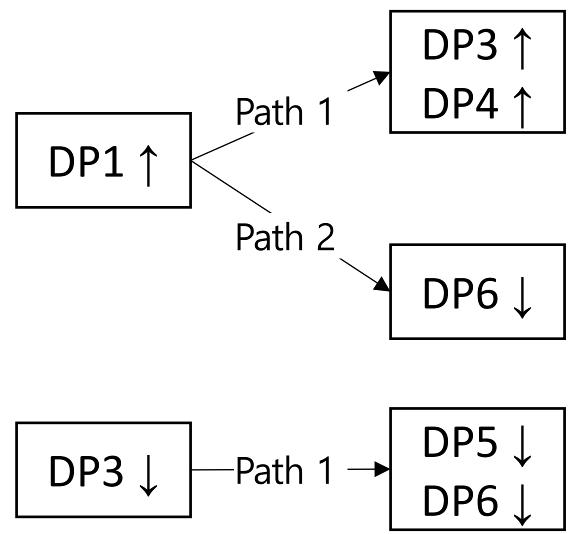
Given 2) change path for each DP

	DP1		DP2		DP3	DP4	DP5	DP6
Path	**(1)**	**(2)**	**(1)**	**(2)**	**(1)**	**(1)**	**(1)**	**(1)**
DP1	X	X		1				
DP2			X	X			1	
DP3	1				X			
DP4	1			1		X		1
DP5					1		X	
DP6		-1	1		1	1		X

i) Initiate DP and direction

- DP1 ↑ - DP3 ↓

ii) Path search of each DP



Path search with matrix multiplication (pre-processing)

	DP1		DP2		DP3	DP4	DP5	DP6
Path	**(1)**	**(2)**	**(1)**	**(2)**	**(1)**	**(1)**	**(1)**	**(1)**
DP1	X	X		1				
DP2			X	X			1	
DP3	1				X			
DP4	1			1		X		1
DP5					1		X	
DP6		-1	1		1	1		X

N = 4

	DP1	DP2	DP3	DP4	DP5	DP6
Path	**(1)**	**(1)**	**(1)**	**(1)**	**(1)**	**(1)**
DP1	X					
DP2		X			1	
DP3	1		X			
DP4	1			X		1
DP5			1		X	
DP6		1	1	1		X

$=\{PM_A\ PM_B\ PM_C\ PM_D\}$

1	**0**	**-1**	**0**	**0**	**0**

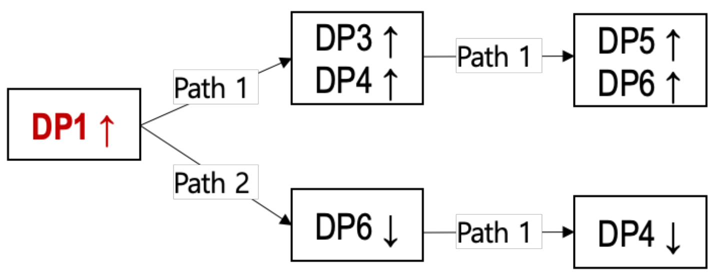
One-hot encoding

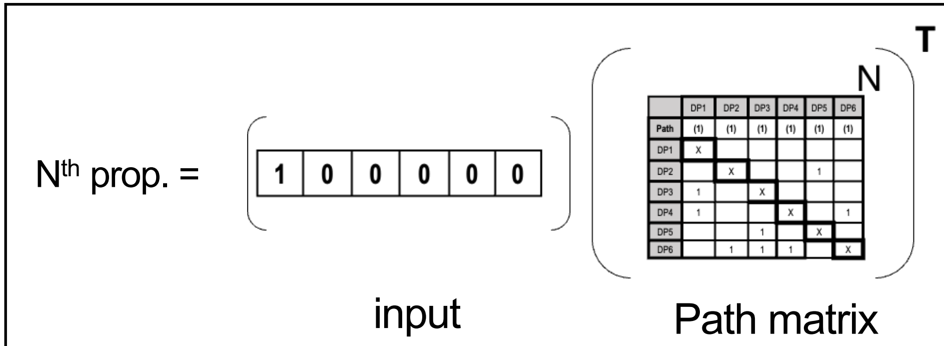
(1)

1	**0**	**0**	**0**	**0**	**0**

(2)

0	**0**	**-1**	**0**	**0**	**0**





(1) * A^T =

0	**0**	**1**	**1**	**0**	**0**

* A^T =

0	**0**	**0**	**0**	**1**	**1**

(1) * B^T =

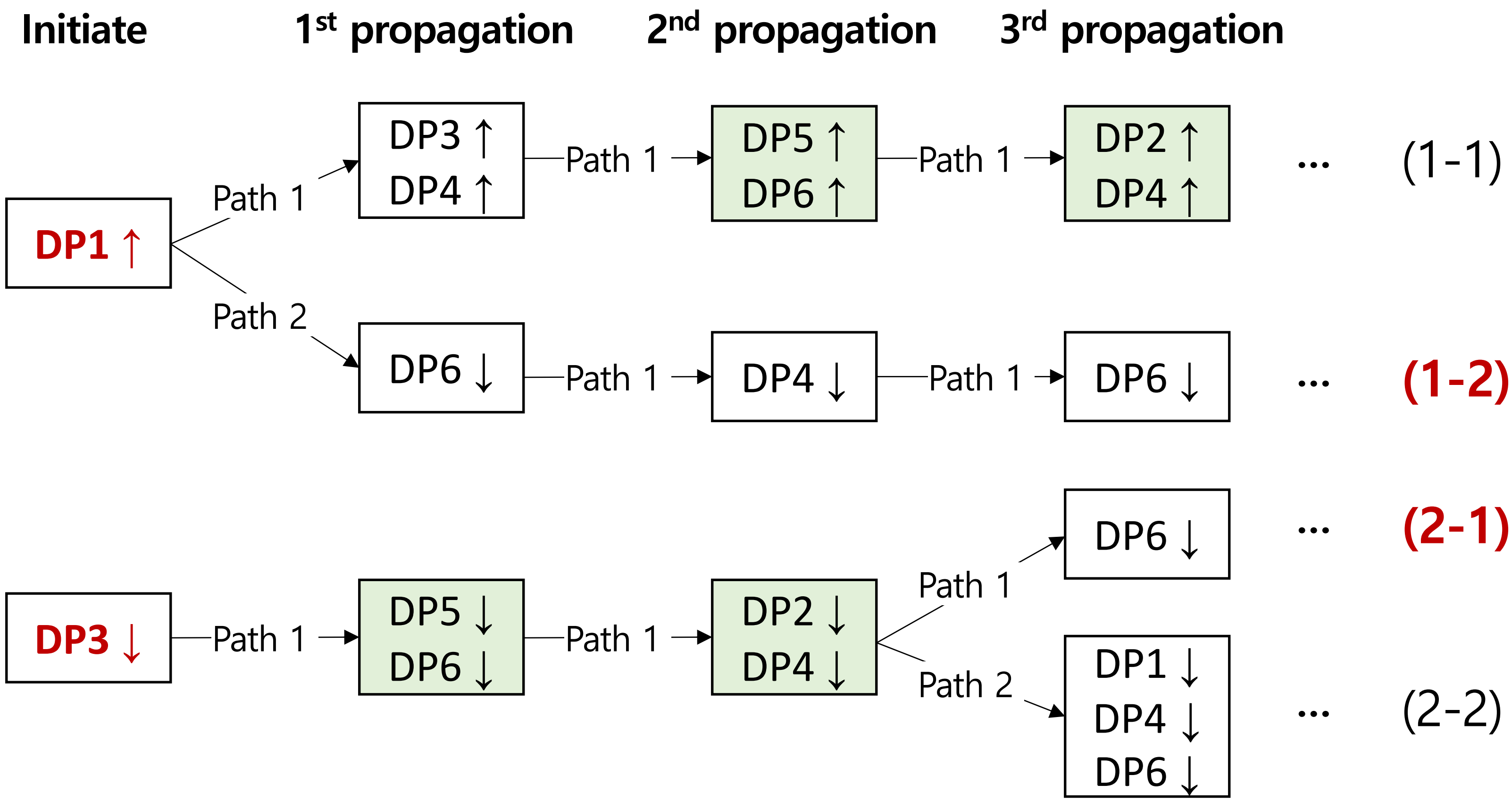
0	**0**	**0**	**0**	**0**	**-1**

* A^T =

0	**0**	**0**	**-1**	**0**	**0**

Appendix)

	DP1		DP2		DP3	DP4	DP5	DP6
Path	**(1)**	**(2)**	**(1)**	**(2)**	**(1)**	**(1)**	**(1)**	**(1)**
DP1	X	X		1				
DP2			X	X			1	
DP3	1				X			
DP4	1			1		X		1
DP5					1		X	
DP6		-1	1		1	1		X



Without impact (severity), count number of DP conflict

Conflict index

$$CI = \sum_{i=1}^{N} \boldsymbol{I}(sign(DP_{i1}) \neq sign(DP_{i2}))$$

Alter.	DP1	DP2	DP3	DP4	DP5	DP6	Conflict Index
1-1 2-1	1	1	1	1	1	1	**5**
	0	-1	-1	-1	-1	-1	
1-1 2-2	1	1	1	1	1	1	**6**
	-1	-1	-1	-1	-1	-1	
1-2 2-1	1	0	0	-1	0	-1	**0**
	0	-1	-1	-1	-1	-1	
1-2 2-2	1	0	0	-1	0	-1	**1**
	-1	-1	-1	-1	-1	-1	

Choose DP combination and path for minimizing conflict number

With impact and direction (conflict이 발생하더라도, severity)

	DP1		DP2		DP3	DP4	DP5	DP6
Path	(1)	(2)	(1)	(2)	(1)	(1)	(1)	(1)
DP1	X	X		0.3				
DP2			X	X			0.7	
DP3	0.3				X			
DP4	0.4			0.2		X		0.3
DP5					0.4		X	
DP6		-0.7	0.5		0.4	0.5		X

N = 4

	DP1	DP2	DP3	DP4	DP5	DP6
Path	(1)	(1)	(1)	(1)	(1)	(1)
DP1	X					
DP2		X			0.7	
DP3	0.3		X			
DP4	0.4			X		0.3
DP5			0.4		X	
DP6		0.5	0.4	0.5		X

{A,B,C,D}

1	0	-1	0	0	0

1	0	0	0	0	0	(1)
0	0	-1	0	0	0	(2)

(1-1) $\sum_{i=1}^{N} CL_{10} * (PM_A{}^{i})^T$

- Direct	1	0	0	0	0	0
- 1st	0	0	0.3	0.4	0	0
- 2nd	0	0	0	0	0.12	0.32
- 3rd	0	0.08	0	0.1	0	0
Output =	1	0.08	0.3	0.5	0.12	0.32

path

(2-1) $\sum_{i=1}^{N} CL_{30} * (PM_A{}^{i})^T$

- Direct	0	0	-1	0	0	0
- 1st	0	0	0	0	-0.4	-0.4
- 2nd	0	-0.28	0	-0.12	0	0
- 3rd	0	0	0	0	0	-0.2
Output =	0	-0.28	-1	-0.12	-0.4	-0.6

path

How to compare conflict degree?

Conflict impact

If A or B = 0 $CI = 0$

If |A| ≥ |B| $$CI = [1-(A+B)] + \frac{(A-B)}{2} = 1 - \frac{A}{2} - \frac{3B}{2}$$

If |B| ≥ |A| $$CI = [1+(A+B)] + \frac{(A-B)}{2} = 1 + \frac{3A}{2} + \frac{B}{2}$$

Alter.	DP1	DP2	DP3	DP4	DP5	DP6	Conflict Index
1-1 2-1	1	0.08	0.3	0.5	0.12	0.32	**4.99**
	0	-0.28	-1	-0.12	-0.4	-0.6	
1-1 2-2	1	0.08	0.3	0.5	0.12	0.32	**5.61**
	-0.08	-0.28	-1	-0.18	-0.4	-0.46	
1-2 2-1	1	0	0	-0.21	0	-0.8	**0**
	0	-0.28	-1	-0.12	-0.4	-0.6	
1-2 2-2	1	0	0	-0.21	0	-0.8	**0.62**
	-0.08	-0.28	-1	-0.18	-0.4	-0.46	

Choose DP combination and path for minimizing conflict

Case i) A+B 고정

A	B	A-B	A+B
1	-1	2	0
0.8	-0.8	1.6	0
0.5	-0.5	1	0
0.4	-0.4	0.8	0
0.1	-0.1	0.2	0

두 부호의 차
클 수록 Conflict

Case ii) A-B 고정

A	B	A-B	A+B
0.5	-0.5	1	0
0.6	-0.4	1	0.2
0.8	-0.2	1	0.6
0.9	-0.1	1	0.8

두 부호의 절대값 편차가
적을수록 Conflict

Case iii) B 고정

A	B	A-B	A+B
0.1	-0.1	0.2	0
0.2	-0.1	0.3	0.1
0.5	-0.1	0.6	0.4
1	-0.1	1.1	0.9

양수 값이
낮을수록 conflict

Case iv) A 고정

A	B	A-B	A+B
0.8	-0.8	1.6	0
0.8	-0.5	1.3	0.3
0.8	-0.3	1.1	0.5
0.8	-0.1	0.9	0.7

음수 값이
낮을수록 conflict

If |A| ≥ |B|)

$$CI = \underbrace{[1-(A+B)]}_{\text{(ii)}} + \underbrace{\frac{(A-B)}{2}}_{\text{(i)}} = 1 - \underbrace{\frac{A}{2}}_{\text{(iii)}} - \underbrace{\frac{3B}{2}}_{\text{(iv)}}$$

(i) , (ii), (iii), (iv)
모두 충족

Case i) A+B 고정

A	B	A-B	A+B
1	-1	2	0
0.8	-0.8	1.6	0
0.5	-0.5	1	0
0.4	-0.4	0.8	0
0.1	-0.1	0.2	0

두 부호의 차
클 수록 Conflict

Case ii) A-B 고정

A	B	A-B	A+B
0.5	-0.5	1	0
0.4	-0.6	1	-0.2
0.2	-0.8	1	-0.6
0.1	-0.9	1	-0.8

두 부호의 절대값 편차가
적을수록 Conflict

Case iii) A 고정

A	B	A-B	A+B
0.1	-0.1	0.2	0
0.1	-0.2	0.3	-0.1
0.1	-0.5	0.6	-0.4
0.1	-1	1.1	-0.9

음수 값이
낮을수록 conflict

Case iv) B 고정

A	B	A-B	A+B
0.8	-0.8	1.6	0
0.5	-0.8	1.3	-0.3
0.3	-0.8	1.1	-0.5
0.1	-0.8	0.9	-0.7

양수 값이
높을수록 conflict

If |B| ≥ |A|)

$$CI = \underbrace{[1+(A+B)]}_{\text{(ii)}} + \underbrace{\frac{(A-B)}{2}}_{\text{(i)}} = 1 + \underbrace{\frac{3A}{2}}_{\text{(iv)}} + \underbrace{\frac{B}{2}}_{\text{(iii)}}$$

(i) , (ii), (iii), (iv)
모두 충족