Counterexamples and Modification to the Termination and Optimality of ADOPT-based Algorithms (Supplementary Material)

Koji Noshiro[a,*], Koji Hasebe[b]

[a]*Degree Programs in Systems and Information Engineering, University of Tsukuba, Japan*
[b]*Department of Computer Science, University of Tsukuba, Japan*

Appendix A. Traces of Counterexamples

In this section, we show the detailed traces of the counterexamples in Section 3. The traces are composed of figures that show sequences of agent actions and tables that show states of agents. In a figure, shaded agents (i.e., nodes) receive messages, process them, and send new messages. Received messages are represented by arrows whose heads are painted in black, while sent messages are represented by arrows whose heads are not painted. We omit some messages, e.g., most THRESHOLD messages, that are not crucial. Additionally, tables show the states of agents when they complete processing the received messages, and some trivial states, e.g., the current context of a root agent, are omitted.

Appendix A.1. Counterexample to Termination

Figures A.1–A.27 and Tables A.1–A.11 show the trace of the counterexample to termination, described in 3.1.

*Corresponding author
Email addresses: noshiro@mas.cs.tsukuba.ac.jp (Koji Noshiro), hasebe@cs.tsukuba.ac.jp (Koji Hasebe)
URL: https://mas.cs.tsukuba.ac.jp/~noshiro (Koji Noshiro)

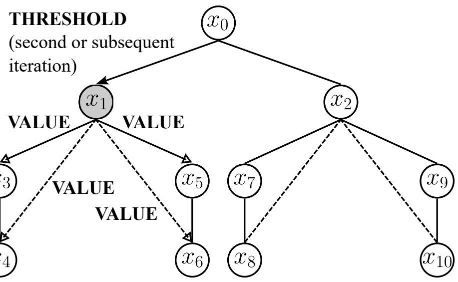

Figure A.1: Step 1 (1). x_1 sends VALUE messages to its children and pseudo-children x_3, x_4, x_5, and x_6.

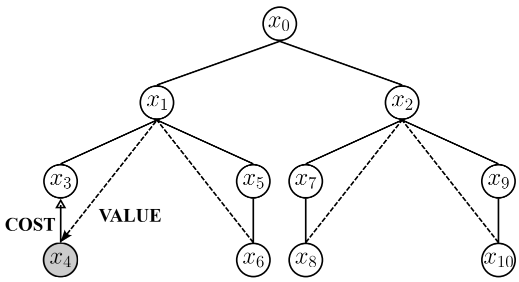

Figure A.2: Step 1 (2). x_4 receives the VALUE message and computes the lower bound using the updated context $CX_4 \ni (x_1, 0)$, and thus $LB_4 = 5$. Then, x_4 reports the lower bound to x_3 by sending a COST message.

Table A.1: States of agents in Steps 0–1 (2).

Variables	Step 0	Step 1 (1)	Step 1 (2)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	∞	∞
TH_0	0	0	0
$th_0(0, x_1)$	0	0	0
$th_0(0, x_2)$	0	0	0
x_1			
d_1	0	0	0
LB_1	0	0	0
$lb_1(0, x_3)$	0	0	0
$lb_1(0, x_5)$	0	0	0
$lb_1(1, x_3)$	0	0	0
$lb_1(1, x_5)$	0	0	0
UB_1	∞	∞	∞
$ub_1(0, x_3)$	∞	∞	∞
$ub_1(0, x_5)$	∞	∞	∞
$ub_1(1, x_3)$	∞	∞	∞
$ub_1(1, x_5)$	∞	∞	∞
TH_1	0	0	0
x_3			
d_3	0	0	0
CX_3	{}	{}	{}
$cx_3(0, x_4)$	{}	{}	{}
LB_3	0	0	0
$lb_3(0, x_4)$	0	0	0
UB_3	∞	∞	∞
$ub_3(0, x_4)$	∞	∞	∞
x_4			
d_4	0	0	0
CX_4	{}	{}	$\boldsymbol{\{(x_1, 0), (x_3, 0)\}}$
LB_4	0	0	**5**
UB_4	∞	∞	**5**
x_5			
d_5	0	0	0
CX_5	{}	{}	{}
$cx_5(0, x_6)$	{}	{}	{}
LB_5	0	0	0
$lb_5(0, x_6)$	0	0	0
UB_5	∞	∞	∞
$ub_5(0, x_6)$	∞	∞	∞
x_6			
d_6	0	0	0
CX_6	{}	{}	{}
LB_6	0	0	0
UB_6	∞	∞	∞

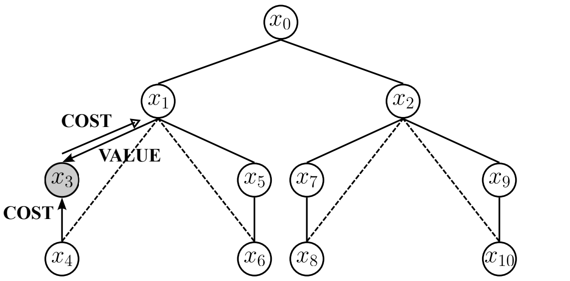

Figure A.3: Step 1 (3). x_3 receives the VALUE message from x_1 and the COST message from x_4 and computes the lower bound as $LB_3 = 5$. Then, x_3 sends a COST message with $CX_3 \ni (x_1, 0)$ and $LB_3 = 5$ to x_1.

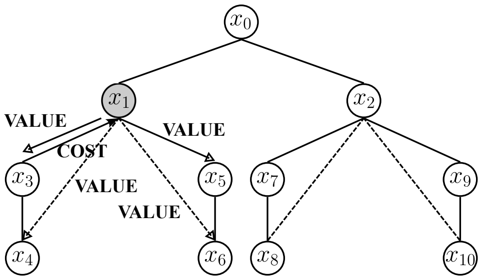

Figure A.4: Step 1 (4). After x_1 receives the COST message, x_1 updates $LB_1(0)$ to 5. Since $LB_1(0) = 5 > TH_1 = 0$, x_1 changes its value d_1 to 1 and sends VALUE messages to its children and pseudo-children.

Table A.2: States of agents in Steps 1 (3)–1 (4).

Variables	Step 1 (3)	Step 1 (4)
x_0		
d_0	0	0
LB_0	0	0
$lb_0(0, x_1)$	0	0
$lb_0(0, x_2)$	0	0
UB_0	∞	∞
$ub_0(0, x_1)$	∞	∞
$ub_0(0, x_2)$	∞	∞
TH_0	0	0
$th_0(0, x_1)$	0	0
$th_0(0, x_2)$	0	0
x_1		
d_1	0	**1**
LB_1	0	0
$lb_1(0, x_3)$	0	**5**
$lb_1(0, x_5)$	0	0
$lb_1(1, x_3)$	0	0
$lb_1(1, x_5)$	0	0
UB_1	∞	∞
$ub_1(0, x_3)$	∞	**5**
$ub_1(0, x_5)$	∞	∞
$ub_1(1, x_3)$	∞	∞
$ub_1(1, x_5)$	∞	∞
TH_1	0	0
x_3		
d_3	0	0
CX_3	$\boldsymbol{\{(x_1, 0)\}}$	$\{(x_1, 0)\}$
$cx_3(0, x_4)$	$\boldsymbol{\{(x_1, 0)\}}$	$\{(x_1, 0)\}$
LB_3	**5**	5
$lb_3(0, x_4)$	**5**	5
UB_3	**5**	5
$ub_3(0, x_4)$	**5**	5
x_4		
d_4	0	0
CX_4	$\{(x_1, 0), (x_3, 0)\}$	$\{(x_1, 0), (x_3, 0)\}$
LB_4	5	5
UB_4	5	5
x_5		
d_5	0	0
CX_5	{}	{}
$cx_5(0, x_6)$	{}	{}
LB_5	0	0
$lb_5(0, x_6)$	0	0
UB_5	∞	∞
$ub_5(0, x_6)$	∞	∞
x_6		
d_6	0	0
CX_6	{}	{}
LB_6	0	0
UB_6	∞	∞

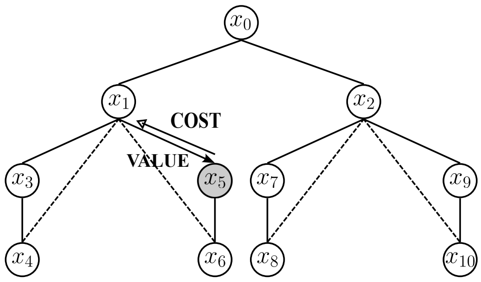

Figure A.5: Step 2 (1). x_5 receives the VALUE messages from x_1 and sends a COST message to x_1, but this COST message does not affect the bounds of x_1 because $LB_5 = 0$ and $UB_5 = \infty$, which are the initial bounds.

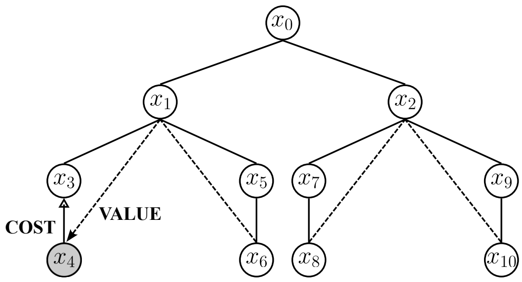

Figure A.6: Step 2 (2). Similar to the procedure in Step 1, x_4 receives and sends messages.

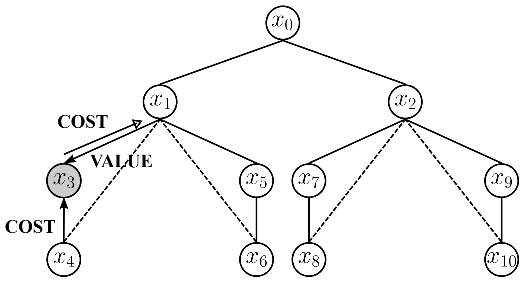

Figure A.7: Step 2 (3). Similar to the procedure in Step 1, x_3 receives and sends messages.

Table A.3: States of agents in Steps 2 (1)–2 (3).

Variables	Step 2 (1)	Step 2 (2)	Step 2 (3)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	∞	∞
TH_0	0	0	0
$th_0(0, x_1)$	0	0	0
$th_0(0, x_2)$	0	0	0
x_1			
d_1	1	1	1
LB_1	0	0	0
$lb_1(0, x_3)$	5	5	5
$lb_1(0, x_5)$	0	0	0
$lb_1(1, x_3)$	0	0	0
$lb_1(1, x_5)$	0	0	0
UB_1	∞	∞	∞
$ub_1(0, x_3)$	5	5	5
$ub_1(0, x_5)$	∞	∞	∞
$ub_1(1, x_3)$	∞	∞	∞
$ub_1(1, x_5)$	∞	∞	∞
TH_1	0	0	0
x_3			
d_3	0	0	0
CX_3	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
$cx_3(0, x_4)$	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
UB_3	5	5	**6**
$lb_3(0, x_4)$	5	5	**6**
UB_3	5	5	**6**
$ub_3(0, x_4)$	5	5	**6**
x_4			
d_4	0	0	0
CX_4	$\{(x_1, 0),$ $(x_3, 0)\}$	$\boldsymbol{\{(x_1, 1),}$ $\boldsymbol{(x_3, 0)\}}$	$\{(x_1, 1),$ $(x_3, 0)\}$
LB_4	5	**6**	6
UB_4	5	**6**	6
x_5			
d_5	0	0	0
CX_5	{}	{}	{}
$cx_5(0, x_6)$	{}	{}	{}
LB_5	0	0	0
$lb_5(0, x_6)$	0	0	0
UB_5	∞	∞	∞
$ub_5(0, x_6)$	∞	∞	∞
x_6			
d_6	0	0	0
CX_6	{}	{}	{}
LB_6	0	0	0
UB_6	∞	∞	∞

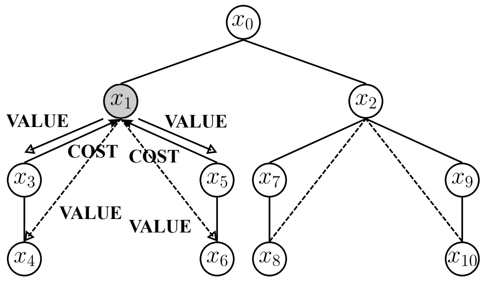

Figure A.8: Step 2 (4). Similar to the procedure in Step 1, x_1 receives and sends messages, and then x_1 updates the lower bound as $LB_1(1) = 6$. Additionally, the threshold of x_1 increases as $TH_1 = LB_1 = \min\{LB_1(0), LB_1(1)\} = 5$ because of ThresholdInvariant. Since $LB_1(1) = 6 > TH_1 = 5$, x_1 changes its value d_1 back to 0.

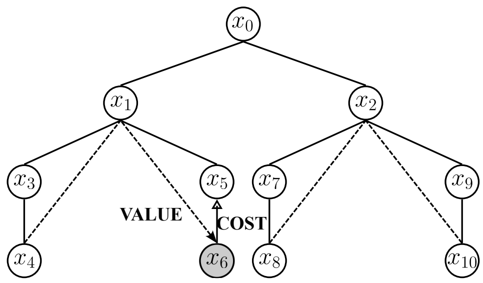

Figure A.9: Step 3 (1). Similar cost calculations and value changes are performed in x_6.

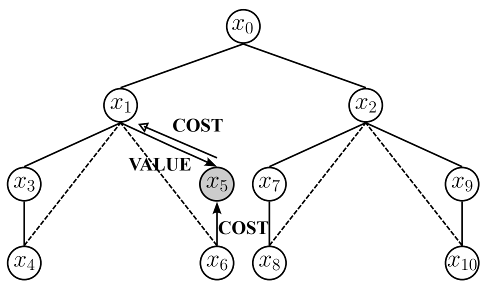

Figure A.10: Step 3 (2). Similar cost calculations and value changes are performed in x_5.

Table A.4: States of agents in Steps 2 (4)–3 (2).

Variables	Step 2 (4)	Step 3 (1)	Step 3 (2)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	∞	∞
TH_0	0	0	0
$th_0(0, x_1)$	0	0	0
$th_0(0, x_2)$	0	0	0
x_1			
d_1	**0**	0	0
LB_1	**5**	5	5
$lb_1(0, x_3)$	5	5	5
$lb_1(0, x_5)$	0	0	0
$lb_1(1, x_3)$	**6**	6	6
$lb_1(1, x_5)$	0	0	0
UB_1	∞	∞	∞
$ub_1(0, x_3)$	5	5	5
$ub_1(0, x_5)$	∞	∞	∞
$ub_1(1, x_3)$	**6**	6	6
$ub_1(1, x_5)$	∞	∞	∞
TH_1	**5**	5	5
x_3			
d_3	0	0	0
CX_3	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
$cx_3(0, x_4)$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
UB_3	6	6	6
$lb_3(0, x_4)$	6	6	6
UB_3	6	6	6
$ub_3(0, x_4)$	6	6	6
x_4			
d_4	0	0	0
CX_4	$\{(x_1, 1), (x_3, 0)\}$	$\{(x_1, 1), (x_3, 0)\}$	$\{(x_1, 1), (x_3, 0)\}$
LB_4	6	6	6
UB_4	6	6	6
x_5			
d_5	0	0	0
CX_5	$\{\}$	$\{\}$	$\boldsymbol{\{(x_1, 0)\}}$
$cx_5(0, x_6)$	$\{\}$	$\{\}$	$\boldsymbol{\{(x_1, 0)\}}$
LB_5	0	0	**2**
$lb_5(0, x_6)$	0	0	**2**
UB_5	∞	∞	**2**
$ub_5(0, x_6)$	∞	∞	**2**
x_6			
d_6	0	0	0
CX_6	$\{\}$	$\boldsymbol{\{(x_1, 0), (x_5, 0)\}}$	$\{(x_1, 0), (x_5, 0)\}$
LB_6	0	**2**	2
UB_6	∞	**2**	2

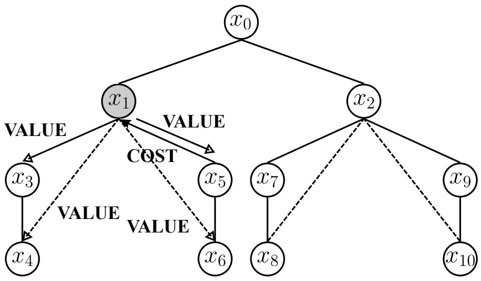

Figure A.11: Step 3 (3). Similar cost calculations and value changes are performed in x_1.

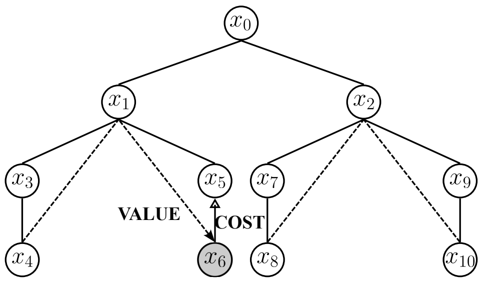

Figure A.12: Step 3 (4). Similar cost calculations and value changes are performed in x_6.

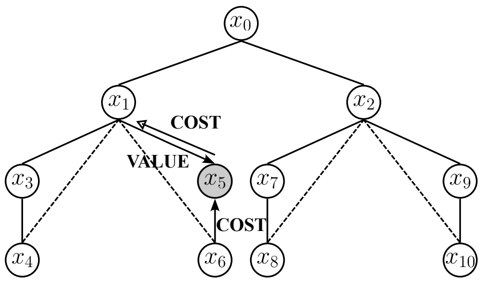

Figure A.13: Step 3 (5). Similar cost calculations and value changes are performed in x_5.

Table A.5: States of agents in Steps 3 (3)–3 (5).

Variables	Step 3 (3)	Step 3 (4)	Step 3 (5)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	∞	∞
TH_0	0	0	0
$th_0(0, x_1)$	0	0	0
$th_0(0, x_2)$	0	0	0
x_1			
d_1	**1**	1	1
LB_1	**6**	6	6
$lb_1(0, x_3)$	5	5	5
$lb_1(0, x_5)$	**2**	2	2
$lb_1(1, x_3)$	6	6	6
$lb_1(1, x_5)$	0	0	0
UB_1	**7**	7	7
$ub_1(0, x_3)$	5	5	5
$ub_1(0, x_5)$	**2**	2	2
$ub_1(1, x_3)$	6	6	6
$ub_1(1, x_5)$	∞	∞	∞
TH_1	**6**	6	6
x_3			
d_3	0	0	0
CX_3	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
$cx_3(0, x_4)$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
LB_3	6	6	6
$lb_3(0, x_4)$	6	6	6
UB_3	6	6	6
$ub_3(0, x_4)$	6	6	6
x_4			
d_4	0	0	0
CX_4	$\{(x_1, 1),$ $(x_3, 0)\}$	$\{(x_1, 1),$ $(x_3, 0)\}$	$\{(x_1, 1),$ $(x_3, 0)\}$
LB_4	6	6	6
UB_4	6	6	6
x_5			
d_5	0	0	0
CX_5	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
$cx_5(0, x_6)$	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
LB_5	2	2	**3**
$lb_5(0, x_6)$	2	2	**3**
UB_5	2	2	**3**
$ub_5(0, x_6)$	2	2	**3**
x_6			
d_6	0	0	0
CX_6	$\{(x_1, 0),$ $(x_5, 0)\}$	$\boldsymbol{\{(x_1, 1),}$ $\boldsymbol{(x_5, 0)\}}$	$\{(x_1, 1),$ $(x_5, 0)\}$
LB_6	2	**3**	3
UB_6	2	**3**	3

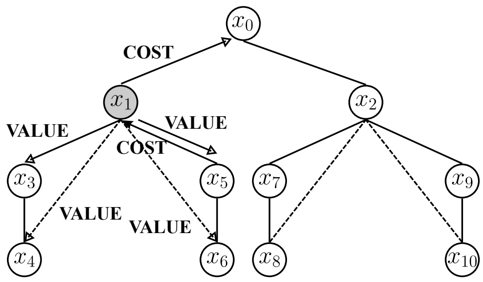

Figure A.14: Step 3 (6). Similar cost calculations and value changes are performed in x_1. After the cost calculation, x_1 sends VALUE messages with the current value $d_1 = 0$ to the lower neighbors and a COST message with $LB_1 = 7$ to x_0.

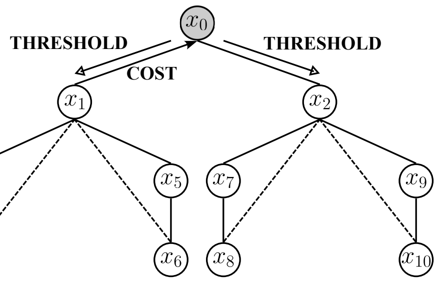

Figure A.15: Step 3 (7). x_0 receives the COST message and then increases LB_0, TH_0, and $th_0(0, x_1)$ to 7.

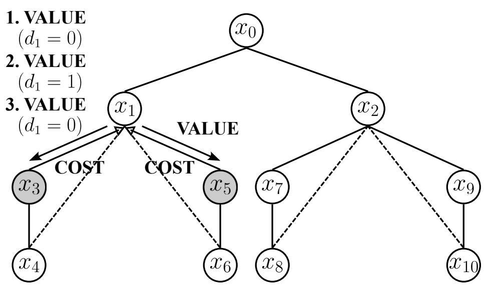

Figure A.16: Step 4. x_3 receives the three VALUE messages from x_1, in which the values are $d_1 = 0, d_1 = 1$, and $d_1 = 0$, in the order of sending. When x_3 processes the first VALUE message, the lower bound of x_3 is reinitialized as $LB_3 = LB_3(0) = lb_3(0, x_4) = 0$ since context $cx_3(0, x_4) \ni (x_1, 1)$, received from x_4 through the COST message, is incompatible with the updated context $CX_3 \ni (x_1, 0)$. After x_3 processes the remaining messages, x_3 sends two types of COST messages to x_1, i.e., the message with $LB_3 = 0$ and $CX_3 = \{(x_1, 0)\}$ and the message with $LB_3 = 0$ and $CX_3 = \{(x_1, 1)\}$. Similarly, x_5 receives the VALUE message with $d_1 = 0$ from x_1 and reinitializes the bounds. Additionally, x_5 sends a COST message with $LB_5 = 0$ and $CX_5 = \{(x_1, 0)\}$ to x_1.

Table A.6: States of agents in Steps 3 (6)–4.

Variables	Step 3 (6)	Step 3 (7)	Step 4
x_0			
d_0	0	0	0
LB_0	0	**7**	7
$lb_0(0, x_1)$	0	**7**	7
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	**7**	7
$ub_0(0, x_2)$	∞	∞	∞
TH_0	0	**7**	7
$th_0(0, x_1)$	0	**7**	7
$th_0(0, x_2)$	0	0	0
x_1			
d_1	**0**	0	0
LB_1	**7**	7	7
$lb_1(0, x_3)$	5	5	5
$lb_1(0, x_5)$	2	2	2
$lb_1(1, x_3)$	6	6	6
$lb_1(1, x_5)$	**3**	3	3
UB_1	7	7	7
$ub_1(0, x_3)$	5	5	5
$ub_1(0, x_5)$	2	2	2
$ub_1(1, x_3)$	6	6	6
$ub_1(1, x_5)$	**3**	3	3
TH_1	**7**	7	7
x_3			
d_3	0	0	0
CX_3	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\boldsymbol{\{(x_1, 0)\}}$
$cx_3(0, x_4)$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\boldsymbol{\{\}}$
LB_3	6	6	**0**
$lb_3(0, x_4)$	6	6	**0**
UB_3	6	6	$\boldsymbol{\infty}$
$ub_3(0, x_4)$	6	6	$\boldsymbol{\infty}$
x_4			
d_4	0	0	0
CX_4	$\{(x_1, 1), (x_3, 0)\}$	$\{(x_1, 1), (x_3, 0)\}$	$\{(x_1, 1), (x_3, 0)\}$
LB_4	6	6	6
UB_4	6	6	6
x_5			
d_5	0	0	0
CX_5	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\boldsymbol{\{(x_1, 0)\}}$
$cx_5(0, x_6)$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\boldsymbol{\{\}}$
LB_5	3	3	**0**
$lb_5(0, x_6)$	3	3	**0**
UB_5	3	3	$\boldsymbol{\infty}$
$ub_5(0, x_6)$	3	3	$\boldsymbol{\infty}$
x_6			
d_6	0	0	0
CX_6	$\{(x_1, 1), (x_5, 0)\}$	$\{(x_1, 1), (x_5, 0)\}$	$\{(x_1, 1), (x_5, 0)\}$
LB_6	3	3	3
UB_6	3	3	3

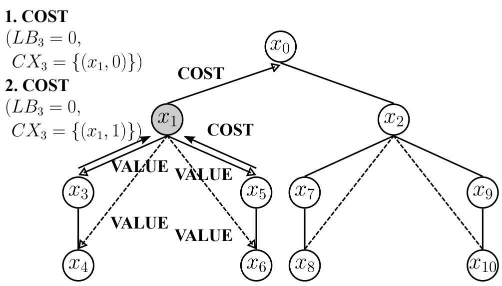

Figure A.17: Step 5 (1). x_1 receives the COST messages from x_3 and x_5 and updates the lower bounds as $LB_1(0) = lb_1(0, x_3)+lb_1(0, x_5) = 0$, $LB_1(1) = lb_1(1, x_3)+lb_1(1, x_5) = 3$. Thus, x_1 obtains the lower bound as $LB_1 = 0$ and keeps the value $d_1 = 0$. Additionally, x_1 sends a COST message with $LB_1 = 0$ to x_0.

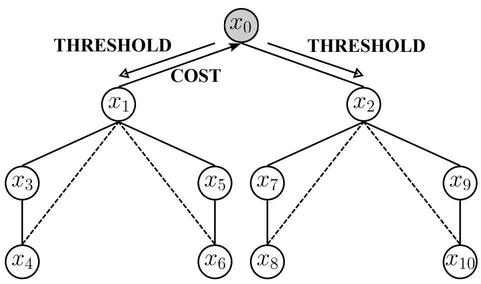

Figure A.18: Step 5 (2). x_0 receives the COST message from x_1 and updates the lower bound as $LB_0 = LB_0(0) = lb_0(0, x_1) + lb_0(0, x_2) = 0$. Although LB_0 decreases, the thresholds for the children are kept as $th_0(0, x_1) = 7$ and $th_0(0, x_2) = 0$.

Table A.7: States of agents in Steps 5 (1)–5 (2).

Variables	Step 5 (1)	Step 5 (2)
x_0		
d_0	0	0
LB_0	7	**0**
$lb_0(0, x_1)$	7	**0**
$lb_0(0, x_2)$	0	0
UB_0	∞	∞
$ub_0(0, x_1)$	7	$\boldsymbol{\infty}$
$ub_0(0, x_2)$	∞	∞
TH_0	7	7
$th_0(0, x_1)$	7	7
$th_0(0, x_2)$	0	0
x_1		
d_1	0	0
LB_1	**0**	0
$lb_1(0, x_3)$	**0**	0
$lb_1(0, x_5)$	**0**	0
$lb_1(1, x_3)$	**0**	0
$lb_1(1, x_5)$	3	3
UB_1	$\boldsymbol{\infty}$	∞
$ub_1(0, x_3)$	$\boldsymbol{\infty}$	∞
$ub_1(0, x_5)$	$\boldsymbol{\infty}$	∞
$ub_1(1, x_3)$	$\boldsymbol{\infty}$	∞
$ub_1(1, x_5)$	3	3
TH_1	7	7
x_3		
d_3	0	0
CX_3	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$
$cx_3(0, x_4)$	$\{\}$	$\{\}$
LB_3	0	0
$lb_3(0, x_4)$	0	0
UB_3	∞	∞
$ub_3(0, x_4)$	∞	∞
x_4		
d_4	0	0
CX_4	$\{(x_1, 1), (x_3, 0)\}$	$\{(x_1, 1), (x_3, 0)\}$
LB_4	6	6
UB_4	6	6
x_5		
d_5	0	0
CX_5	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$
$cx_5(0, x_6)$	$\{\}$	$\{\}$
LB_5	0	0
$lb_5(0, x_6)$	0	0
UB_5	∞	∞
$ub_5(0, x_6)$	∞	∞
x_6		
d_6	0	0
CX_6	$\{(x_1, 1), (x_5, 0)\}$	$\{(x_1, 1), (x_5, 0)\}$
LB_6	3	3
UB_6	3	3

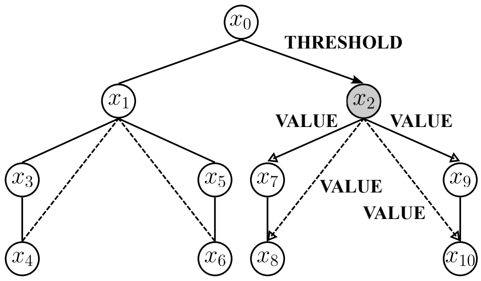

Figure A.19: Step 6 (1). The same process is performed in the subtree rooted at x_2.

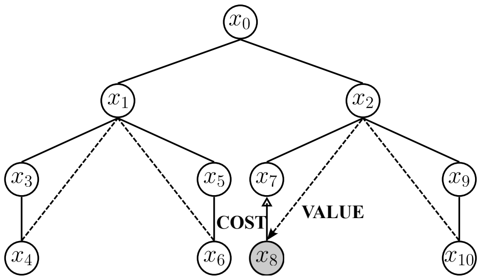

Figure A.20: Step 6 (2). The same process is performed in the subtree rooted at x_2.

Table A.8: States of agents in Steps 6 (1)–6 (2).

Variables	Step 6 (1)	Step 6 (2)
x_0		
d_0	0	0
LB_0	0	0
$lb_0(0, x_1)$	0	0
$lb_0(0, x_2)$	0	0
UB_0	∞	∞
$ub_0(0, x_1)$	∞	∞
$ub_0(0, x_2)$	∞	∞
TH_0	7	7
$th_0(0, x_1)$	7	7
$th_0(0, x_2)$	0	0
x_2		
d_2	0	0
LB_2	0	0
$lb_2(0, x_7)$	0	0
$lb_2(0, x_9)$	0	0
$lb_2(1, x_7)$	0	0
$lb_2(1, x_9)$	0	0
UB_2	∞	∞
$ub_2(0, x_7)$	∞	∞
$ub_2(0, x_9)$	∞	∞
$ub_2(1, x_7)$	∞	∞
$ub_2(1, x_9)$	∞	∞
TH_2	0	0
x_7		
d_7	0	0
CX_7	{}	{}
$cx_7(0, x_8)$	{}	{}
LB_7	0	0
$lb_7(0, x_8)$	0	0
UB_7	∞	∞
$ub_7(0, x_8)$	∞	∞
x_8		
d_8	0	0
CX_8	{}	$\boldsymbol{\{(x_1, 0), (x_7, 0)\}}$
LB_8	0	**5**
UB_8	∞	**5**
x_9		
d_9	0	0
CX_9	{}	{}
$cx_9(0, x_{10})$	{}	{}
LB_9	0	0
$lb_9(0, x_{10})$	0	0
UB_9	∞	∞
$ub_9(0, x_{10})$	∞	∞
x_{10}		
d_{10}	0	0
CX_{10}	{}	{}
LB_{10}	0	0
UB_{10}	∞	∞

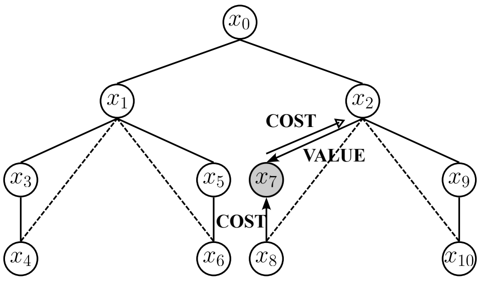

Figure A.21: Step 6 (3). The same process is performed in the subtree rooted at x_2.

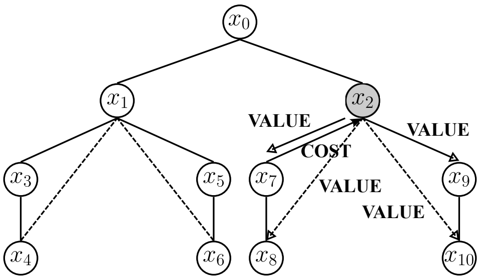

Figure A.22: Step 6 (4). The same process is performed in the subtree rooted at x_2.

Table A.9: States of agents in Steps 6 (3)–6 (4).

Variables	Step 6 (3)	Step 6 (4)
x_0		
d_0	0	0
LB_0	0	0
$lb_0(0, x_1)$	0	0
$lb_0(0, x_2)$	0	0
UB_0	∞	∞
$ub_0(0, x_1)$	∞	∞
$ub_0(0, x_2)$	∞	∞
TH_0	7	7
$th_0(0, x_1)$	7	7
$th_0(0, x_2)$	0	0
x_2		
d_2	0	**1**
LB_2	0	0
$lb_2(0, x_7)$	0	**5**
$lb_2(0, x_9)$	0	0
$lb_2(1, x_7)$	0	0
$lb_2(1, x_9)$	0	0
UB_2	∞	∞
$ub_2(0, x_7)$	∞	**5**
$ub_2(0, x_9)$	∞	∞
$ub_2(1, x_7)$	∞	∞
$ub_2(1, x_9)$	∞	∞
TH_2	0	0
x_7		
d_7	0	0
CX_7	$\boldsymbol{\{(x_1, 0)\}}$	$\{(x_1, 0)\}$
$cx_7(0, x_8)$	$\boldsymbol{\{(x_1, 0)\}}$	$\{(x_1, 0)\}$
LB_7	**5**	5
$lb_7(0, x_8)$	**5**	5
UB_7	**5**	5
$ub_7(0, x_8)$	**5**	5
x_8		
d_8	0	0
CX_8	$\{(x_1, 0),$ $(x_7, 0)\}$	$\{(x_1, 0),$ $(x_7, 0)\}$
LB_8	5	5
UB_8	5	5
x_9		
d_9	0	0
CX_9	$\{\}$	$\{\}$
$cx_9(0, x_{10})$	$\{\}$	$\{\}$
LB_9	0	0
$lb_9(0, x_{10})$	0	0
UB_9	∞	∞
$ub_9(0, x_{10})$	∞	∞
x_{10}		
d_{10}	0	0
CX_{10}	$\{\}$	$\{\}$
LB_{10}	0	0
UB_{10}	∞	∞

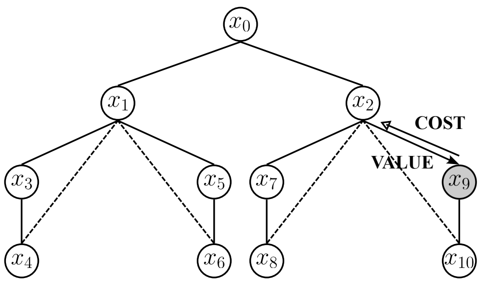

Figure A.23: Step 6 (5). The same process is performed in the subtree rooted at x_2.

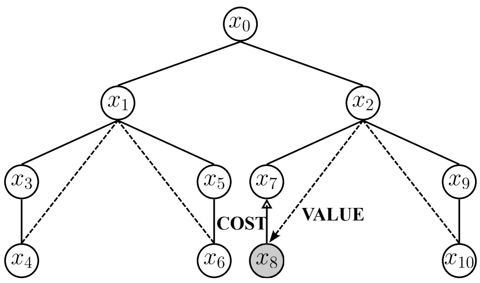

Figure A.24: Step 6 (6). The same process is performed in the subtree rooted at x_2.

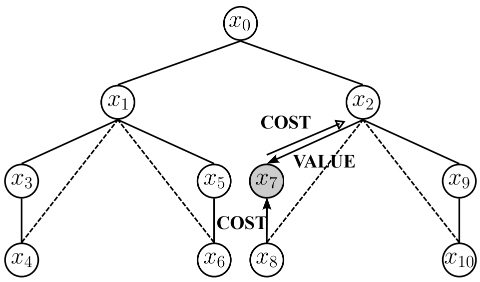

Figure A.25: Step 6 (7). The same process is performed in the subtree rooted at x_2.

Table A.10: States of agents in Steps 6 (5)–6 (7).

Variables	Step 6 (5)	Step 6 (6)	Step 6 (7)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	∞	∞
TH_0	7	7	7
$th_0(0, x_1)$	7	7	7
$th_0(0, x_2)$	0	0	0
x_2			
d_2	1	1	1
LB_2	0	0	0
$lb_2(0, x_7)$	5	5	5
$lb_2(0, x_9)$	0	0	0
$lb_2(1, x_7)$	0	0	0
$lb_2(1, x_9)$	0	0	0
UB_2	∞	∞	∞
$ub_2(0, x_7)$	5	5	5
$ub_2(0, x_9)$	∞	∞	∞
$ub_2(1, x_7)$	∞	∞	∞
$ub_2(1, x_9)$	∞	∞	∞
TH_2	0	0	0
x_7			
d_7	0	0	0
CX_7	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
$cx_7(0, x_8)$	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
LB_7	5	5	**6**
$lb_7(0, x_8)$	5	5	**6**
UB_7	5	5	**6**
$ub_7(0, x_8)$	5	5	**6**
x_8			
d_8	0	0	0
CX_8	$\{(x_1, 0), (x_7, 0)\}$	$\boldsymbol{\{(x_1, 1), (x_7, 0)\}}$	$\{(x_1, 1), (x_7, 0)\}$
LB_8	5	**6**	6
UB_8	5	**6**	6
x_9			
d_9	0	0	0
CX_9	{}	{}	{}
$cx_9(0, x_{10})$	{}	{}	{}
LB_9	0	0	0
$lb_9(0, x_{10})$	0	0	0
UB_9	∞	∞	∞
$ub_9(0, x_{10})$	∞	∞	∞
x_{10}			
d_{10}	0	0	0
CX_{10}	{}	{}	{}
LB_{10}	0	0	0
UB_{10}	∞	∞	∞

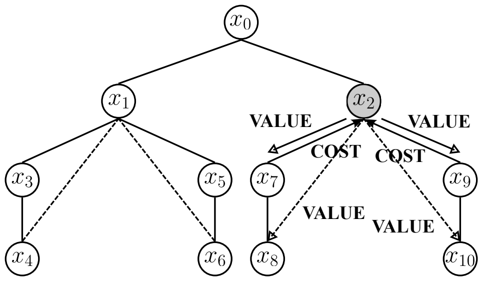

Figure A.26: Step 6 (8). The same process is performed in the subtree rooted at x_2.

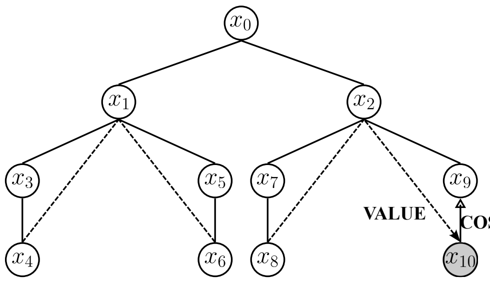

Figure A.27: Step 6 (9). The same process is performed in the subtree rooted at x_2.

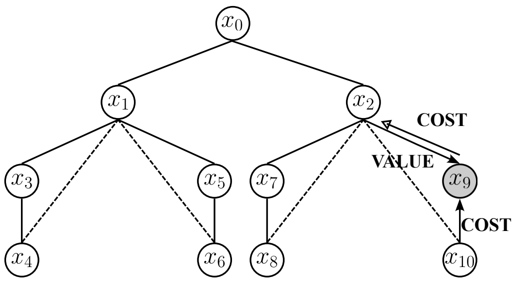

Figure A.28: Step 6 (10). The same process is performed in the subtree rooted at x_2.

Table A.11: States of agents in Steps 6 (8)–6 (10).

Variables	Step 6 (8)	Step 6 (9)	Step 6 (10)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	∞	∞
TH_0	7	7	7
$th_0(0, x_1)$	7	7	7
$th_0(0, x_2)$	0	0	0
x_2			
d_2	**0**	0	0
LB_2	**5**	5	5
$lb_2(0, x_7)$	5	5	5
$lb_2(0, x_9)$	0	0	0
$lb_2(1, x_7)$	**6**	6	6
$lb_2(1, x_9)$	0	0	0
UB_2	∞	∞	∞
$ub_2(0, x_7)$	5	5	5
$ub_2(0, x_9)$	∞	∞	∞
$ub_2(1, x_7)$	**6**	6	6
$ub_2(1, x_9)$	∞	∞	∞
TH_2	**5**	5	5
x_7			
d_7	0	0	0
CX_7	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
$cx_7(0, x_8)$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
LB_7	6	6	6
$lb_7(0, x_8)$	6	6	6
UB_7	6	6	6
$ub_7(0, x_8)$	6	6	6
x_8			
d_8	0	0	0
CX_8	$\{(x_1, 1), (x_7, 0)\}$	$\{(x_1, 1), (x_7, 0)\}$	$\{(x_1, 1), (x_7, 0)\}$
LB_8	6	6	6
UB_8	6	6	6
x_9			
d_9	0	0	0
CX_9	$\{\}$	$\{\}$	$\boldsymbol{\{(x_2, 0)\}}$
$cx_9(0, x_{10})$	$\{\}$	$\{\}$	$\boldsymbol{\{(x_2, 0)\}}$
LB_9	0	0	**2**
$lb_9(0, x_{10})$	0	0	**2**
UB_9	∞	∞	**2**
$ub_9(0, x_{10})$	∞	∞	**2**
x_{10}			
d_{10}	0	0	0
CX_{10}	$\{\}$	$\boldsymbol{\{(x_1, 0), (x_9, 0)\}}$	$\{(x_1, 0), (x_9, 0)\}$
LB_{10}	0	**2**	2
UB_{10}	∞	**2**	2

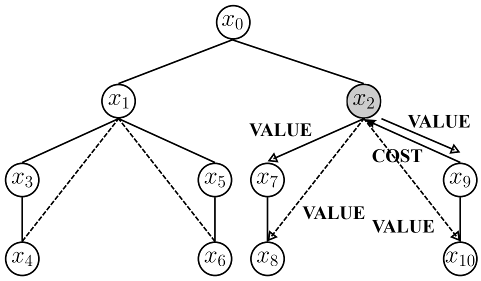

Figure A.29: Step 6 (11). The same process is performed in the subtree rooted at x_2.

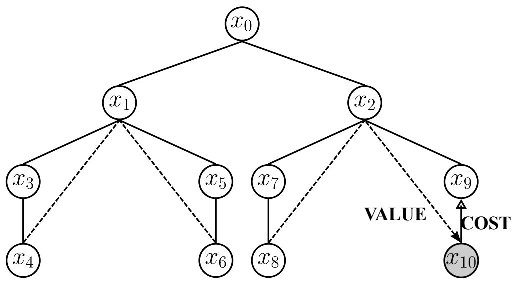

Figure A.30: Step 6 (12). The same process is performed in the subtree rooted at x_2.

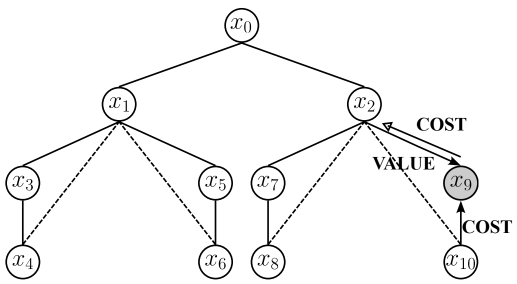

Figure A.31: Step 6 (13). The same process is performed in the subtree rooted at x_2.

Table A.12: States of agents in Steps 6 (11)–6 (13).

Variables	Step 6 (11)	Step 6 (12)	Step 6 (13)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	∞	∞
TH_0	7	7	7
$th_0(0, x_1)$	7	7	7
$th_0(0, x_2)$	0	0	0
x_2			
d_2	**1**	1	1
LB_2	**6**	6	6
$lb_2(0, x_7)$	5	5	5
$lb_2(0, x_9)$	**2**	2	2
$lb_2(1, x_7)$	6	6	6
$lb_2(1, x_9)$	0	0	0
UB_2	**7**	7	7
$ub_2(0, x_7)$	5	5	5
$ub_2(0, x_9)$	**2**	2	2
$ub_2(1, x_7)$	6	6	6
$ub_2(1, x_9)$	∞	∞	∞
TH_2	**6**	6	6
x_7			
d_7	0	0	0
CX_7	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
$cx_7(0, x_8)$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$
LB_7	6	6	6
$lb_7(0, x_8)$	6	6	6
UB_7	6	6	6
$ub_7(0, x_8)$	6	6	6
x_8			
d_8	0	0	0
CX_8	$\{(x_1, 1), (x_7, 0)\}$	$\{(x_1, 1), (x_7, 0)\}$	$\{(x_1, 1), (x_7, 0)\}$
LB_8	6	6	6
UB_8	6	6	6
x_9			
d_9	0	0	0
CX_9	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
$cx_9(0, x_{10})$	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$	$\boldsymbol{\{(x_1, 1)\}}$
LB_9	2	2	**3**
$lb_9(0, x_{10})$	2	2	**3**
UB_9	2	2	**3**
$ub_9(0, x_{10})$	2	2	**3**
x_{10}			
d_{10}	0	0	0
CX_{10}	$\{(x_1, 0), (x_9, 0)\}$	$\boldsymbol{\{(x_1, 1), (x_9, 0)\}}$	$\{(x_1, 1), (x_9, 0)\}$
LB_{10}	2	**3**	3
UB_{10}	2	**3**	3

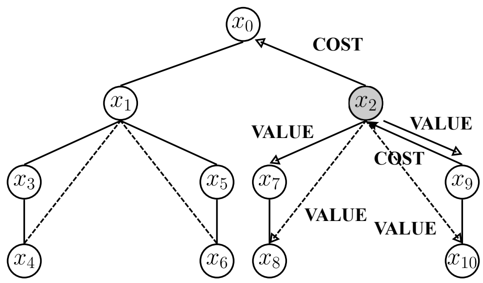

Figure A.32: Step 6 (14). The same process is performed in the subtree rooted at x_2.

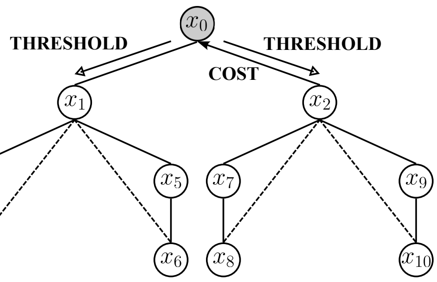

Figure A.33: Step 6 (15). The same process is performed in the subtree rooted at x_2.

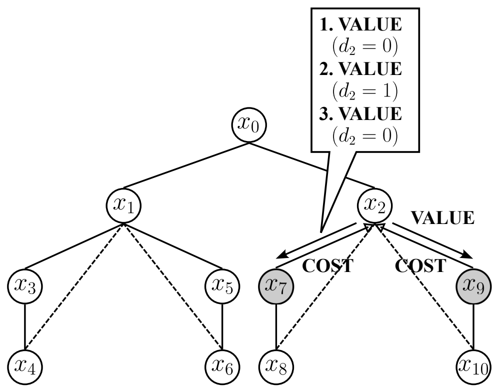

Figure A.34: Step 6 (16). The same process is performed in the subtree rooted at x_2.

Table A.13: States of agents in Steps 6 (14)–6 (16).

Variables	Step 6 (14)	Step 6 (15)	Step 6 (16)
x_0			
d_0	0	0	0
LB_0	0	**7**	7
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_2)$	0	**7**	7
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_2)$	∞	**7**	7
TH_0	7	7	7
$th_0(0, x_1)$	7	0	0
$th_0(0, x_2)$	0	**7**	7
x_2			
d_2	**0**	0	0
LB_2	**7**	7	7
$lb_2(0, x_7)$	5	5	5
$lb_2(0, x_9)$	2	2	2
$lb_2(1, x_7)$	6	6	6
$lb_2(1, x_9)$	**3**	3	3
UB_2	7	7	7
$ub_2(0, x_7)$	5	5	5
$ub_2(0, x_9)$	2	2	2
$ub_2(1, x_7)$	6	6	6
$ub_2(1, x_9)$	**3**	3	3
TH_2	**7**	7	7
x_7			
d_7	0	0	0
CX_7	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\boldsymbol{\{(x_1, 0)\}}$
$cx_7(0, x_8)$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	**{}**
LB_7	6	6	**0**
$lb_7(0, x_8)$	6	6	**0**
UB_7	6	6	$\boldsymbol{\infty}$
$ub_7(0, x_8)$	6	6	$\boldsymbol{\infty}$
x_8			
d_8	0	0	0
CX_8	$\{(x_1, 1), (x_7, 0)\}$	$\{(x_1, 1), (x_7, 0)\}$	$\{(x_1, 1), (x_7, 0)\}$
LB_8	6	6	6
UB_8	6	6	6
x_9			
d_9	0	0	0
CX_9	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	$\boldsymbol{\{(x_1, 0)\}}$
$cx_9(0, x_{10})$	$\{(x_1, 1)\}$	$\{(x_1, 1)\}$	**{}**
LB_9	3	3	**0**
$lb_9(0, x_{10})$	3	3	**0**
UB_9	3	3	$\boldsymbol{\infty}$
$ub_9(0, x_{10})$	3	3	$\boldsymbol{\infty}$
x_{10}			
d_{10}	0	0	0
CX_{10}	$\{(x_1, 1), (x_9, 0)\}$	$\{(x_1, 1), (x_9, 0)\}$	$\{(x_1, 1), (x_9, 0)\}$
LB_{10}	3	3	3
UB_{10}	3	3	3

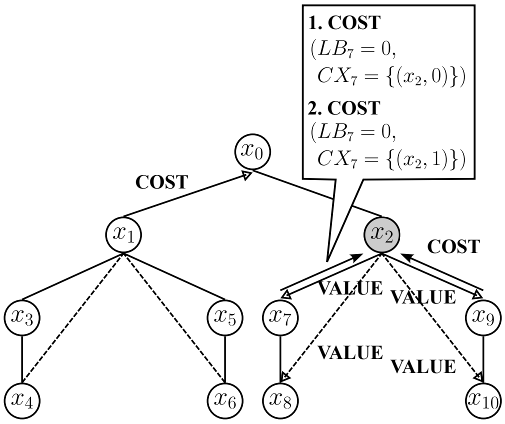

Figure A.35: Step 6 (17). The same process is performed in the subtree rooted at x_2.

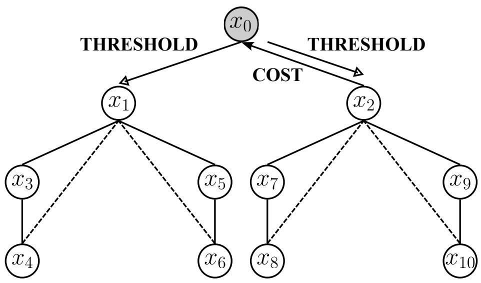

Figure A.36: Step 6 (18). The same process is performed in the subtree rooted at x_2.

Table A.14: States of agents in Steps 6 (17)–6 (18).

Variables	Step 6 (17)	Step 6 (18)
x_0		
d_0	0	0
LB_0	7	**0**
$lb_0(0, x_1)$	0	0
$lb_0(0, x_2)$	7	**0**
UB_0	∞	∞
$ub_0(0, x_1)$	∞	∞
$ub_0(0, x_2)$	7	$\boldsymbol{\infty}$
TH_0	7	7
$th_0(0, x_1)$	0	0
$th_0(0, x_2)$	7	7
x_2		
d_2	0	0
LB_2	**0**	0
$lb_2(0, x_7)$	**0**	0
$lb_2(0, x_9)$	**0**	0
$lb_2(1, x_7)$	**0**	0
$lb_2(1, x_9)$	3	3
UB_2	$\boldsymbol{\infty}$	∞
$ub_2(0, x_7)$	$\boldsymbol{\infty}$	∞
$ub_2(0, x_9)$	$\boldsymbol{\infty}$	∞
$ub_2(1, x_7)$	$\boldsymbol{\infty}$	∞
$ub_2(1, x_9)$	3	3
TH_2	7	7
x_7		
d_7	0	0
CX_7	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$
$cx_7(0, x_8)$	$\{\}$	$\{\}$
LB_7	0	0
$lb_7(0, x_8)$	0	0
UB_7	∞	∞
$ub_7(0, x_8)$	∞	∞
x_8		
d_8	0	0
CX_8	$\{(x_1, 1),$ $(x_7, 0)\}$	$\{(x_1, 1),$ $(x_7, 0)\}$
LB_8	6	6
UB_8	6	6
x_9		
d_9	0	0
CX_9	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$
$cx_9(0, x_{10})$	$\{\}$	$\{\}$
LB_9	0	0
$lb_9(0, x_{10})$	0	0
UB_9	∞	∞
$ub_9(0, x_{10})$	∞	∞
x_{10}		
d_{10}	0	0
CX_{10}	$\{(x_1, 1),$ $(x_9, 0)\}$	$\{(x_1, 1),$ $(x_9, 0)\}$
LB_{10}	3	3
UB_{10}	3	3

Appendix A.2. Counterexample to Optimality Caused by Initialization

Figures A.37–A.40 and Tables A.15–A.16 show the trace of the counterexample to optimality caused by initialization, described in 3.2.

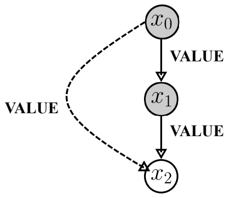

Figure A.37: Step 1 (1). Agents send VALUE messages to their children and pseudo-children.

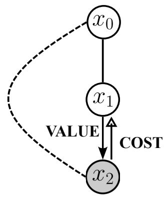

Figure A.38: Step 1 (2). x_2 receives the VALUE message only from its parent x_1, which means that the messages from the pseudo-parent x_0 are delayed. Here, x_2 updates the current context as $CX_2 = \{(x_1, 0)\}$. Since local cost $\delta_i(d, CX)$ is defined as $\delta_i(d, CX) := \sum_{(x_j,d_j)\in CX} f_{i,j}(d, d_j)$, x_2 computes the local costs as $\delta_2(0, \{(x_1, 0)\}) = f_{1,2}(0, 0) = 0$ and $\delta_2(1, \{(x_1, 0)\}) = f_{1,2}(0, 1) = 0$. Thus, the bounds of x_2 are obtained as $LB_2 = UB_2 = 0$. Then, x_2 sends a COST message with $LB_2 = UB_2 = 0$ to x_1.

Table A.15: States of agents in Steps 0–1 (2).

Variables	Step 0	Step 1 (1)	Step 1 (2)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(1, x_1)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(1, x_1)$	∞	∞	∞
TH_0	0	0	0
x_1			
d_1	0	0	0
CX_1	{}	{}	{}
$cx_1(0, x_2)$	{}	{}	{}
LB_1	0	0	0
$lb_1(0, x_2)$	0	0	0
UB_1	∞	∞	∞
$ub_1(0, x_2)$	∞	∞	∞
TH_1	0	0	0
x_2			
d_2	0	0	0
CX_2	{}	{}	$\mathbf{\{(x_1, 0)\}}$
LB_2	0	0	0
UB_2	∞	∞	**0**

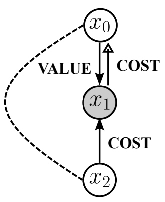

Figure A.39: Step 2 (1). After x_1 receives the VALUE message from x_0 and the COST message from x_2, x_1 computes the bounds as $LB_1 = UB_1 = 0$.

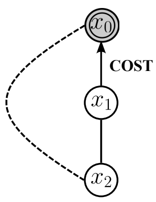

Figure A.40: Step 2 (2). x_0 updates the bounds as $LB_0(0) = UB_0(0) = 0$ through the COST message sent from x_1. Since $LB_0 = TH_0 = UB_0 = UB_0(0) = 0$ due to ThresholdInvariant, x_0 keeps its value as $d_0 = 0$ and satisfies the termination condition. However, the variable value $d_0 = 0$ is suboptimal.

Table A.16: States of agents in Steps 2 (1)–2 (2).

Variables	Step 2 (1)	Step 2 (2)
x_0		
d_0	0	0
LB_0	0	0
$lb_0(0, x_1)$	0	0
$lb_0(1, x_1)$	0	0
UB_0	∞	**0**
$ub_0(0, x_1)$	∞	**0**
$ub_0(1, x_1)$	∞	∞
TH_0	0	0
x_1		
d_1	0	0
CX_1	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
LB_1	0	0
$lb_1(0, x_2)$	0	0
UB_1	**0**	0
$ub_1(0, x_2)$	**0**	0
TH_1	0	0
x_2		
d_2	0	0
CX_2	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$
LB_2	0	0
UB_2	0	0

Appendix A.3. Counterexample to Optimality Caused by TERMINATE Messages

Figures A.41–A.56 and Tables A.17–A.23 show the trace of the counterexample to optimality caused by TERMINATE messages, described in 3.3. Additionally, the trace where x_2 performs reinitialization when it receives a TERMINATE message, described in "Cause of Counterexample", is shown as Step 6' in Figures A.57–A.59 and Tables A.24 and A.25.

Table A.17: States of agents in Steps 0 (1)–0 (2).

Variables	Step 0 (1)	Step 0 (2)
x_0		
d_0	0	0
LB_0	0	0
$lb_0(0, x_1)$	0	0
$lb_0(0, x_4)$	0	0
$lb_0(1, x_1)$	0	0
$lb_0(1, x_4)$	0	0
UB_0	∞	∞
$ub_0(0, x_1)$	∞	∞
$ub_0(0, x_4)$	∞	∞
$ub_0(1, x_1)$	∞	∞
$ub_0(1, x_4)$	∞	∞
TH_0	0	0
$th_0(0, x_1)$	0	0
$th_0(0, x_4)$	0	0
$th_0(1, x_1)$	0	0
$th_0(1, x_4)$	0	0
x_1		
d_1	0	0
CX_1	{}	{}
$cx_1(0, x_2)$	{}	{}
LB_1	0	0
$lb_1(0, x_2)$	0	0
UB_1	∞	∞
$ub_1(0, x_2)$	∞	∞
TH_1	0	0
x_2		
d_2	0	0
CX_2	{}	{}
$cx_2(0, x_3)$	{}	{}
$cx_2(1, x_3)$	{}	{}
LB_2	0	0
$lb_2(0, x_3)$	0	0
$lb_2(1, x_3)$	0	0
UB_2	∞	∞
$ub_2(0, x_3)$	∞	∞
$ub_2(1, x_3)$	∞	∞
TH_2	0	0
x_3		
d_3	0	0
CX_3	{}	$\{(x_0, 0), (x_2, 0)\}$
LB_3	0	**1**
UB_3	∞	**1**
x_4		
d_4	0	0
CX_4	{}	{}
LB_4	0	0
UB_4	∞	∞

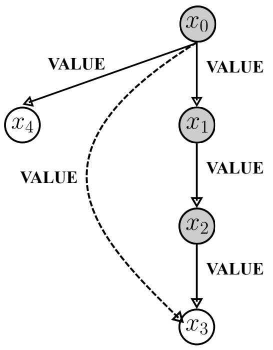

Figure A.41: Step 0 (1). Agents calculate the cost in the case where x_0 takes the value 0.

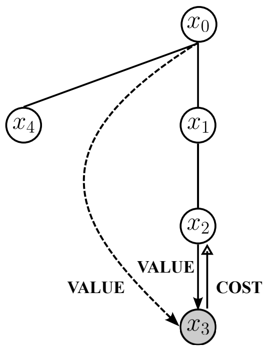

Figure A.42: Step 0 (2). Agents calculate the cost in the case where x_0 takes the value 0.

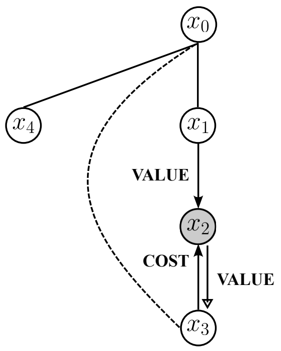

Figure A.43: Step 0 (3). Agents calculate the cost in the case where x_0 takes the value 0.

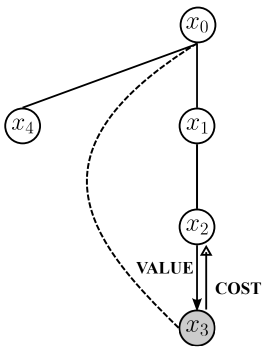

Figure A.44: Step 0 (4). Agents calculate the cost in the case where x_0 takes the value 0.

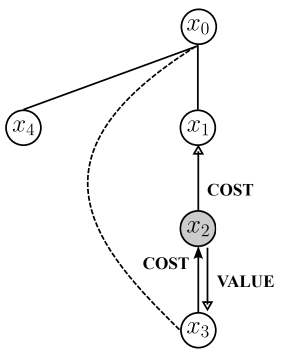

Figure A.45: Step 0 (5). Agents calculate the cost in the case where x_0 takes the value 0.

Table A.18: States of agents in Steps 0 (3)–0 (5).

Variables	Step 0 (3)	Step 0 (4)	Step 0 (5)
x_0			
d_0	0	0	0
LB_0	0	0	0
$lb_0(0, x_1)$	0	0	0
$lb_0(0, x_4)$	0	0	0
$lb_0(1, x_1)$	0	0	0
$lb_0(1, x_4)$	0	0	0
UB_0	∞	∞	∞
$ub_0(0, x_1)$	∞	∞	∞
$ub_0(0, x_4)$	∞	∞	∞
$ub_0(1, x_1)$	∞	∞	∞
$ub_0(1, x_4)$	∞	∞	∞
TH_0	0	0	0
$th_0(0, x_1)$	0	0	0
$th_0(0, x_4)$	0	0	0
$th_0(1, x_1)$	0	0	0
$th_0(1, x_4)$	0	0	0
x_1			
d_1	0	0	0
CX_1	$\{\}$	$\{\}$	$\{\}$
$cx_1(0, x_2)$	$\{\}$	$\{\}$	$\{\}$
LB_1	0	0	0
$lb_1(0, x_2)$	0	0	0
UB_1	∞	∞	∞
$ub_1(0, x_2)$	∞	∞	∞
TH_1	0	0	0
x_2			
d_2	**1**	1	**0**
CX_2	$\boldsymbol{\{(x_0, 0), (x_1, 0)\}}$	$\{(x_0, 0), (x_1, 0)\}$	$\{(x_0, 0), (x_1, 0)\}$
$cx_2(0, x_3)$	$\boldsymbol{\{(x_0, 0)\}}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_2(1, x_3)$	$\{\}$	$\{\}$	$\boldsymbol{\{(x_0, 0)\}}$
LB_2	0	0	**1**
$lb_2(0, x_3)$	**1**	1	1
$lb_2(1, x_3)$	0	0	**11**
UB_2	**1**	1	1
$ub_2(0, x_3)$	**1**	1	1
$ub_2(1, x_3)$	∞	∞	**11**
TH_2	0	0	**1**
x_3			
d_3	0	0	0
CX_3	$\{(x_0, 0), (x_2, 0)\}$	$\boldsymbol{\{(x_0, 0), (x_2, 1)\}}$	$\{(x_0, 0), (x_2, 1)\}$
LB_3	1	**11**	11
UB_3	1	**11**	11
x_4			
d_4	0	0	0
CX_4	$\{\}$	$\{\}$	$\{\}$
LB_4	0	0	0
UB_4	∞	∞	∞

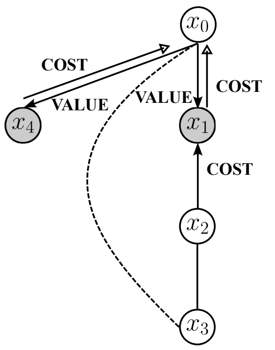

Figure A.46: Step 0 (6). Agents calculate the cost in the case where x_0 takes the value 0.

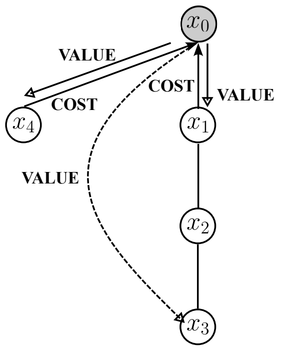

Figure A.47: Step 1 (1). x_0 sends VALUE messages with $d_0 = 1$ to x_1, x_3, and x_4.

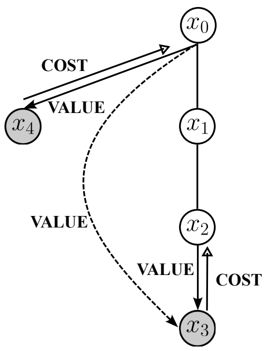

Figure A.48: Step 1 (2). x_3 and x_4 receive the VALUE messages. At this moment, x_3 updates the context and the bounds as $CX_3 = \{(x_0, 1), (x_2, 0)\}$ and $LB_3 = UB_3 = 200$, and x_4 also updates them as $CX_4 = \{(x_0, 1)\}$ and $LB_4 = UB_4 = 1000$. Then, they send COST messages to their parents: from x_3 to x_2 and from x_4 to x_0.

Table A.19: States of agents in Steps 0 (6)–1 (2).

Variables	Step 0 (6)	Step 1 (1)	Step 1 (2)
x_0			
d_0	0	**1**	1
LB_0	0	0	0
$lb_0(0, x_1)$	0	**1**	1
$lb_0(0, x_4)$	0	0	0
$lb_0(1, x_1)$	0	0	0
$lb_0(1, x_4)$	0	0	0
UB_0	∞	**1**	1
$ub_0(0, x_1)$	∞	**1**	1
$ub_0(0, x_4)$	∞	**0**	0
$ub_0(1, x_1)$	∞	∞	∞
$ub_0(1, x_4)$	∞	∞	∞
TH_0	0	0	0
$th_0(0, x_1)$	0	**1**	1
$th_0(0, x_4)$	0	0	0
$th_0(1, x_1)$	0	0	0
$th_0(1, x_4)$	0	0	0
x_1			
d_1	0	0	0
CX_1	$\boldsymbol{\{(x_0, 0)\}}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\boldsymbol{\{(x_0, 0)\}}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
LB_1	**1**	1	1
$lb_1(0, x_2)$	**1**	1	1
UB_1	**1**	1	1
$ub_1(0, x_2)$	**1**	1	1
TH_1	**1**	1	1
x_2			
d_2	0	0	0
CX_2	$\{(x_0, 0), (x_1, 0)\}$	$\{(x_0, 0), (x_1, 0)\}$	$\{(x_0, 0), (x_1, 0)\}$
$cx_2(0, x_3)$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_2(1, x_3)$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
LB_2	1	1	1
$lb_2(0, x_3)$	1	1	1
$lb_2(1, x_3)$	11	11	11
UB_2	1	1	1
$ub_2(0, x_3)$	1	1	1
$ub_2(1, x_3)$	11	11	11
TH_2	1	1	1
x_3			
d_3	0	0	**1**
CX_3	$\{(x_0, 0), (x_2, 1)\}$	$\{(x_0, 0), (x_2, 1)\}$	$\boldsymbol{\{(x_0, 1), (x_2, 0)\}}$
LB_3	11	11	**200**
UB_3	11	11	**200**
x_4			
d_4	0	0	0
CX_4	$\boldsymbol{\{(x_0, 0)\}}$	$\{(x_0, 0)\}$	$\boldsymbol{\{(x_0, 1)\}}$
LB_4	0	0	**1000**
UB_4	**0**	0	**1000**

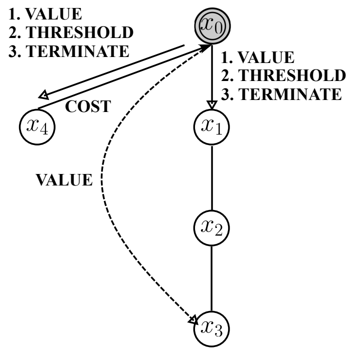

Figure A.49: Step 1 (3). After x_0 receives the COST message from x_4, x_0 changes its value d_0 back to 0 since the bounds are obtained as $LB_0(1) = 1000$ and $LB_0 = TH_0 = UB_0 = UB_0(0) = 1$ due to ThresholdInvariant. Therefore, x_0 satisfies the termination condition. When x_0 executes termination, x_0 sends VALUE messages with $d_0 = 0$ to its lower neighbors (i.e., x_1, x_3, and x_4) and THRESHOLD and TERMINATE messages to its children (i.e., x_1 and x_4) in this order.

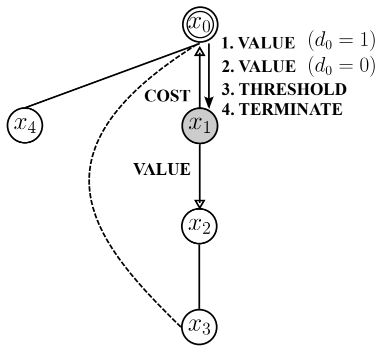

Figure A.50: Step 2. x_1 receives the VALUE messages from x_0, including the message that x_0 sent when $d_0 = 1$, in the order of sending. Then, x_1 changes CX_1 from $\{(x_0, 0)\}$ into $\{(x_0, 1)\}$ and returns it to $\{(x_0, 0)\}$. Since $\{(x_0, 1)\}$ is incompatible with $cx_1(0, x_2) = \{(x_0, 0)\}$, x_1 reinitializes the bounds as $LB_1 = 0$ and $UB_1 = \infty$. By contrast, TH_1 is not changed from 1. Furthermore, x_1 receives the THRESHOLD and TERMINATE messages from x_0, and then x_1 records receiving the TERMINATE message but does not terminate because $TH_1 = 1 < UB_1 = \infty$. Additionally, x_1 sends a VALUE message to x_2.

Table A.20: States of agents in Steps 1 (3)–3.

Variables	Step 1 (3)	Step 2	Step 3
x_0			
d_0	**0**	0	0
LB_0	**1**	1	1
$lb_0(0, x_1)$	1	1	1
$lb_0(0, x_4)$	0	0	0
$lb_0(1, x_1)$	0	0	0
$lb_0(1, x_4)$	**1000**	1000	1000
UB_0	1	1	1
$ub_0(0, x_1)$	1	1	1
$ub_0(0, x_4)$	0	0	0
$ub_0(1, x_1)$	∞	∞	∞
$ub_0(1, x_4)$	**1000**	1000	1000
TH_0	**1**	1	1
$th_0(0, x_1)$	1	1	1
$th_0(0, x_4)$	0	0	0
$th_0(1, x_1)$	0	0	0
$th_0(1, x_4)$	**1000**	1000	1000
x_1			
d_1	0	0	0
CX_1	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\{(x_0, 0)\}$	**{}**	{}
LB_1	1	**0**	0
$lb_1(0, x_2)$	1	**0**	0
UB_1	1	$\boldsymbol{\infty}$	∞
$ub_1(0, x_2)$	1	$\boldsymbol{\infty}$	∞
TH_1	1	1	1
x_2			
d_2	0	0	0
CX_2	$\{(x_0, 0), (x_1, 0)\}$	$\{(x_0, 0), (x_1, 0)\}$	$\{(x_0, 0), (x_1, 0)\}$
$cx_2(0, x_3)$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_2(1, x_3)$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
LB_2	1	1	1
$lb_2(0, x_3)$	1	1	1
$lb_2(1, x_3)$	11	11	11
UB_2	1	1	1
$ub_2(0, x_3)$	1	1	1
$ub_2(1, x_3)$	11	11	11
TH_2	1	1	1
x_3			
d_3	1	1	1
CX_3	$\{(x_0, 1), (x_2, 0)\}$	$\{(x_0, 1), (x_2, 0)\}$	$\{(x_0, 1), (x_2, 0)\}$
LB_3	200	200	200
UB_3	200	200	200
x_4			
d_4	0	0	0
CX_4	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
LB_4	1000	1000	1000
UB_4	1000	1000	1000

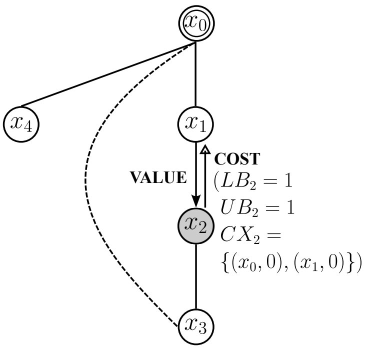

Figure A.51: Step 3. x_2 receives only the message from x_1, not from x_3, which means that the messages from x_3 are delayed. Then, x_2 sends a COST message to x_1 with $CX_2 = \{(x_0, 0), (x_1, 0)\}$ and $LB_2 = UB_2 = 1$, which are the same states as in Step 0.

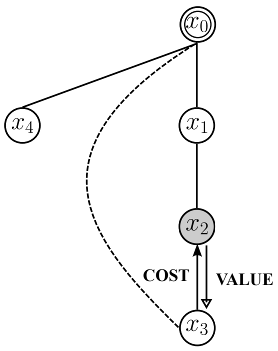

Figure A.52: Step 4 (1). x_2 receives the COST message from x_3 with $CX_3 = \{(x_0, 1), (x_2, 0)\}$ and $LB_3 = UB_3 = 200$. Since x_2 is not a neighbor of x_0, x_2 updates CX_2 from $\{(x_0, 0), (x_1, 0)\}$ to $\{(x_0, 1), (x_1, 0)\}$. Then, the bounds of x_2 are reinitialized and updated by the bounds in the message: $LB_2(0) = UB_2(0) = 200, LB_2(1) = 0$, and $UB_2(1) = \infty$. x_2 also changes its value d_2 to 1 because $LB_2(0) > TH_2 = 1$, and sends a VALUE message to x_3.

Table A.21: States of agents in Steps 3–4 (1).

Variables	Step 3	Step 4 (1)
x_0		
d_0	0	0
LB_0	1	1
$lb_0(0, x_1)$	1	1
$lb_0(0, x_4)$	0	0
$lb_0(1, x_1)$	0	0
$lb_0(1, x_4)$	1000	1000
UB_0	1	1
$ub_0(0, x_1)$	1	1
$ub_0(0, x_4)$	0	0
$ub_0(1, x_1)$	∞	∞
$ub_0(1, x_4)$	1000	1000
TH_0	1	1
$th_0(0, x_1)$	1	1
$th_0(0, x_4)$	0	0
$th_0(1, x_1)$	0	0
$th_0(1, x_4)$	1000	1000
x_1		
d_1	0	0
CX_1	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\{\}$	$\{\}$
LB_1	0	0
$lb_1(0, x_2)$	0	0
UB_1	∞	∞
$ub_1(0, x_2)$	∞	∞
TH_1	1	1
x_2		
d_2	0	**1**
CX_2	$\{(x_0, 0), (x_1, 0)\}$	$\boldsymbol{\{(x_0, 1), (x_1, 0)\}}$
$cx_2(0, x_3)$	$\{(x_0, 0)\}$	$\boldsymbol{\{(x_0, 1)\}}$
$cx_2(1, x_3)$	$\{(x_0, 0)\}$	$\boldsymbol{\{\}}$
LB_2	1	**0**
$lb_2(0, x_3)$	1	**200**
$lb_2(1, x_3)$	11	**0**
UB_2	1	**200**
$ub_2(0, x_3)$	1	**200**
$ub_2(1, x_3)$	11	$\boldsymbol{\infty}$
TH_2	1	1
x_3		
d_3	1	1
CX_3	$\{(x_0, 1), (x_2, 0)\}$	$\{(x_0, 1), (x_2, 0)\}$
LB_3	200	200
UB_3	200	200
x_4		
d_4	0	0
CX_4	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
LB_4	1000	1000
UB_4	1000	1000

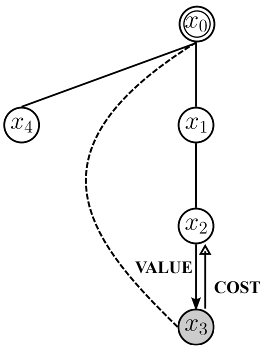

Figure A.53: Step 4 (2). After x_3 receives only the VALUE message from x_2 but not the messages from x_0, x_3 updates the current context and the bounds as $CX_3 = \{(x_0, 1), (x_2, 1)\}$ and $LB_3 = UB_3 = 100$. Next, x_3 sends a COST message to x_2 again.

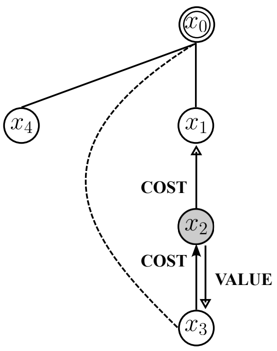

Figure A.54: Step 4 (3). x_2 receives it. Then, x_2 computes the bounds as $LB_2(1) = UB_2(1) = 100$ and updates the threshold as $TH_2 = 100$ because of ThresholdInvariant.

Table A.22: States of agents in Steps 4 (2)–4 (3).

Trace	Step 4 (2)	Step 4 (3)
x_0		
d_0	0	0
LB_0	1	1
$lb_0(0, x_1)$	1	1
$lb_0(0, x_4)$	0	0
$lb_0(1, x_1)$	0	0
$lb_0(1, x_4)$	1000	1000
UB_0	1	1
$ub_0(0, x_1)$	1	1
$ub_0(0, x_4)$	0	0
$ub_0(1, x_1)$	∞	∞
$ub_0(1, x_4)$	1000	1000
TH_0	1	1
$th_0(0, x_1)$	1	1
$th_0(0, x_4)$	0	0
$th_0(1, x_1)$	0	0
$th_0(1, x_4)$	1000	1000
x_1		
d_1	0	0
CX_1	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\{\}$	$\{\}$
LB_1	0	0
$lb_1(0, x_2)$	0	0
UB_1	∞	∞
$ub_1(0, x_2)$	∞	∞
TH_1	1	1
x_2		
d_2	1	1
CX_2	$\{(x_0, 1), (x_1, 0)\}$	$\{(x_0, 1), (x_1, 0)\}$
$cx_2(0, x_3)$	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
$cx_2(1, x_3)$	$\{\}$	$\mathbf{\{(x_0, 1)\}}$
LB_2	0	**100**
$lb_2(0, x_3)$	200	200
$lb_2(1, x_3)$	0	**100**
UB_2	200	**100**
$ub_2(0, x_3)$	200	200
$ub_2(1, x_3)$	∞	**100**
TH_2	1	**100**
x_3		
d_3	1	1
CX_3	$\mathbf{\{(x_0, 1), (x_2, 1)\}}$	$\{(x_0, 1), (x_2, 1)\}$
LB_3	**100**	100
UB_3	**100**	100
x_4		
d_4	0	0
CX_4	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
LB_4	1000	1000
UB_4	1000	1000

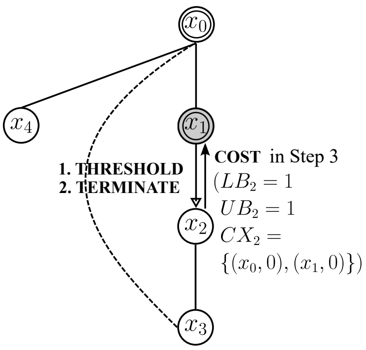

Figure A.55: Step 5. x_1 receives the COST message with $CX_2 = \{(x_0, 0), (x_1, 0)\}$ and $LB_2 = UB_2 = 1$, sent from x_2 in Step 3. Since this context is compatible with $CX_1 = \{(x_0, 0)\}$, x_1 updates the bounds as $LB_1 = UB_1 = 1$. At this moment, the termination condition is satisfied because $TH_1 = UB_1 = 1$. Thus, x_1 sends two messages to x_2: a THRESHOLD message with $th_1(0, x_2) = 1$ and $CX_1 = \{(x_0, 0)\}$ and a TERMINATE message with $CX_1 \cup \{(x_1, 0)\} = \{(x_0, 0), (x_1, 0)\}$. Then, x_1 terminates.

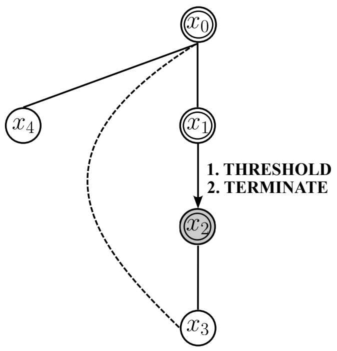

Figure A.56: Step 6. When x_2 receives the THRESHOLD message from x_1, x_2 does not update TH_2 since context $\{(x_0, 0)\}$ in the message is incompatible with $CX_2 = \{(x_0, 1), (x_1, 0)\}$, and therefore retains its threshold as $TH_2 = 100$. Next, x_2 receives the TERMINATE message from x_1. Although CX_2 is changed to $\{(x_0, 0), (x_1, 0)\}$, the bounds of x_2 are not changed since reinitialization is not performed when an agent receives a TERMINATE message. Therefore, x_2 terminates with the suboptimal value $d_2 = 1$ because x_2 has already satisfied the termination condition with $UB_2 = UB_2(1) = TH_2 = 100$ and received the TERMINATE message.

Table A.23: States of agents in Steps 5–6.

Variables	Step 5	Step 6
x_0		
d_0	0	0
LB_0	1	1
$lb_0(0, x_1)$	1	1
$lb_0(0, x_4)$	0	0
$lb_0(1, x_1)$	0	0
$lb_0(1, x_4)$	1000	1000
UB_0	1	1
$ub_0(0, x_1)$	1	1
$ub_0(0, x_4)$	0	0
$ub_0(1, x_1)$	∞	∞
$ub_0(1, x_4)$	1000	1000
TH_0	1	1
$th_0(0, x_1)$	1	1
$th_0(0, x_4)$	0	0
$th_0(1, x_1)$	0	0
$th_0(1, x_4)$	1000	1000
x_1		
d_1	0	0
CX_1	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\boldsymbol{\{(x_0, 0)\}}$	$\{(x_0, 0)\}$
LB_1	**1**	1
$lb_1(0, x_2)$	**1**	1
UB_1	**1**	1
$ub_1(0, x_2)$	**1**	1
TH_1	1	1
x_2		
d_2	1	1
CX_2	$\{(x_0, 1), (x_1, 0)\}$	$\boldsymbol{\{(x_0, 0), (x_1, 0)\}}$
$cx_2(0, x_3)$	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
$cx_2(1, x_3)$	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
LB_2	100	100
$lb_2(0, x_3)$	200	200
$lb_2(1, x_3)$	100	100
UB_2	100	100
$ub_2(0, x_3)$	200	200
$ub_2(1, x_3)$	100	100
TH_2	100	100
x_3		
d_3	1	1
CX_3	$\{(x_0, 1), (x_2, 1)\}$	$\{(x_0, 1), (x_2, 1)\}$
LB_3	100	100
UB_3	100	100
x_4		
d_4	0	0
CX_4	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
LB_4	1000	1000
UB_4	1000	1000

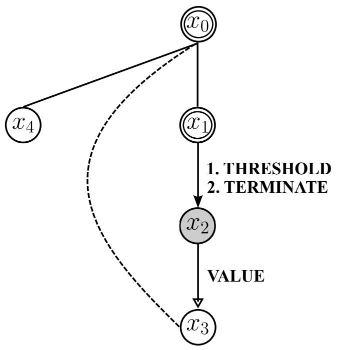

Figure A.57: Step 6' (1). The bounds of x_2 are reinitialized when x_2 receives the TERMINATE message from x_1 in Step 6. In this case, the bounds of x_2 are obtained as $LB_2(0) = LB_2(1) = 0$ and $UB_2(0) = UB_2(1) = \infty$; and the threshold of x_2 is obtained as $TH_2 = 100$.

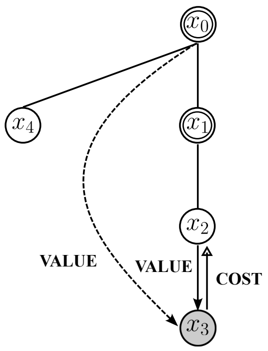

Figure A.58: Step 6' (2). After x_3 receives the VALUE message with the value $d_0 = 0$ from x_0, x_3 updates the current context as $CX_3 = \{(x_0, 0), (x_2, 1)\}$ and the bounds as $LB_3 = UB_3 = 11$. Then, x_3 sends a COST message to x_2.

Table A.24: States of agents in Steps 6' (1)–6' (2).

Variables	Step 6' (1)	Step 6' (2)
x_0		
d_0	0	0
LB_0	1	1
$lb_0(0, x_1)$	1	1
$lb_0(0, x_4)$	0	0
$lb_0(1, x_1)$	0	0
$lb_0(1, x_4)$	1000	1000
UB_0	1	1
$ub_0(0, x_1)$	1	1
$ub_0(0, x_4)$	0	0
$ub_0(1, x_1)$	∞	∞
$ub_0(1, x_4)$	1000	1000
TH_0	1	1
$th_0(0, x_1)$	1	1
$th_0(0, x_4)$	0	0
$th_0(1, x_1)$	0	0
$th_0(1, x_4)$	1000	1000
x_1		
d_1	0	0
CX_1	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\{(x_0, 0)\}$	$\{(x_0, 0)\}$
LB_1	1	1
$lb_1(0, x_2)$	1	1
UB_1	1	1
$ub_1(0, x_2)$	1	1
TH_1	1	1
x_2		
d_2	1	1
CX_2	$\boldsymbol{\{(x_0, 0), (x_1, 0)\}}$	$\{(x_0, 0), (x_1, 0)\}$
$cx_2(0, x_3)$	**{}**	{}
$cx_2(1, x_3)$	**{}**	{}
LB_2	**0**	0
$lb_2(0, x_3)$	**0**	0
$lb_2(1, x_3)$	**0**	0
UB_2	$\boldsymbol{\infty}$	∞
$ub_2(0, x_3)$	$\boldsymbol{\infty}$	∞
$ub_2(1, x_3)$	$\boldsymbol{\infty}$	∞
TH_2	100	100
x_3		
d_3	1	**0**
CX_3	$\{(x_0, 1), (x_2, 1)\}$	$\boldsymbol{\{(x_0, 0), (x_2, 1)\}}$
LB_3	100	**11**
UB_3	100	**11**
x_4		
d_4	0	0
CX_4	$\{(x_0, 1)\}$	$\{(x_0, 1)\}$
LB_4	1000	1000
UB_4	1000	1000

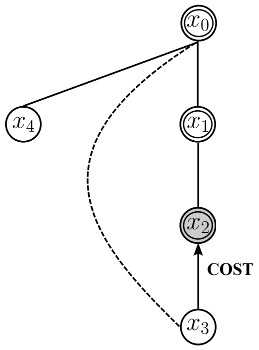

Figure A.59: Step 6' (3). x_2 receives the COST message from x_3. At this point, x_2 computes the bounds as $LB_2(1) = UB_2(1) = 11$ because context $\{(x_0, 0), (x_2, 1)\}$ in the COST message is compatible with $CX_2 = \{(x_0, 0), (x_1, 0)\}$. Then, x_2 updates the threshold as $TH_2 = 11$ due to ThresholdInvariant. Since $UB_2 = UB_2(1) = TH_2$, x_2 does not change its variable value d_2 from 1. Therefore, x_2 satisfies the termination condition and terminates with the suboptimal value $d_2 = 1$.

Table A.25: States of agents in Step 6' (3).

Variables	Step 6' (3)
x_0	
d_0	0
LB_0	1
$lb_0(0, x_1)$	1
$lb_0(0, x_4)$	0
$lb_0(1, x_1)$	0
$lb_0(1, x_4)$	1000
UB_0	1
$ub_0(0, x_1)$	1
$ub_0(0, x_4)$	0
$ub_0(1, x_1)$	∞
$ub_0(1, x_4)$	1000
TH_0	1
$th_0(0, x_1)$	1
$th_0(0, x_4)$	0
$th_0(1, x_1)$	0
$th_0(1, x_4)$	1000
x_1	
d_1	0
CX_1	$\{(x_0, 0)\}$
$cx_1(0, x_2)$	$\{(x_0, 0)\}$
LB_1	1
$lb_1(0, x_2)$	1
UB_1	1
$ub_1(0, x_2)$	1
TH_1	1
x_2	
d_2	1
CX_2	$\{(x_0, 0),$ $(x_1, 0)\}$
$cx_2(0, x_3)$	$\{\}$
$cx_2(1, x_3)$	$\boldsymbol{\{(x_0, 0)\}}$
LB_2	0
$lb_2(0, x_3)$	0
$lb_2(1, x_3)$	**11**
UB_2	**11**
$ub_2(0, x_3)$	∞
$ub_2(1, x_3)$	**11**
TH_2	**11**
x_3	
d_3	0
CX_3	$\{(x_0, 0),$ $(x_2, 1)\}$
LB_3	11
UB_3	11
x_4	
d_4	0
CX_4	$\{(x_0, 1)\}$
LB_4	1000
UB_4	1000

References

[1] Z. Chen, C. He, Z. He, M. Chen, BD-ADOPT: A hybrid DCOP algorithm with best-first and depth-first search strategies, Artificial Intelligence Review 50 (2) (2018) 161–199. doi:10.1007/s10462-017-9540-z.

[2] F. Fioretto, E. Pontelli, W. Yeoh, Distributed constraint optimization problems and applications: A survey, Journal of Artificial Intelligence Research 61 (2018) 623–698. doi:10.1613/jair.5565.

[3] P. Gutierrez, P. Meseguer, W. Yeoh, Generalizing ADOPT and BnB-ADOPT, in: T. Walsh (Ed.), Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, 2011, pp. 554–559. doi:10.5591/978-1-57735-516-8/IJCAI11-100.

[4] P. Gutierrez, P. Meseguer, Saving messages in ADOPT-based algorithms, in: AAMAS 2010 Workshop: Distributed Constraint Reasoning, Toronto, Canada, 2010, pp. 53–64.

[5] M. Jain, M. E. Taylor, M. Tambe, M. Yokoo, DCOPs meet the real world: Exploring unknown reward matrices with applications to mobile sensor networks., in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, 2009, pp. 181–186.

[6] R. N. Lass, J. B. Kopena, E. A. Sultanik, D. N. Nguyen, C. P. Dugan, P. J. Modi, W. C. Regli, Coordination of first responders under communication and resource constraints, in: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Vol. 3, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2008, pp. 1409–1412.

[7] R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, P. Varakantham, Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling, in: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Vol. 1 of AAMAS '04, IEEE Computer Society, USA, 2004, pp. 310–317.

[8] P. J. Modi, W.-M. Shen, M. Tambe, M. Yokoo, Adopt: Asynchronous distributed constraint optimization with quality guarantees, Artificial Intelligence 161 (1-2) (2005) 149–180. doi:10.1016/j.artint.2004.09.003.

[9] F. Pecora, P. J. Modi, P. Scerri, Reasoning about and dynamically posting n-ary constraints in ADOPT, in: Proceedings of Seventh Workshop on Distributed Constraint Reasoning, 2006, pp. 57–71.

[10] M. C. Silaghi, M. Yokoo, ADOPT-ing: Unifying asynchronous distributed optimization with asynchronous backtracking, Autonomous Agents and Multi-Agent Systems 19 (2) (2009) 89–123. doi:10.1007/s10458-008-9069-2.

[11] M. C. Silaghi, M. Yokoo, Nogood based asynchronous distributed optimization (ADOPT ng), in: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, ACM Press, Hakodate, Japan, 2006, p. 1389. doi:10.1145/1160633.1160894.

[12] G. Weiss, Multiagent Systems, 2nd Edition, EBSCO Ebook Academic Collection, MIT Press, 2013.

[13] W. Yeoh, A. Felner, S. Koenig, BnB-ADOPT: An asynchronous branch-and-bound DCOP algorithm, Journal of Artificial Intelligence Research 38 (2010) 85–133. doi:10.1613/jair.2849.

[14] W. Yeoh, A. Felner, S. Koenig, IDB-ADOPT: A depth-first search DCOP algorithm, in: A. Oddi, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints, Vol. 5655 of Lecture Notes in Artificial Intelligence, Springer, Rome, Italy, 2009, pp. 132–146. doi:10.1007/978-3-642-03251-6{_}9.

[15] M. Yokoo, E. H. Durfee, T. Ishida, K. Kuwabara, The distributed constraint satisfaction problem: Formalization and algorithms, IEEE Transactions on Knowledge and Data Engineering 10 (5) (1998) 673–685. doi:10.1109/69.729707.

[16] K. Noshiro, K. Hasebe, Flaws of termination and optimality in ADOPT-based algorithms (in press), in: Proceedings of the 32nd International Joint Conference on Artificial Intelligence, 2023.