Improving AC–Algorithms With Double–Support Checks

M.R.C. van Dongen and
J.A. Bowen
University College Cork

August 26, 2000

Outline

- Constraint Networks.
- Arc–Consistency.
- Existing Arc–Consistency Algorithms.
- Double–Support Checks.
- Experimental Results.
- Discussion.

Constraint Networks

Let X be a set of variables. For all $x \in X$ let $D(x)$ denote the domain of x. Finally let $S = \{x_{i_1}, \ldots, x_{i_m}\} \in 2^X \setminus \emptyset$.

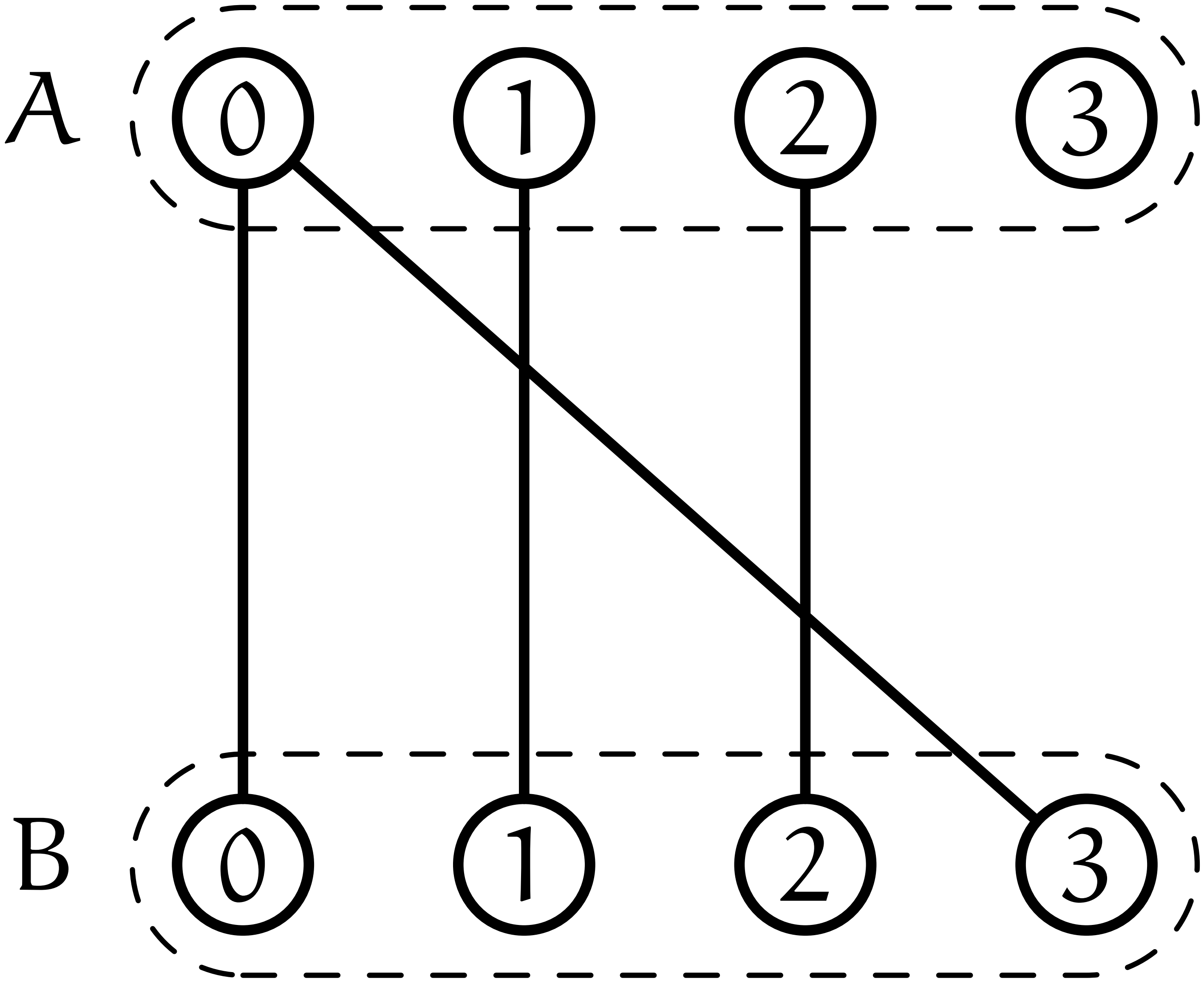
C_S is called a *constraint* on S if $C_S \subseteq \bigtimes_{x \in S} D(x)$.

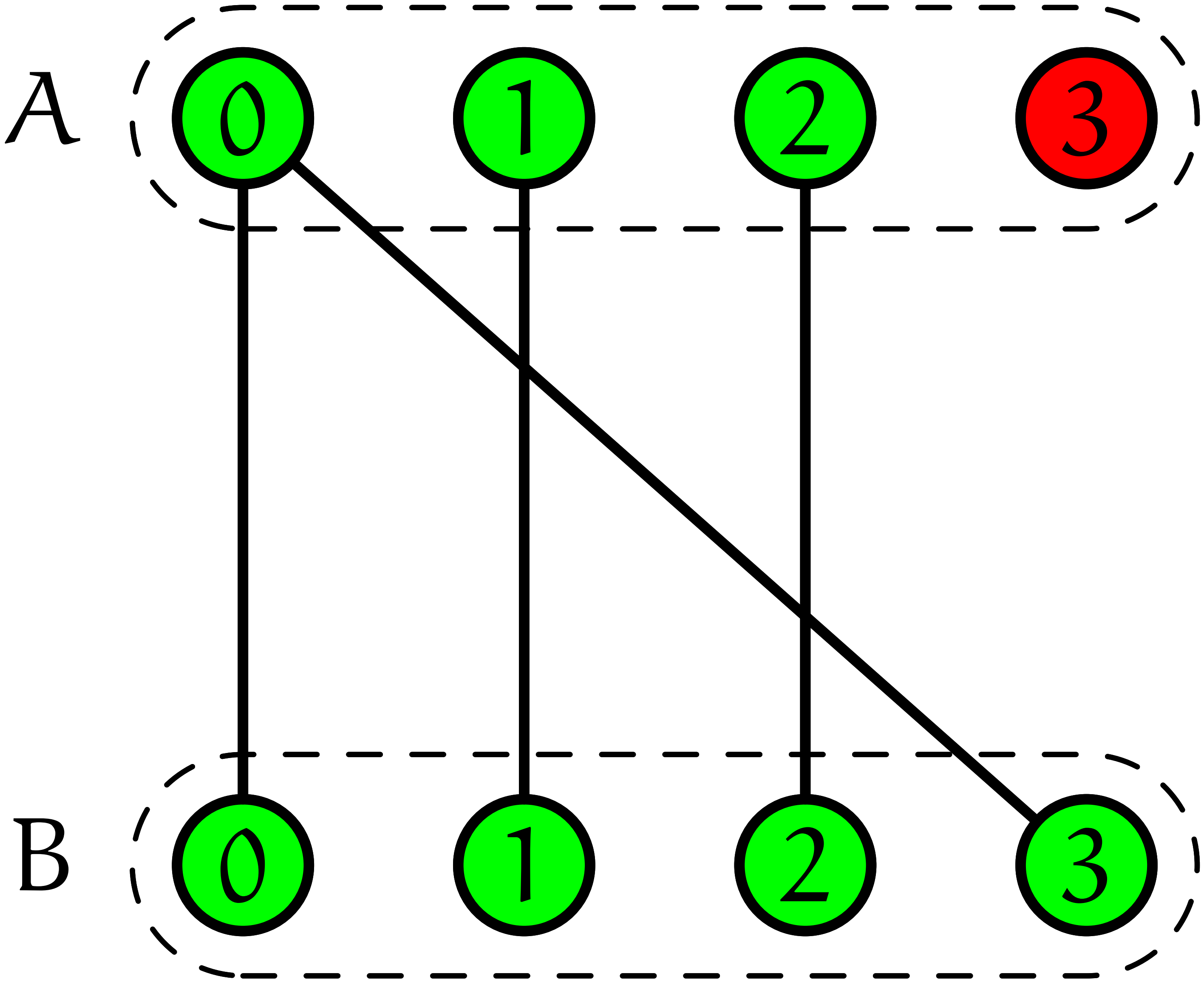
If $(v_{i_1}, \ldots, v_{i_m}) \in C_S$ it is said to *satisfy* C_S.

A *constraint network* is a collection of variables and constraints on those variables.

Arc–Consistency

A constraint–network is called *arc–consistent* iff for every variable, say A it holds that for every value, say v_{A}, in $D(A)$ and for every constraint $C_{\{A,B\}}$ in the constraint network there is a value, say v_{B}, in $D(B)$ s.t. v_{B} *supports* v_{A}.


A
0
1
2
3
B
0
1
2
3

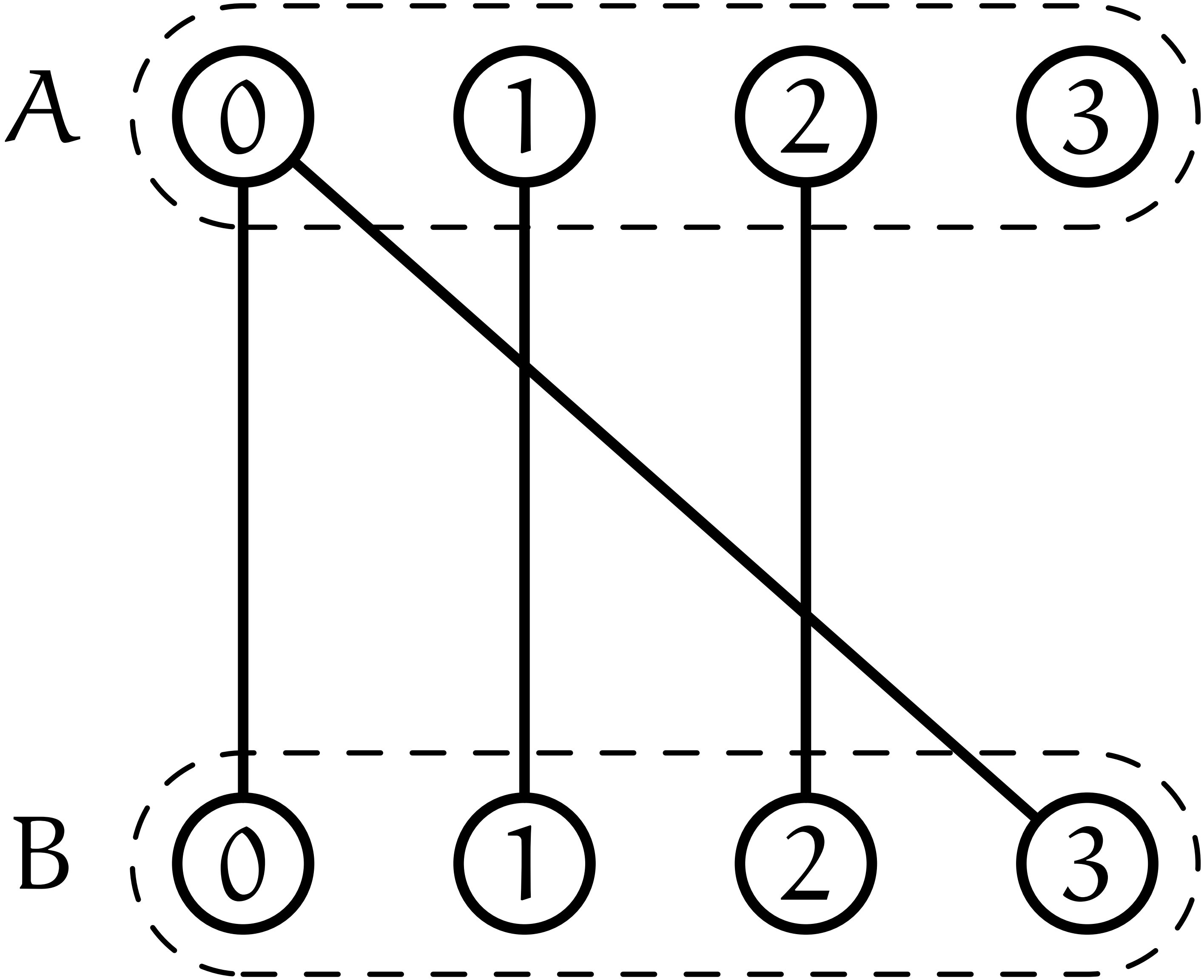

A
0
1
2
3
B
0
1
2
3

Existing Arc–Consistency Algorithms

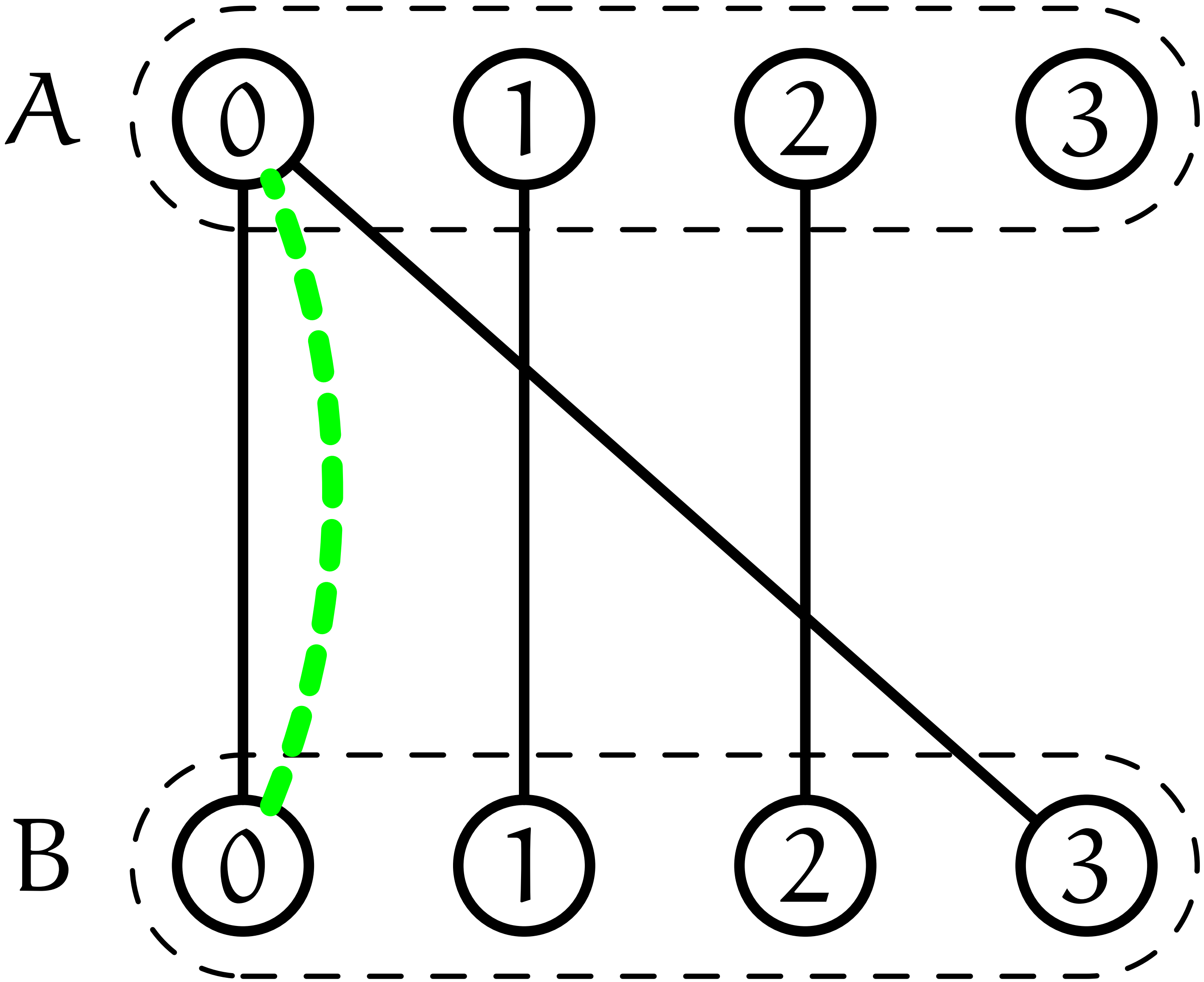
DEE Uses a queue of edges. Finds support for the values in the domains at both ends of the edge.

AC-3 Uses a queue of arcs. When processing the arc from A to B it finds support for the values in $\mathrm{D}(A)$ with $\mathrm{D}(\mathrm{B})$. It has a $\mathbf{O}(\mathrm{ed}^3)$ time–complexity.

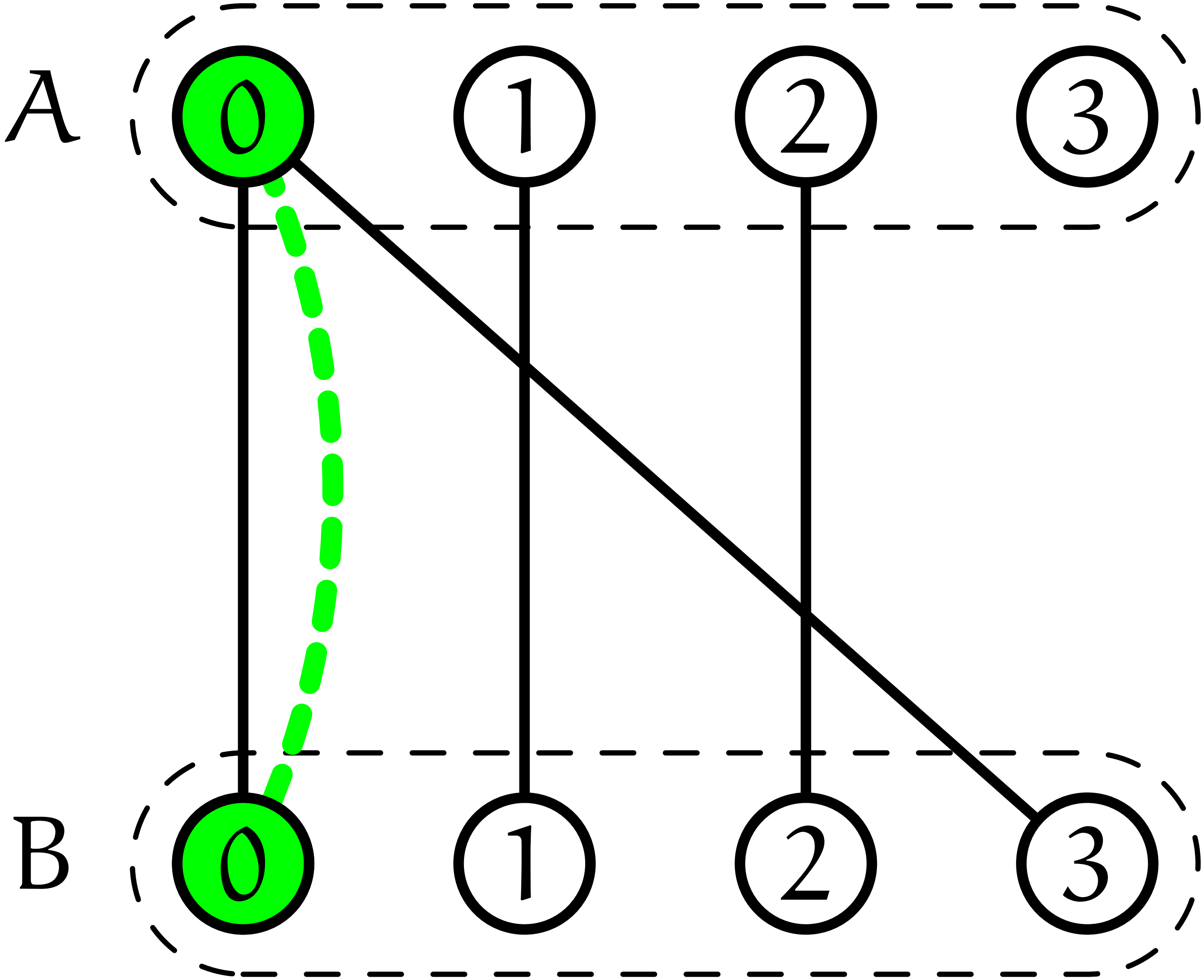
AC-7 Never repeats a consistency–check. It has an optimal $\mathbf{O}(\mathrm{ed}^2)$ time–complexity.



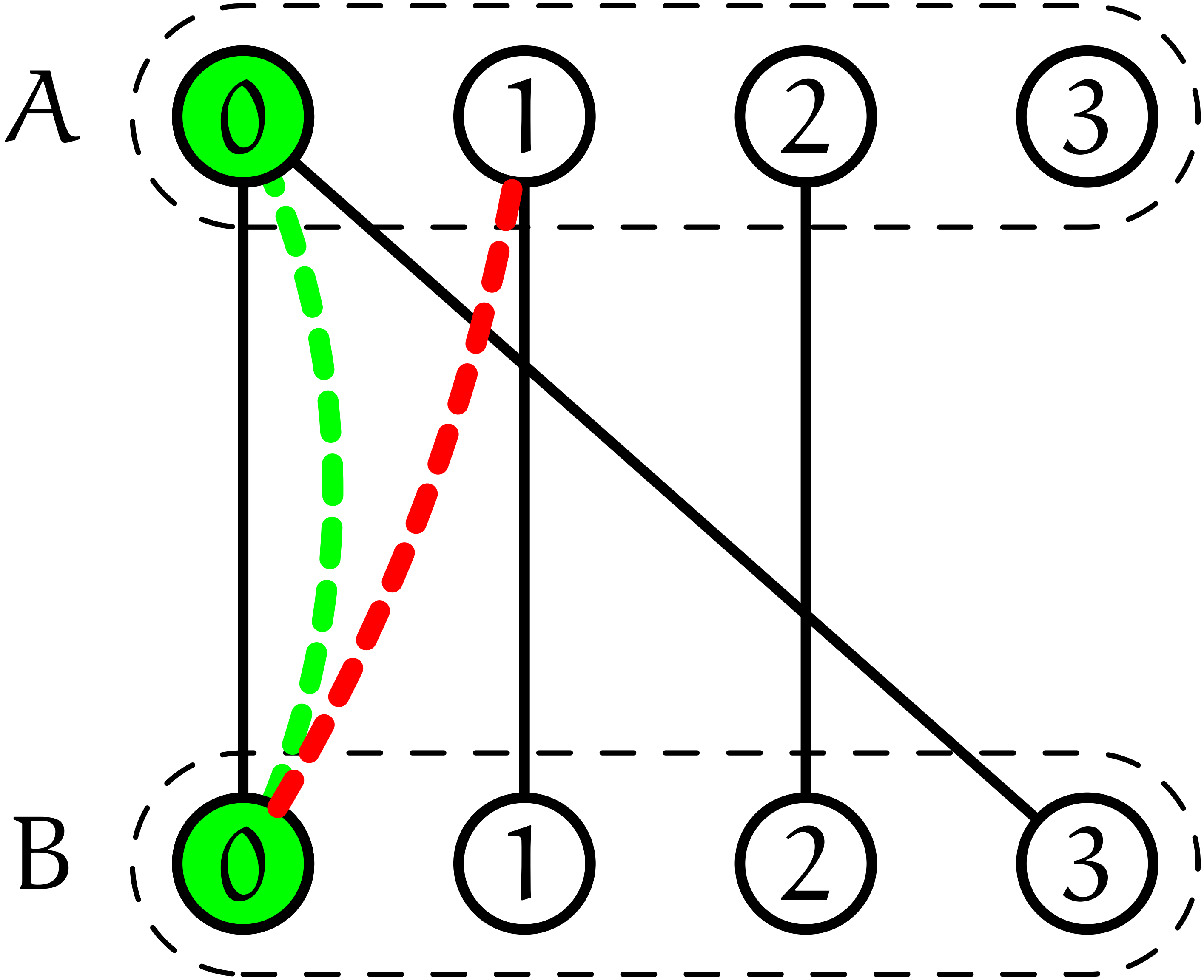
AC-7 #CC (0)



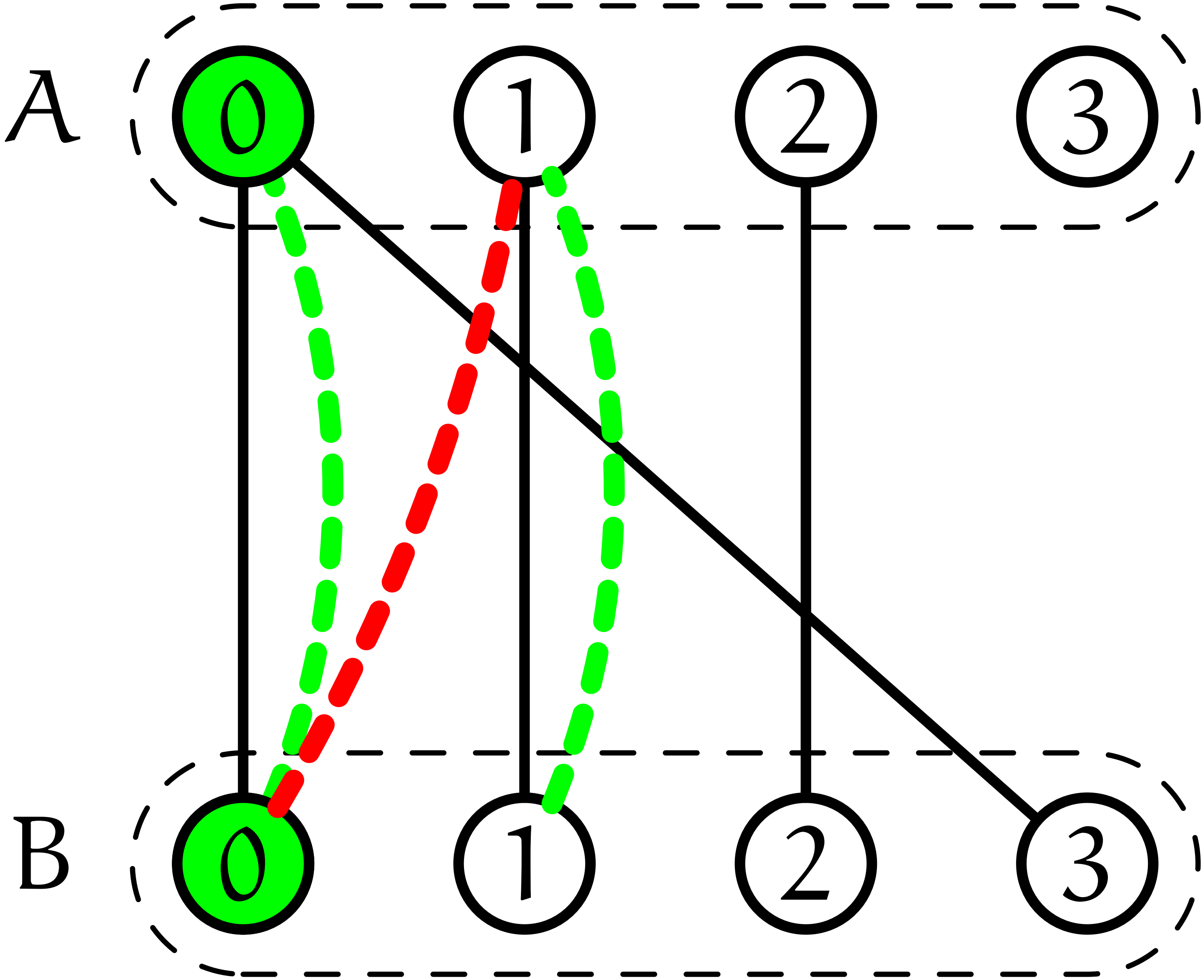
AC-7 #CC (1)



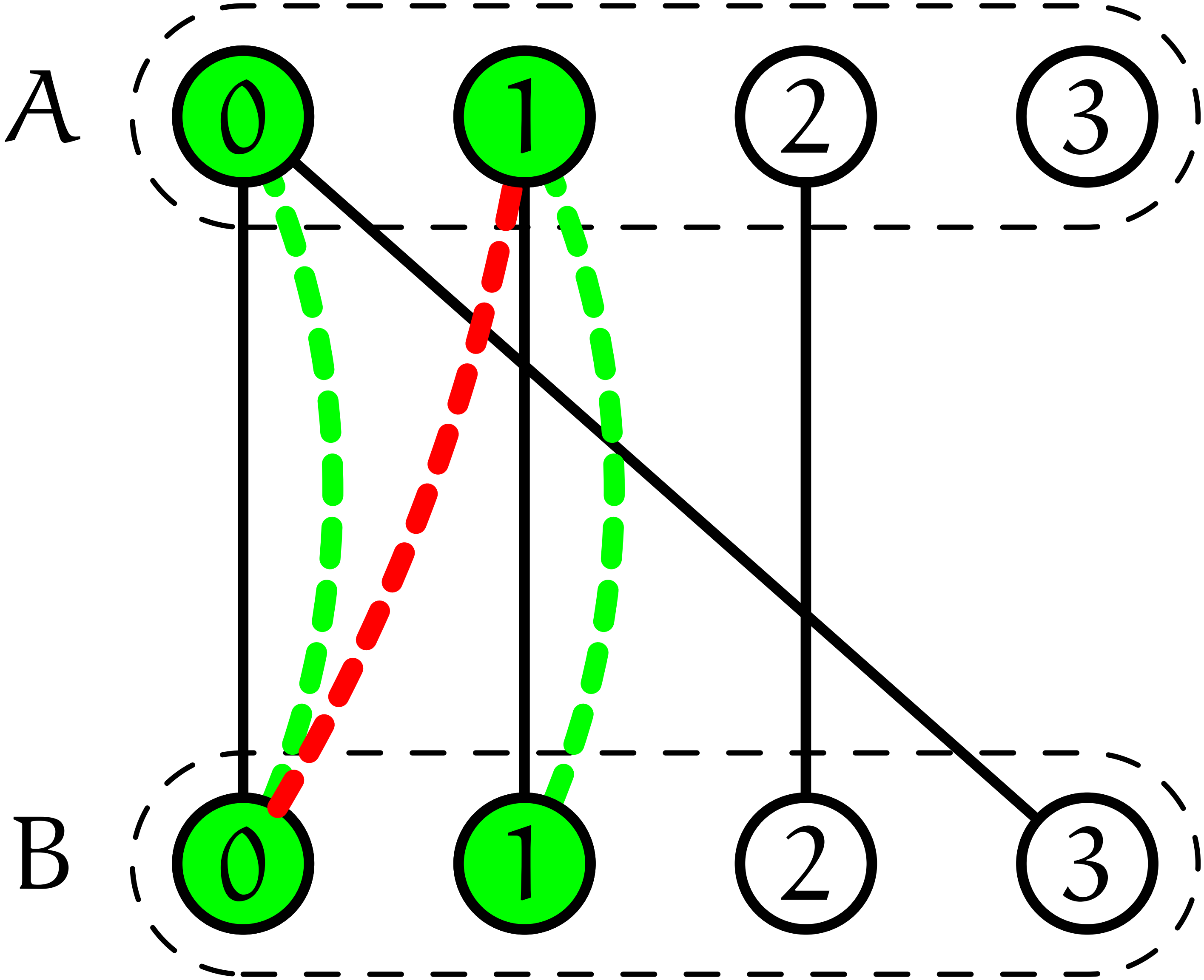
AC-7 #CC (1)



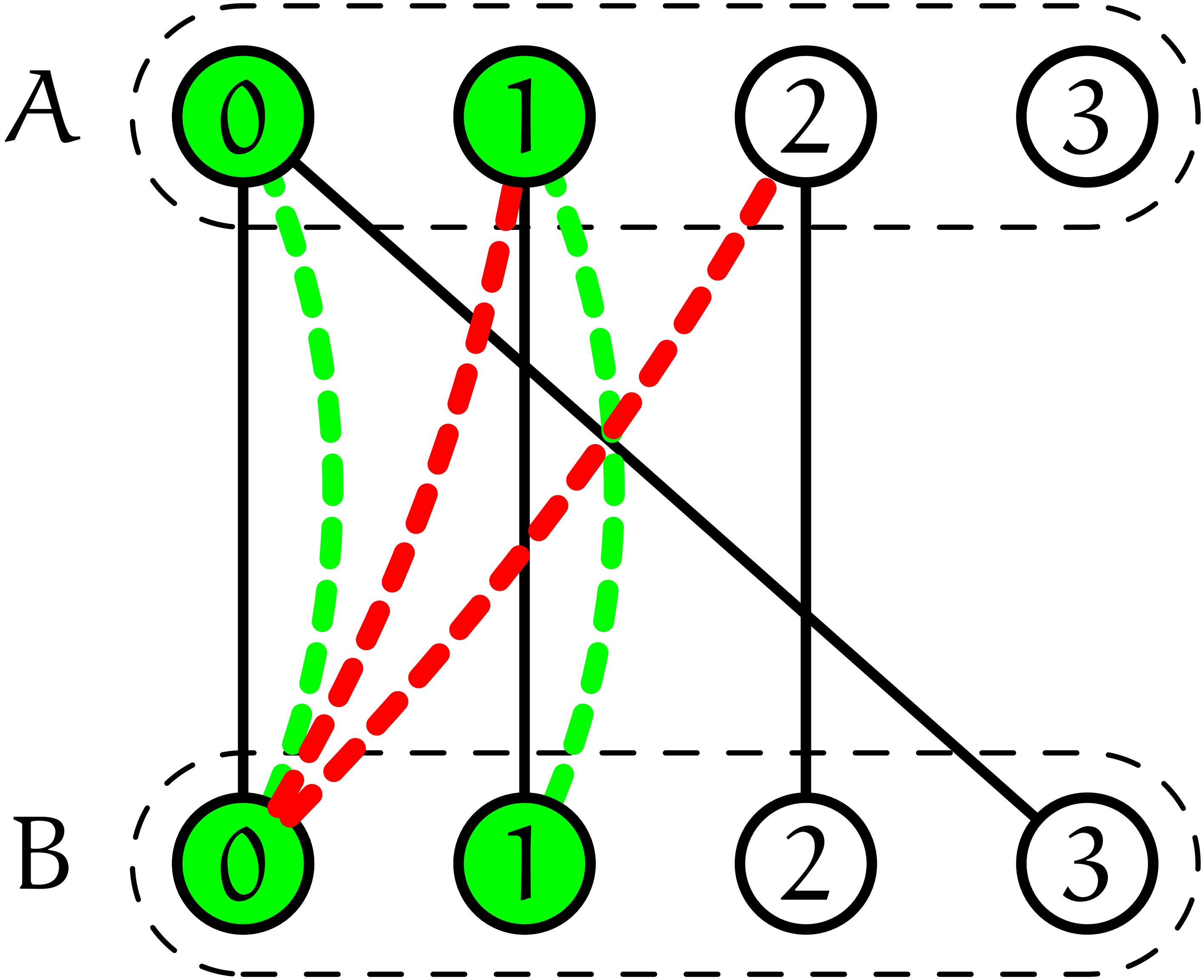
AC-7 #CC (2)



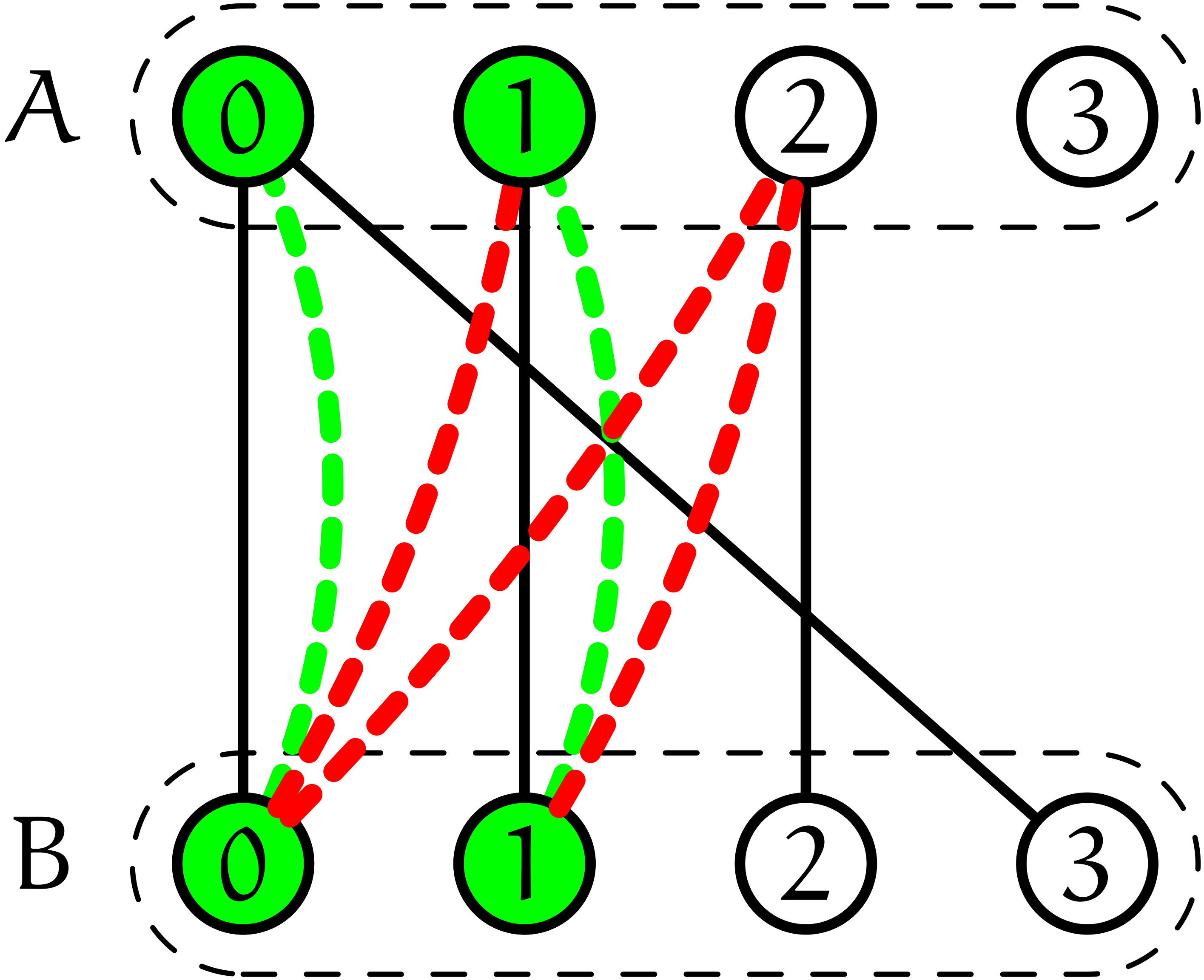
AC-7 #CC (3)



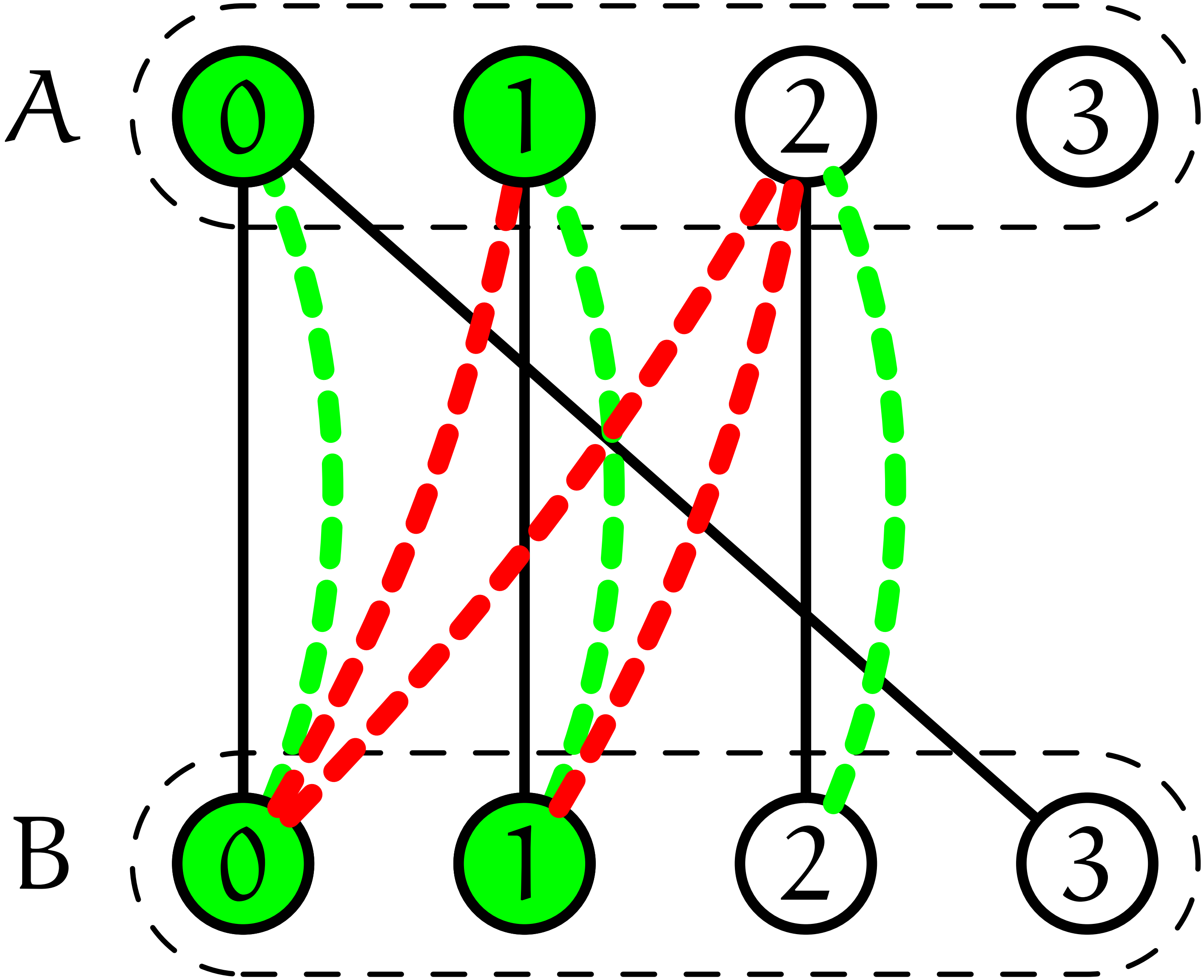
AC-7 #CC (3)



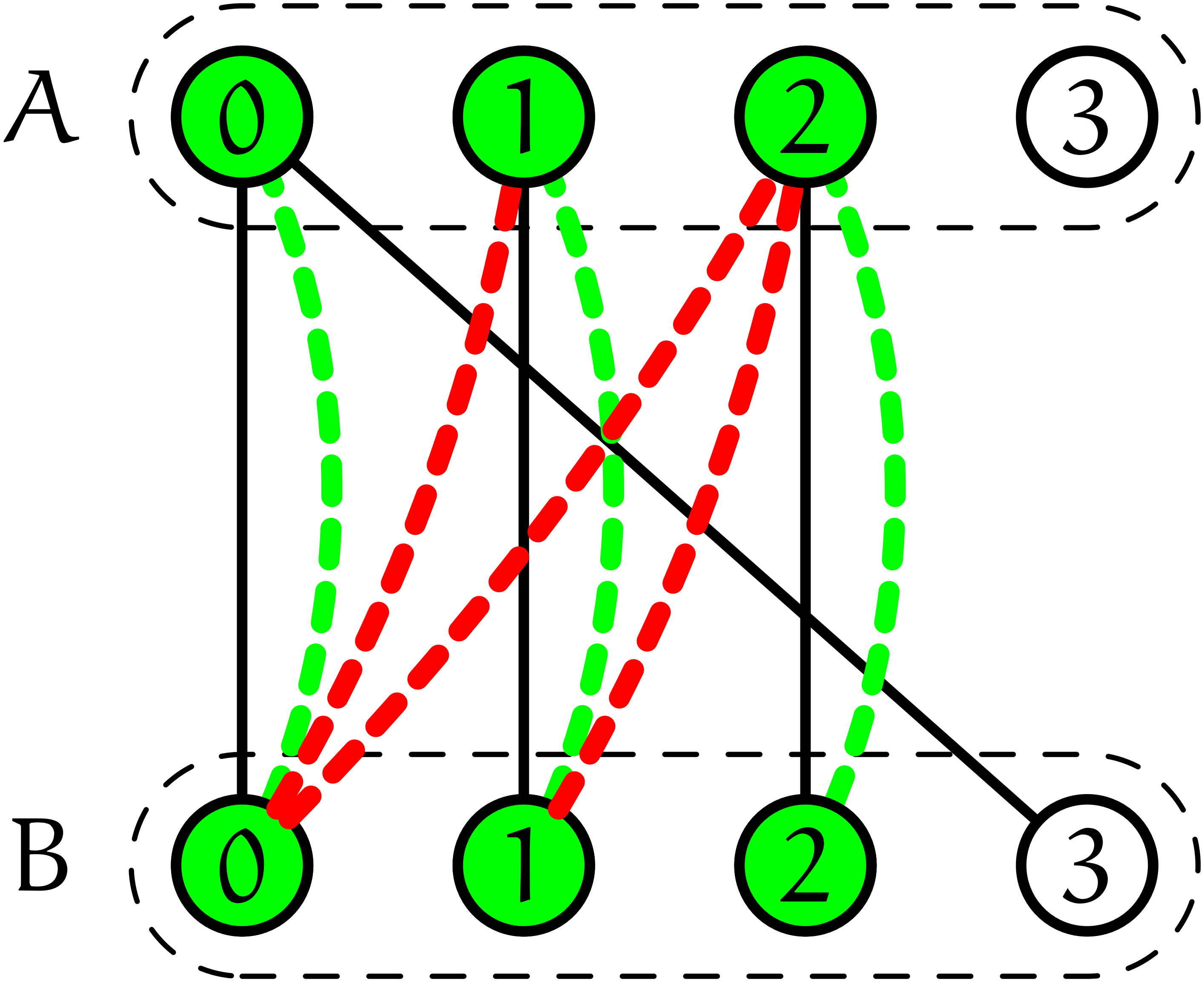
AC-7 #CC (4)



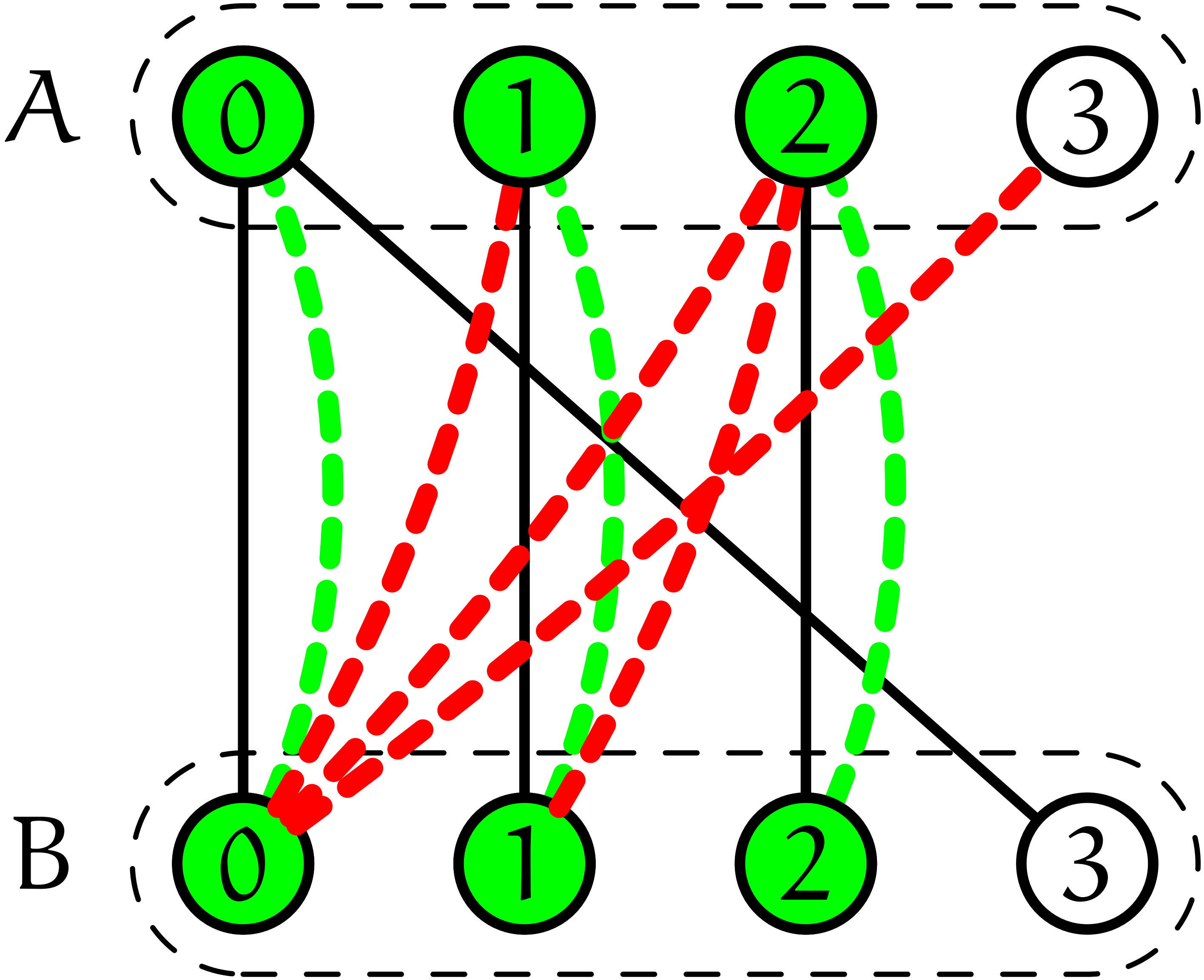
AC-7 #CC (5)



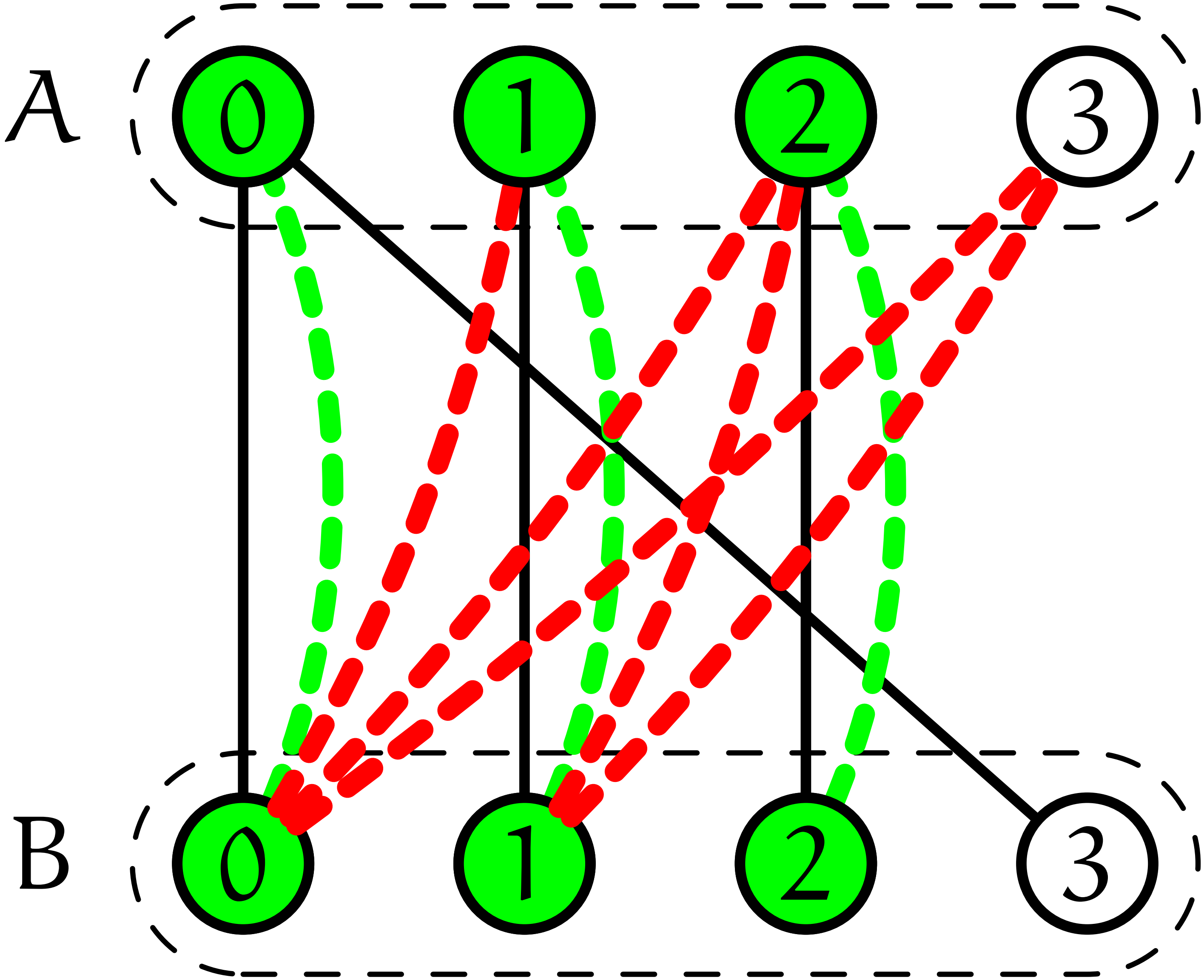
AC-7 #CC (6)



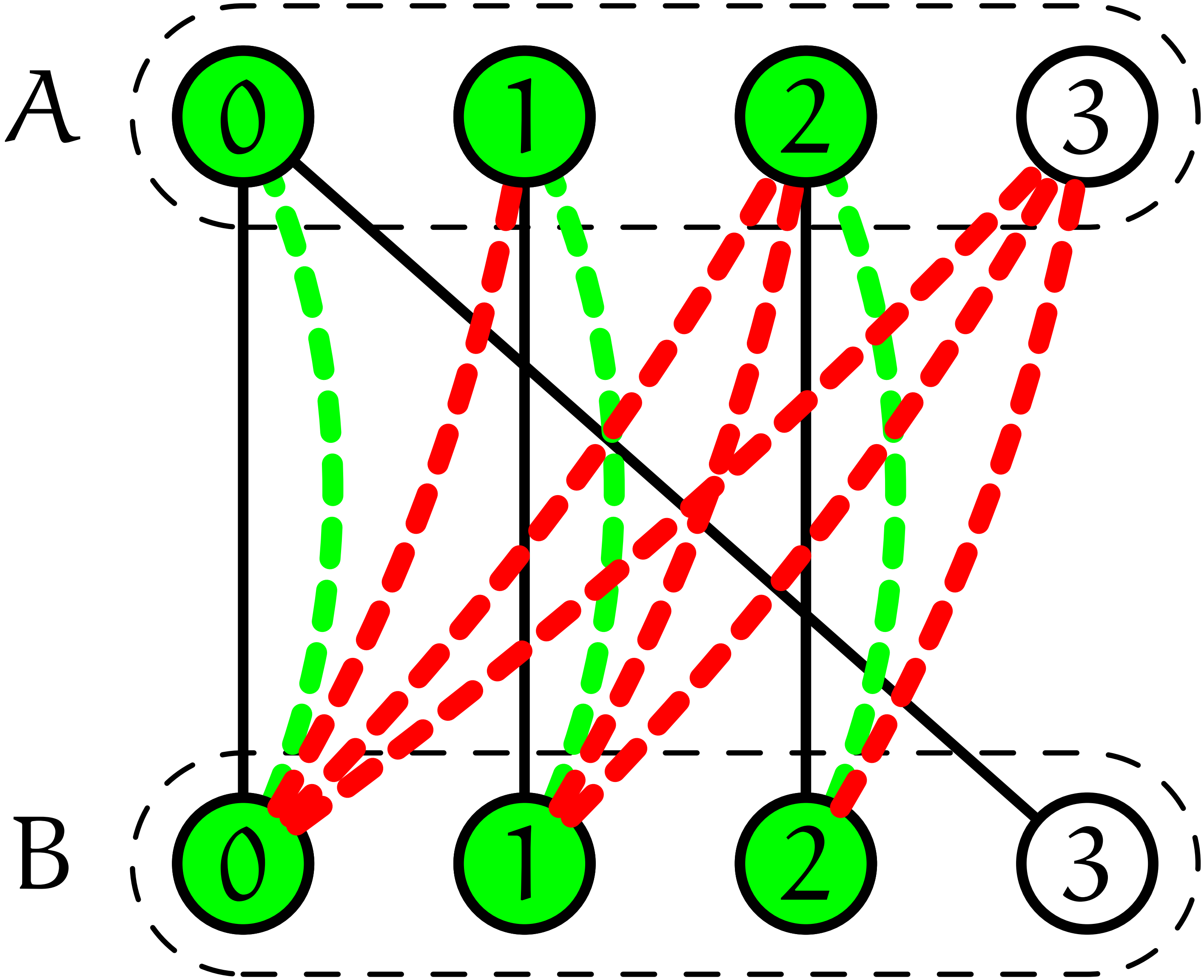
AC-7 #CC (6)



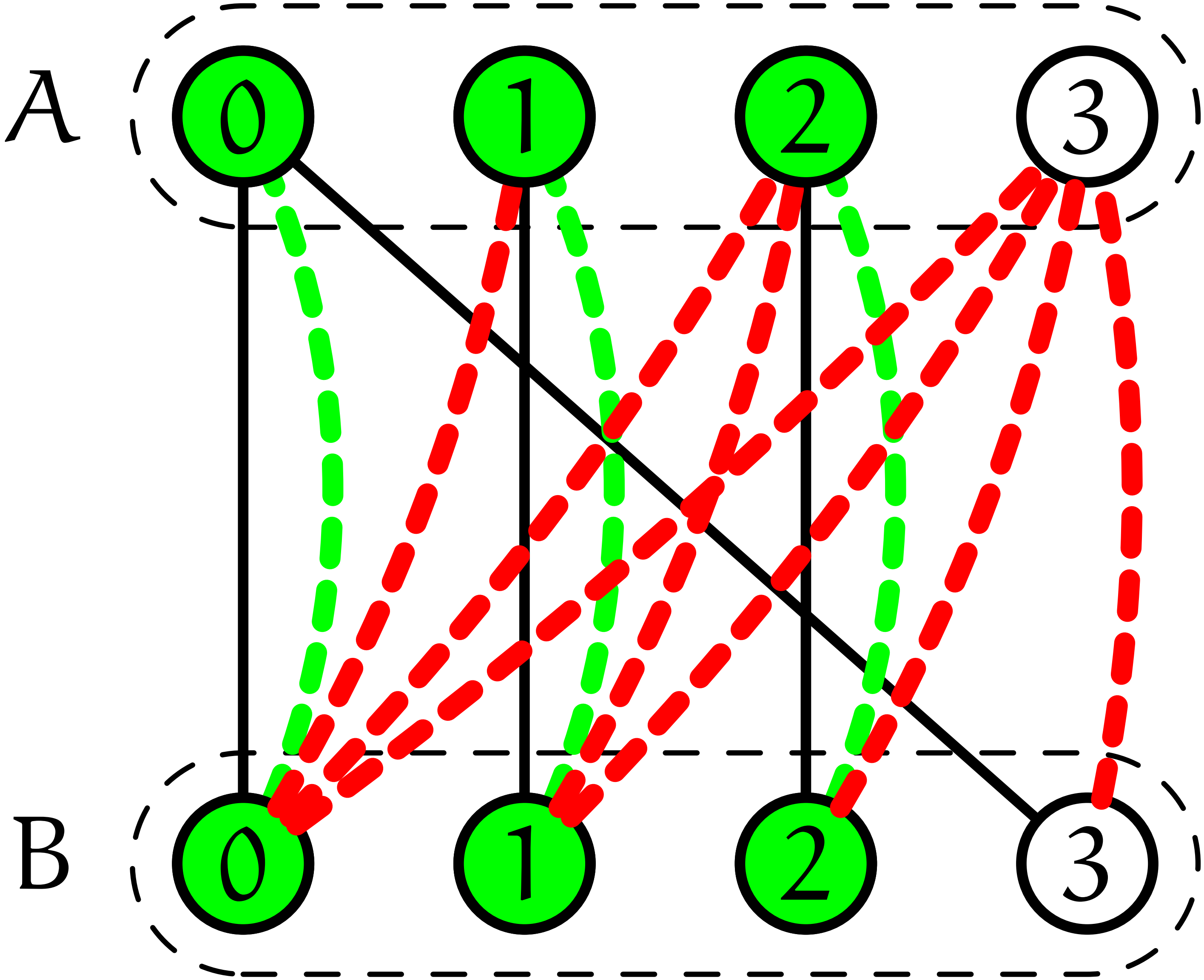
AC-7 #CC (7)



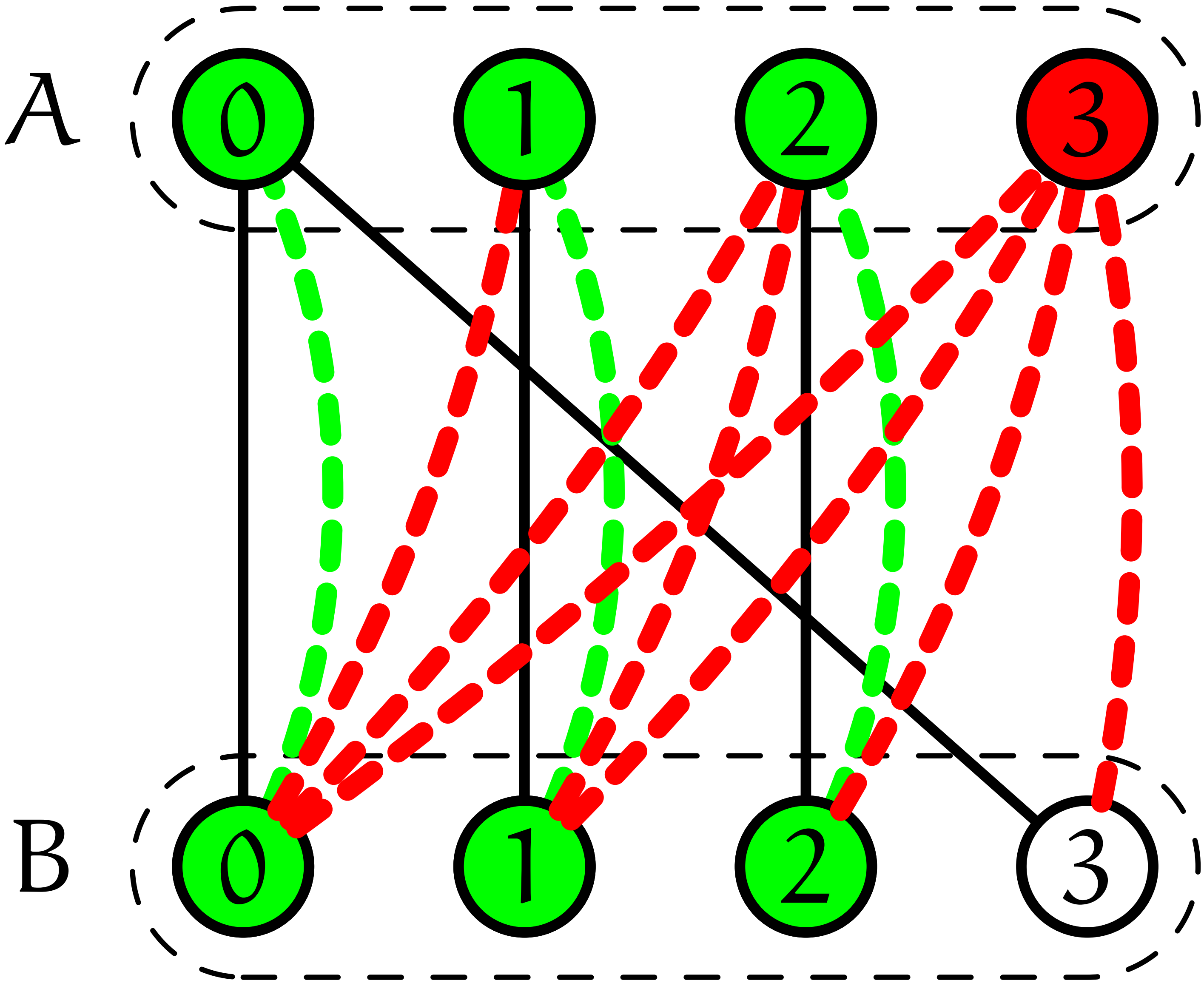
AC-7 #CC (8)



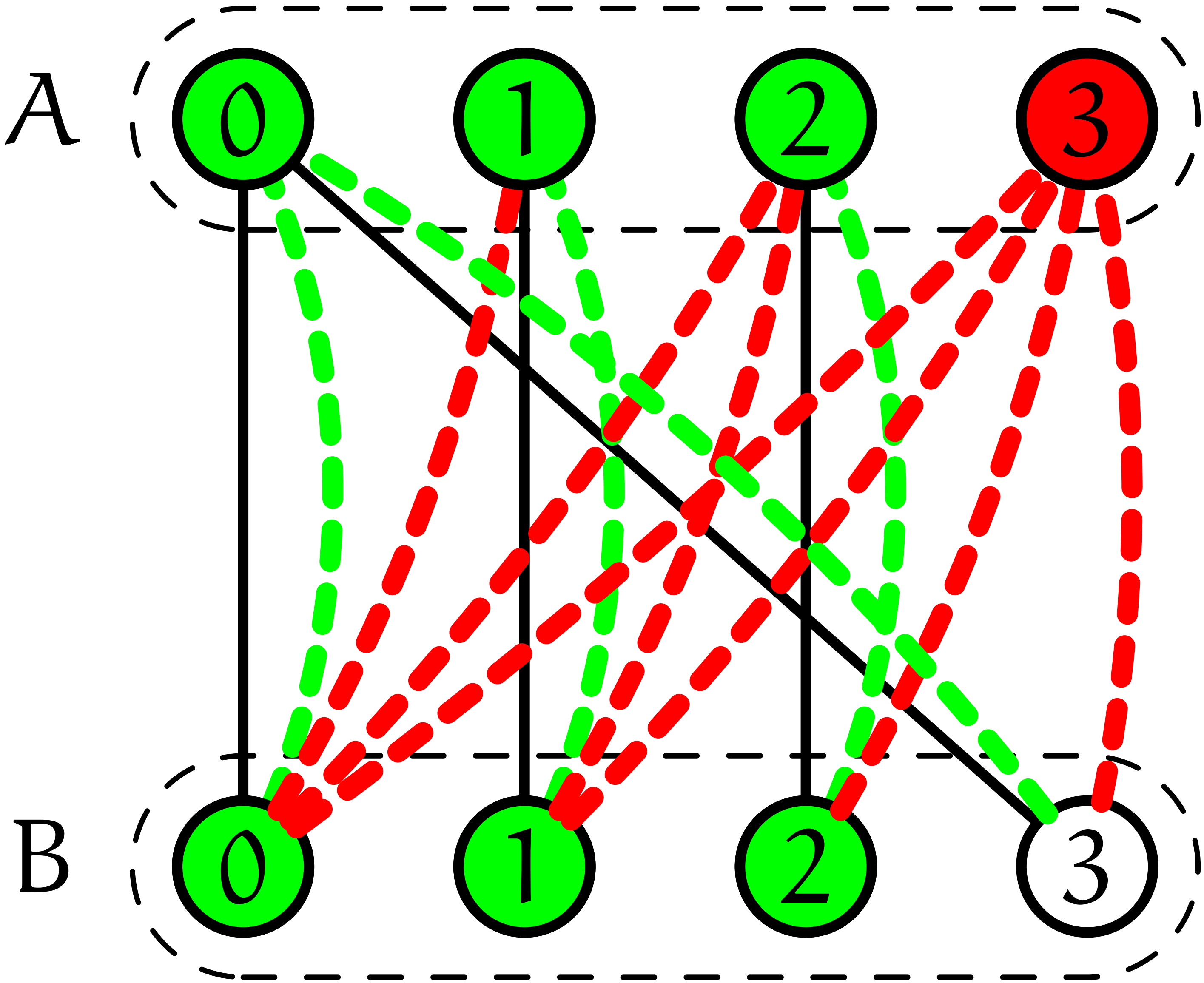
AC-7 #CC (9)



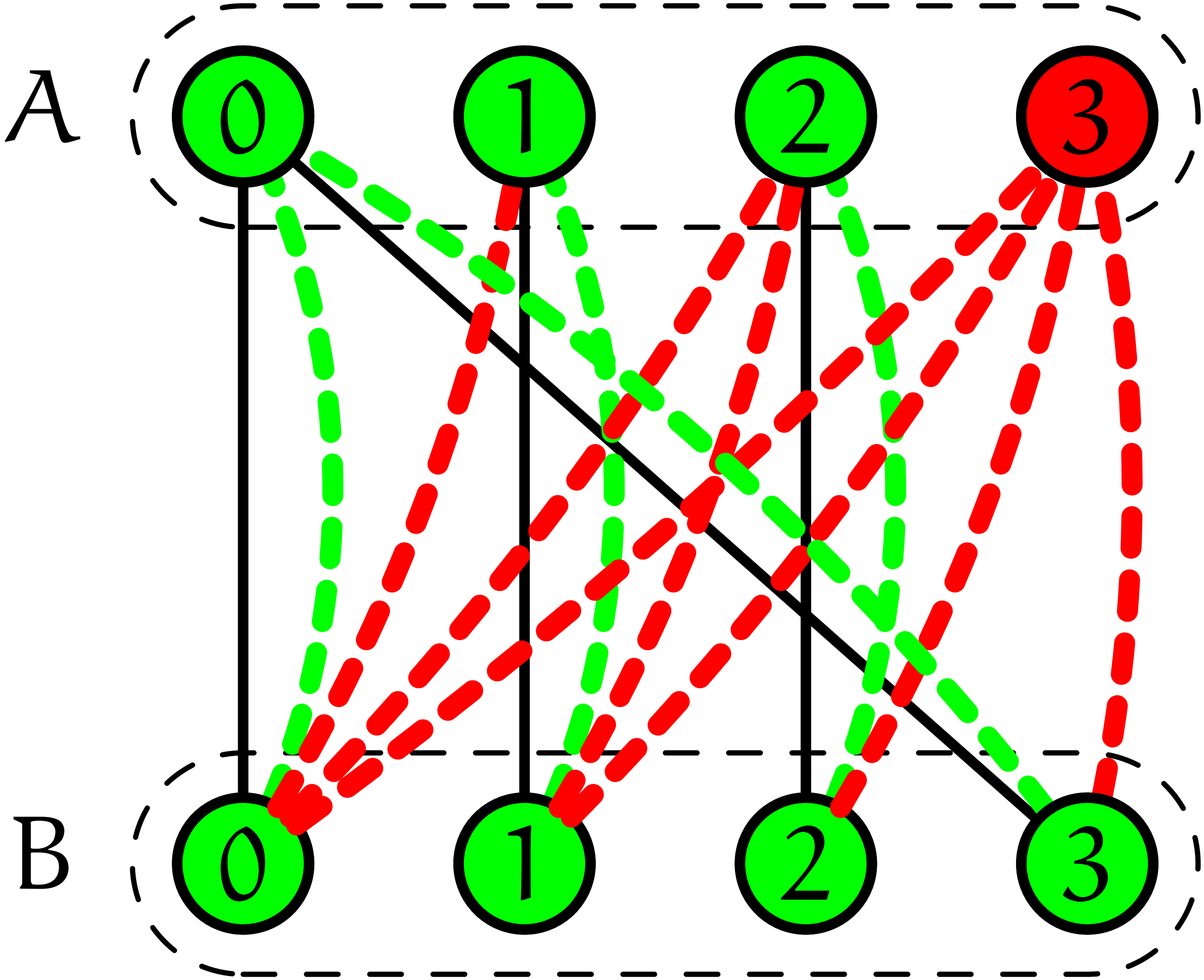
AC-7 #CC (10)



AC-7 #CC (10)



AC-7 #CC (11)

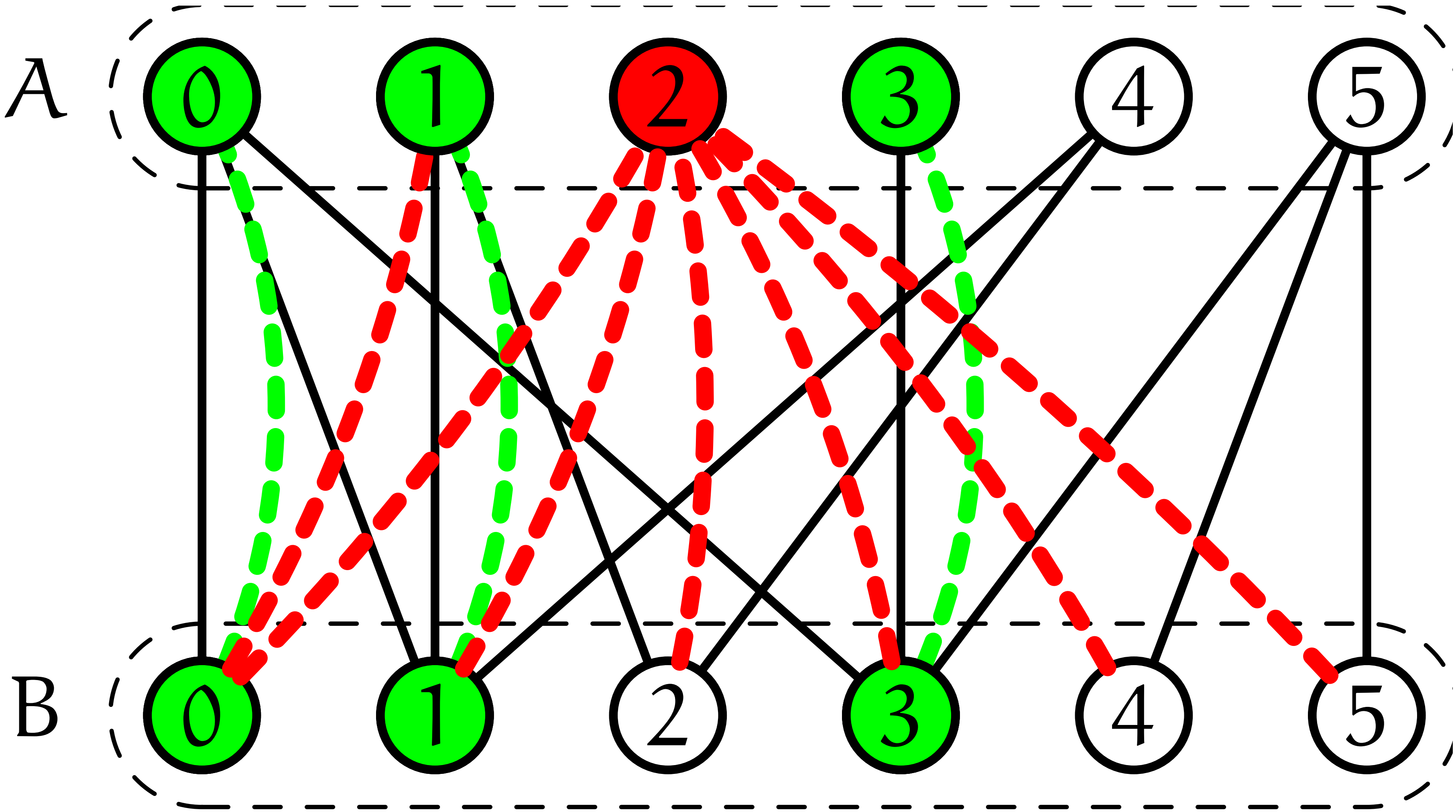


AC-7 #CC (11)

Double Support Checks

A *double–support check* is a consistency–check which seeks to find support for *two* values, whose support–statuses before the check are unknown.

Note 1. *To minimise the number of consistency–checks the number of successful double–support checks has to be maximised.*

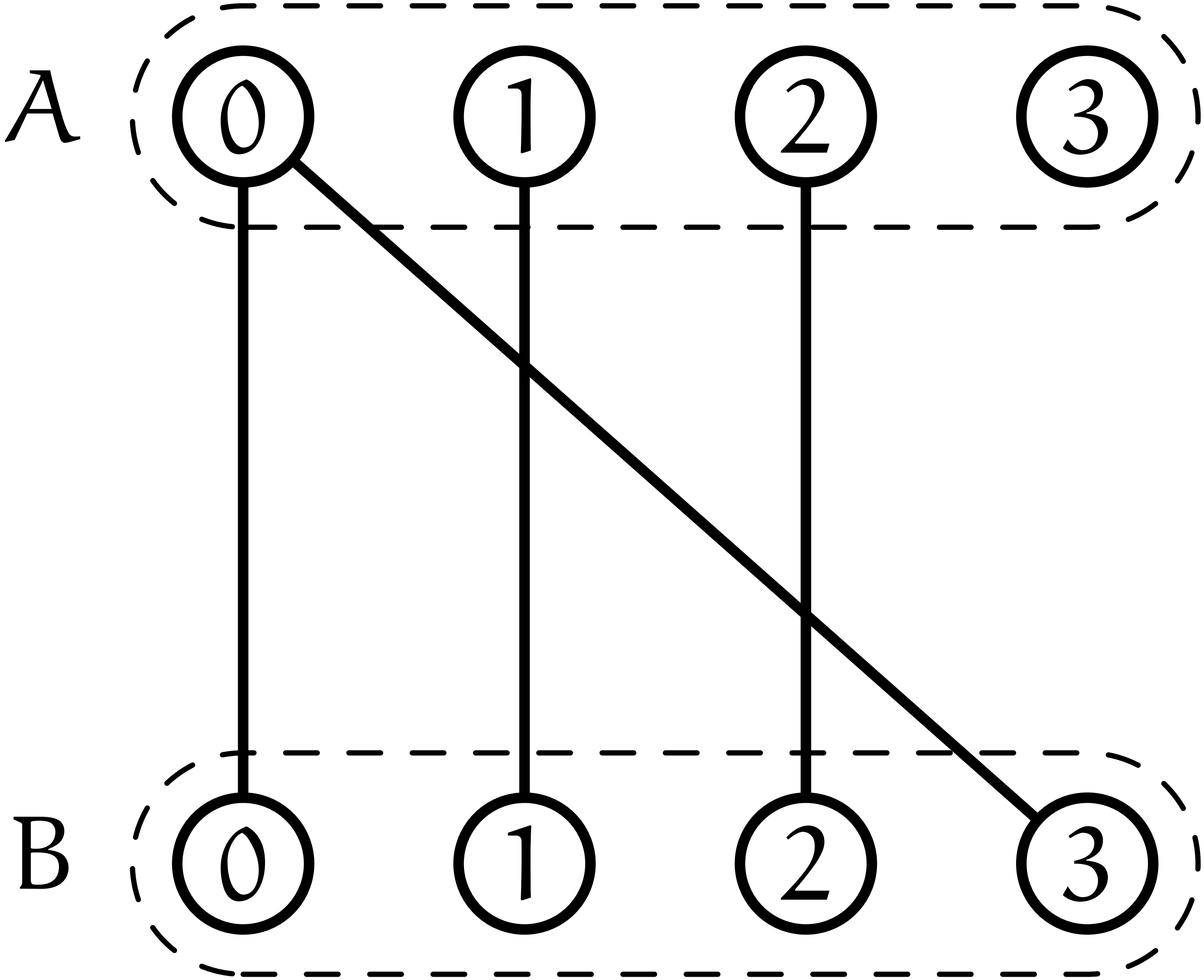

A
0
1
2
3
4
5
B
0
1
2
3
4
5

AC-3_b

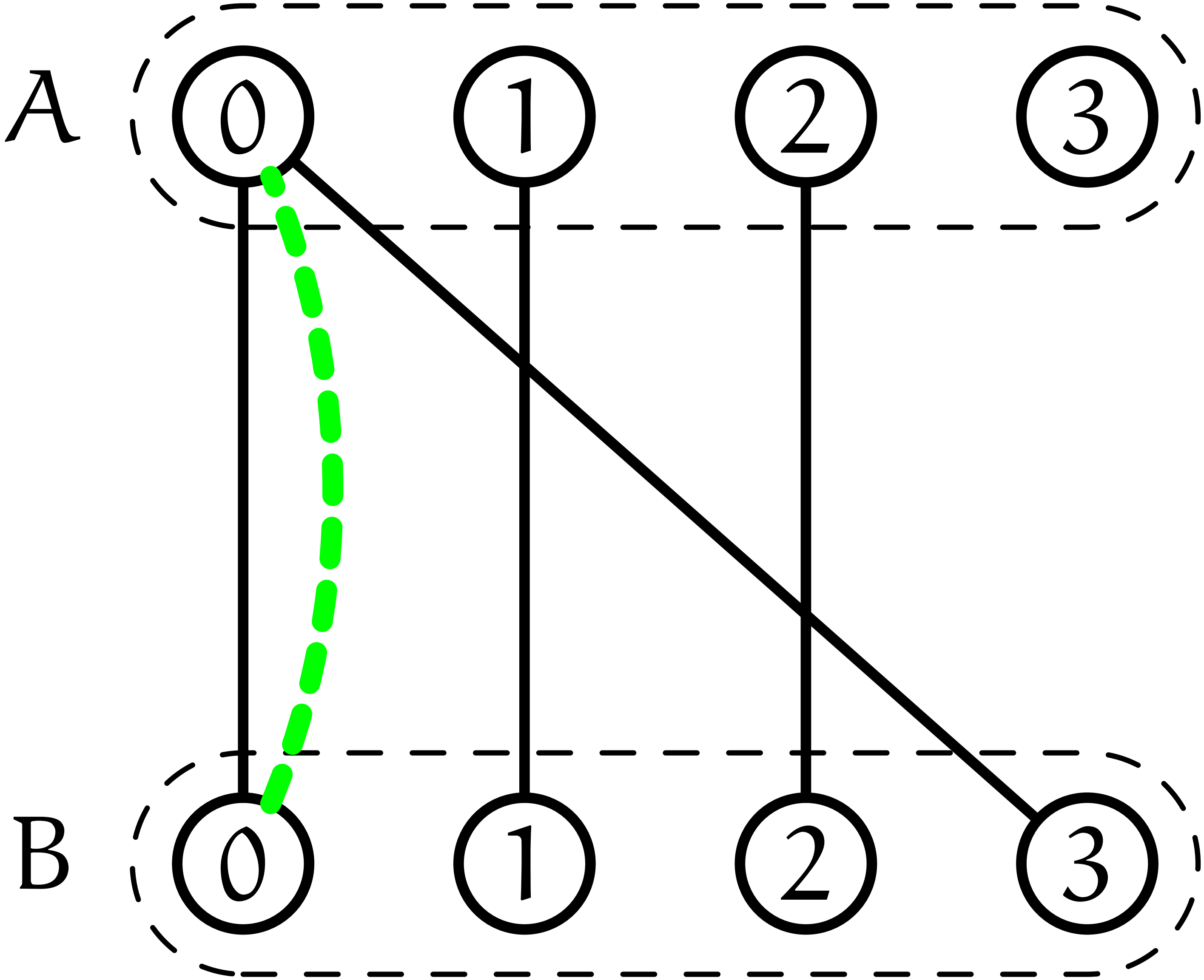
- It is a cross–breed between AC-3 and DEE.

- It uses a heuristic which attempts to maximise the number of *successful* double–support checks.

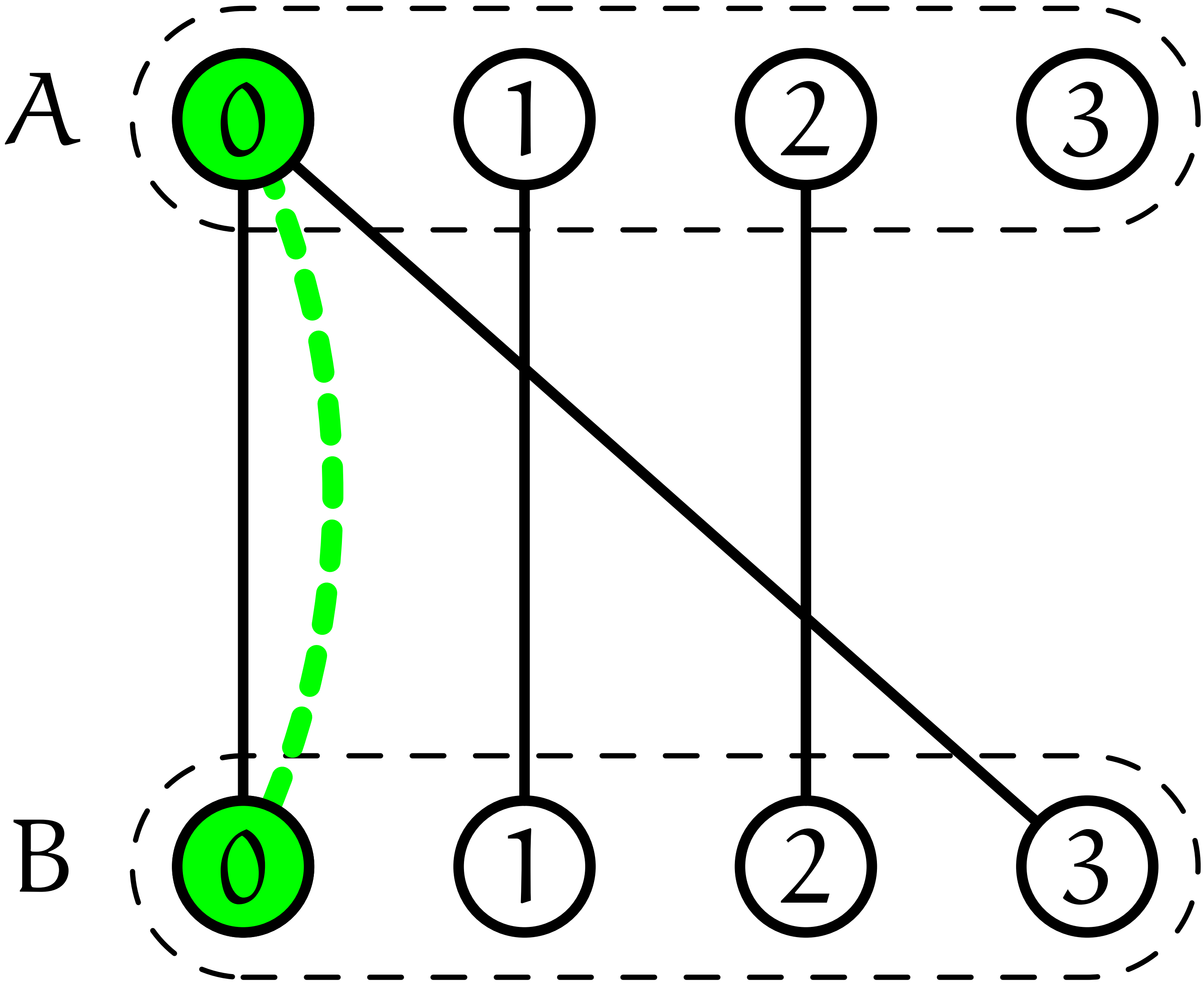
- It has a $\mathbf{O}(ed^3)$ time–complexity.



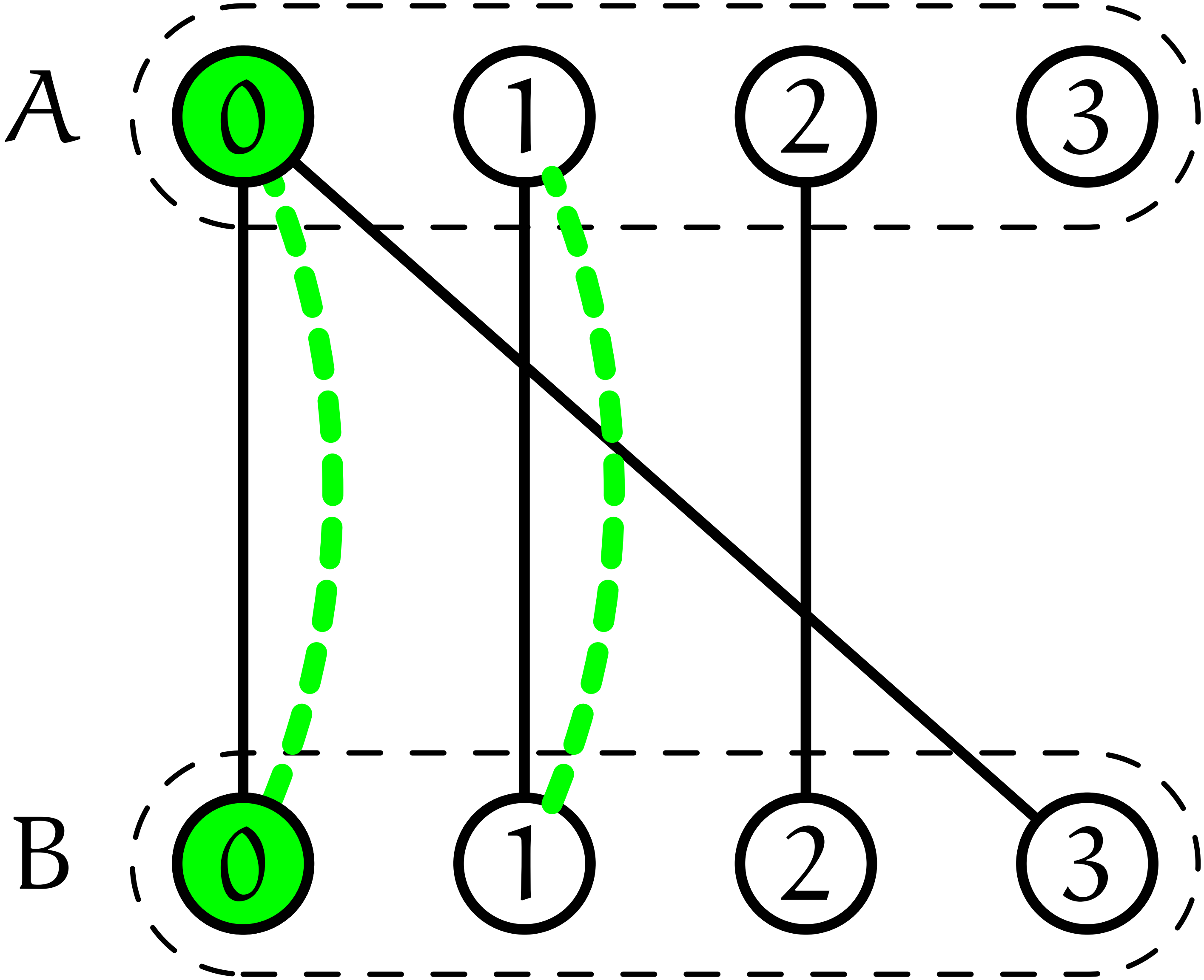
AC-3$_b$ #CC (0)



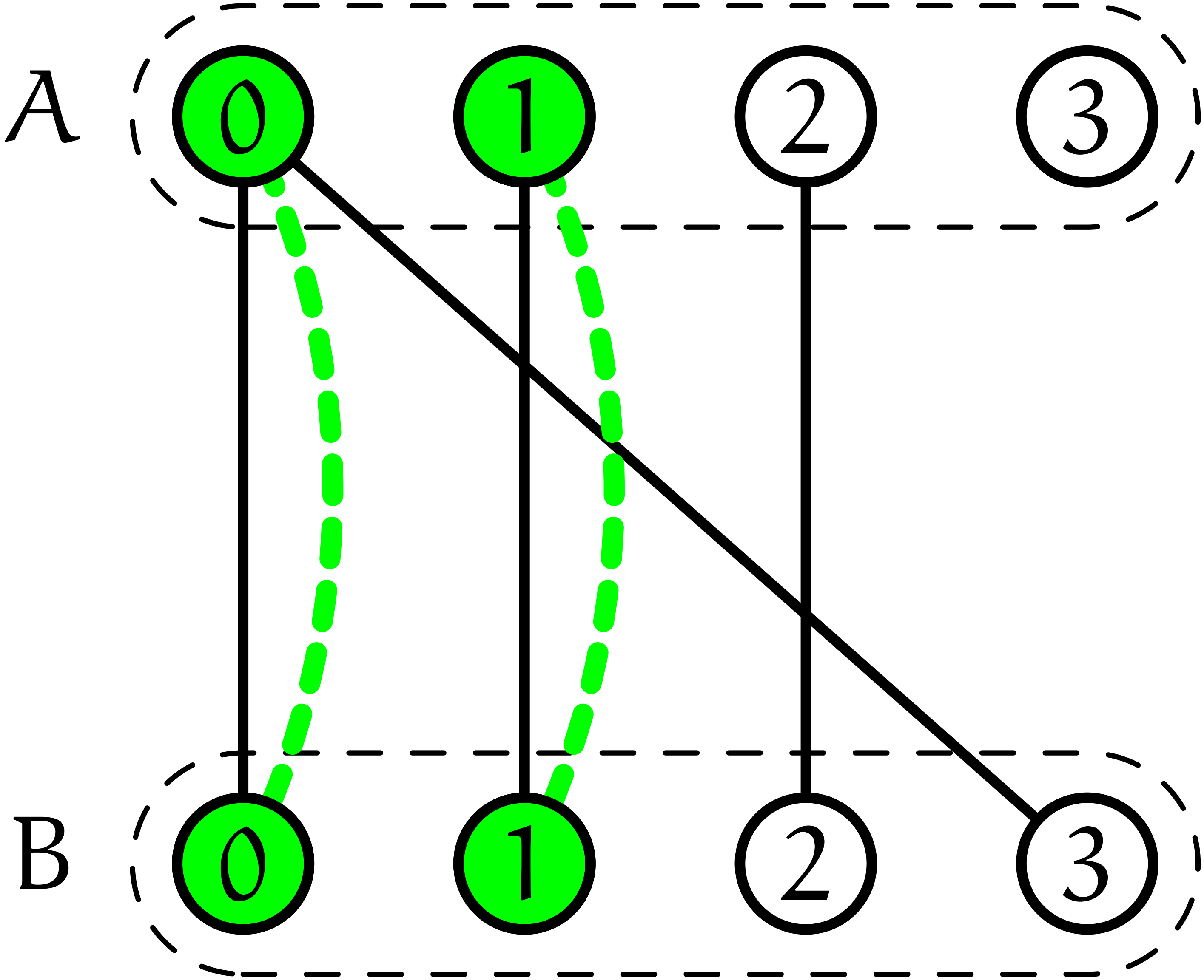
AC-3$_b$ #CC (1)



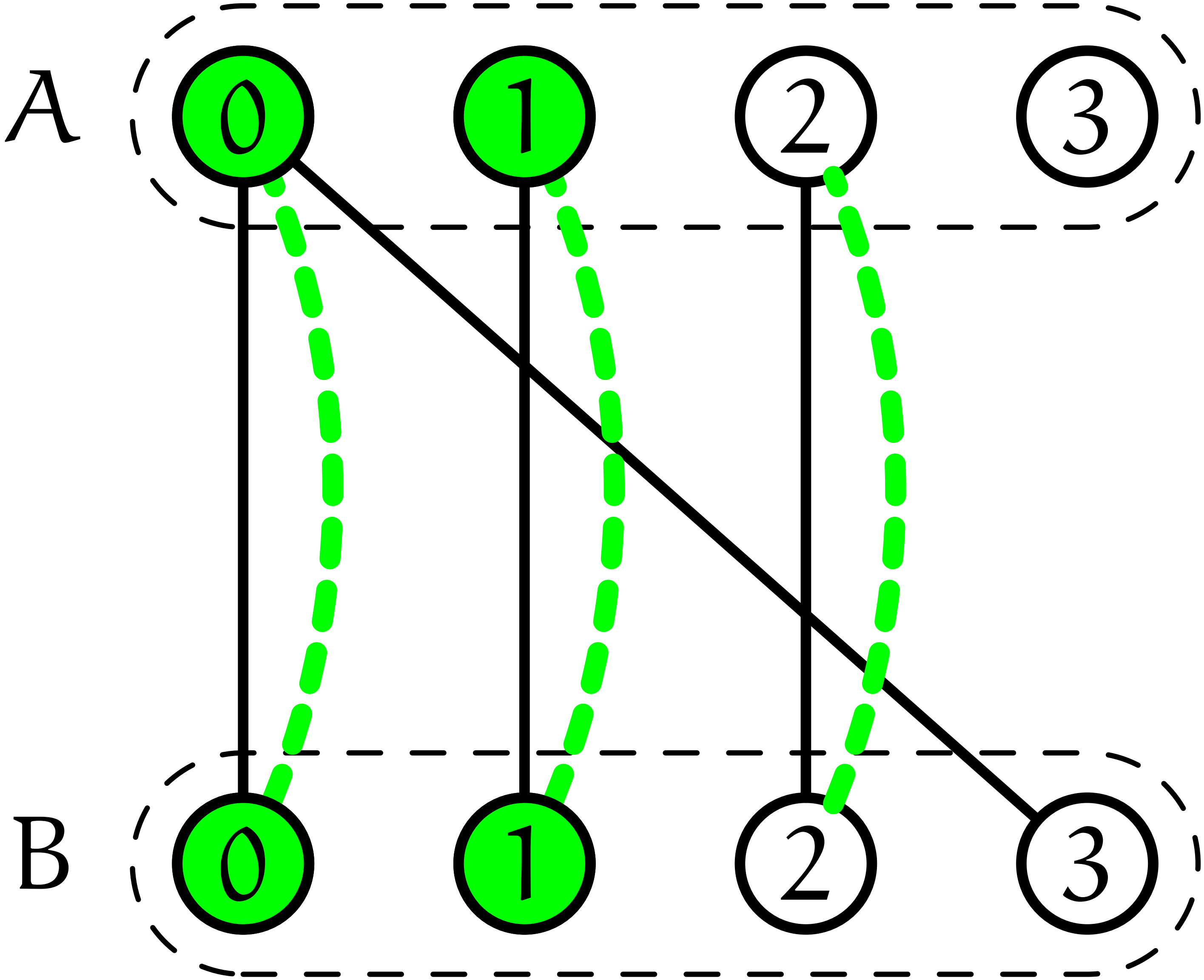
AC-3$_b$ #CC (1)



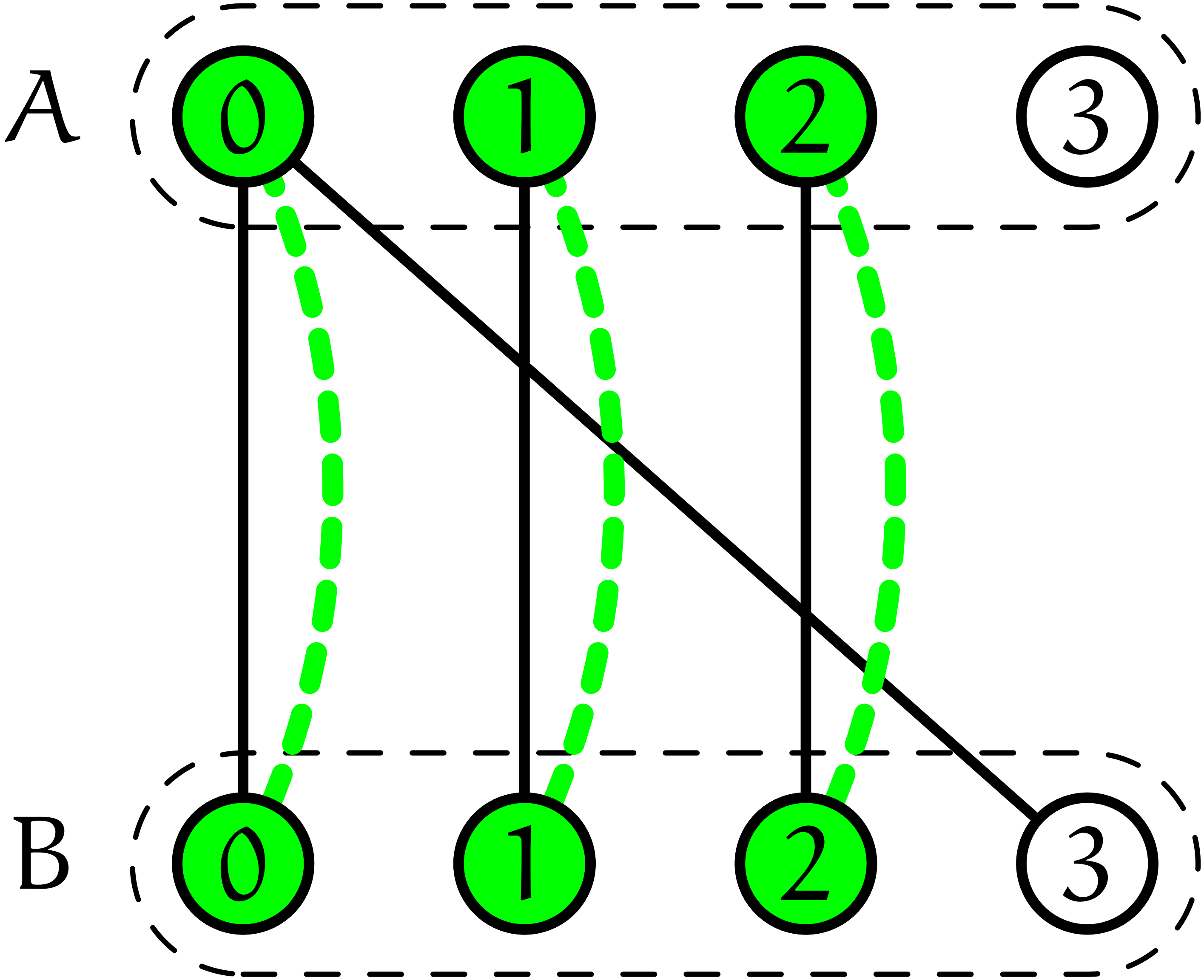
AC-3$_b$ #CC (2)



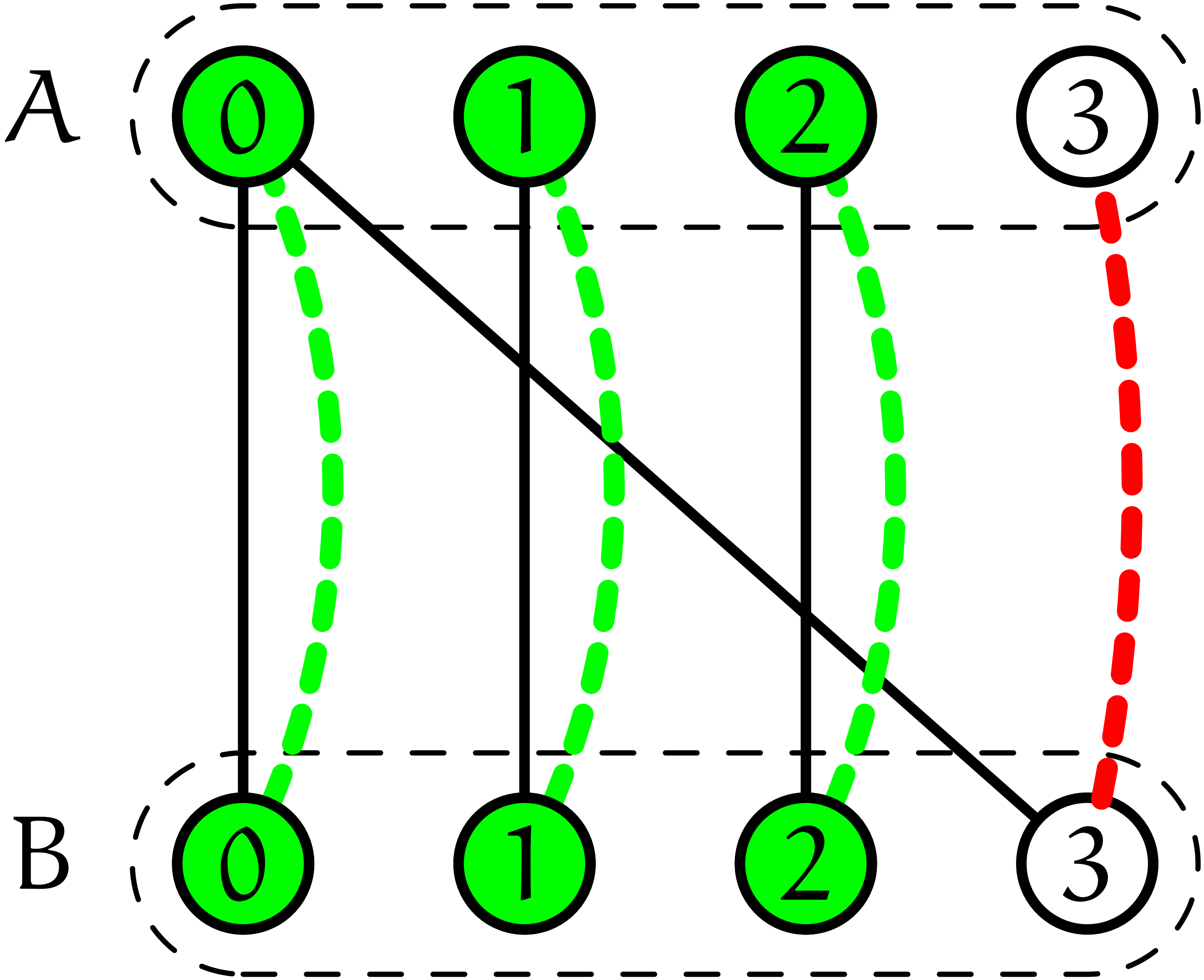
AC-3$_b$ #CC (2)



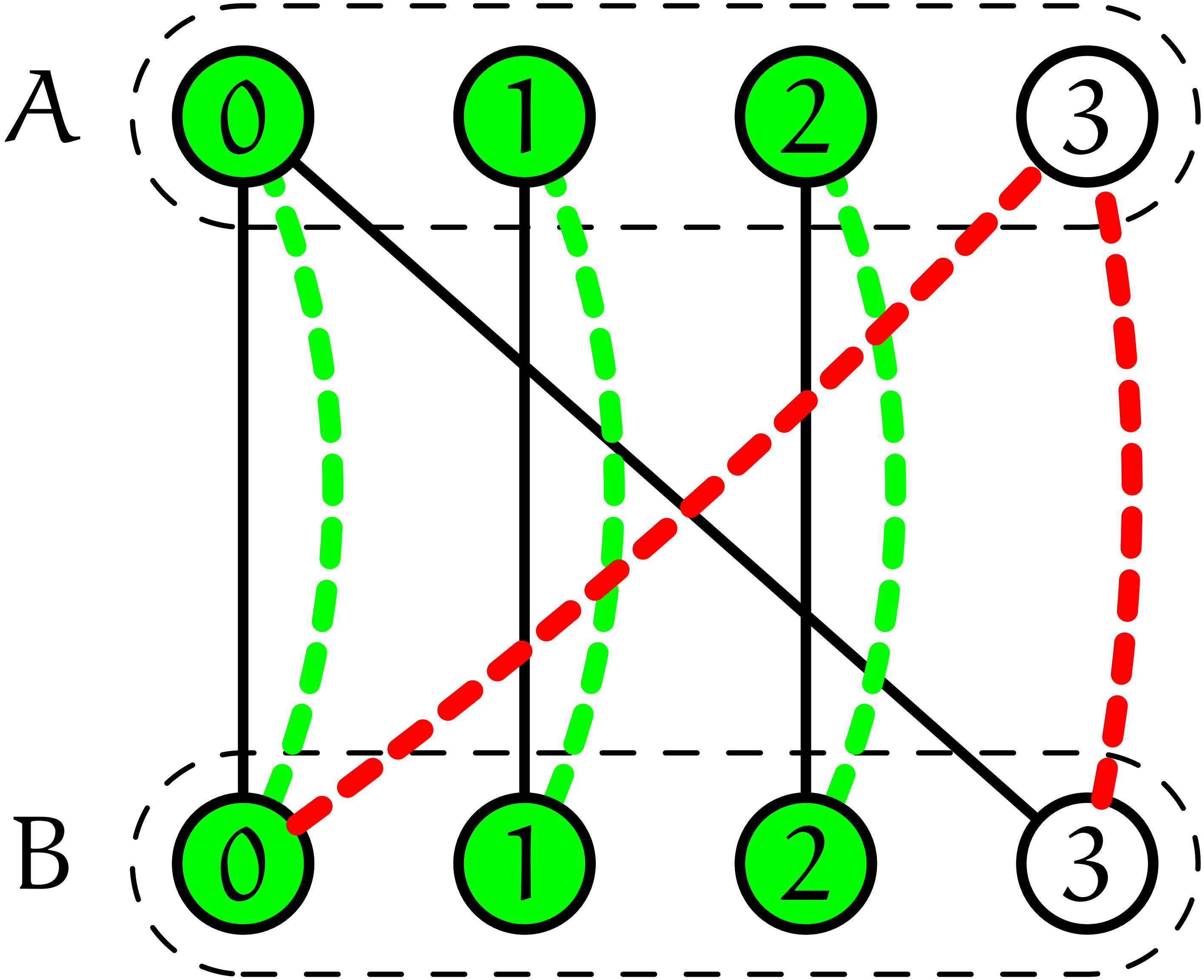
AC-3$_b$ #CC (3)



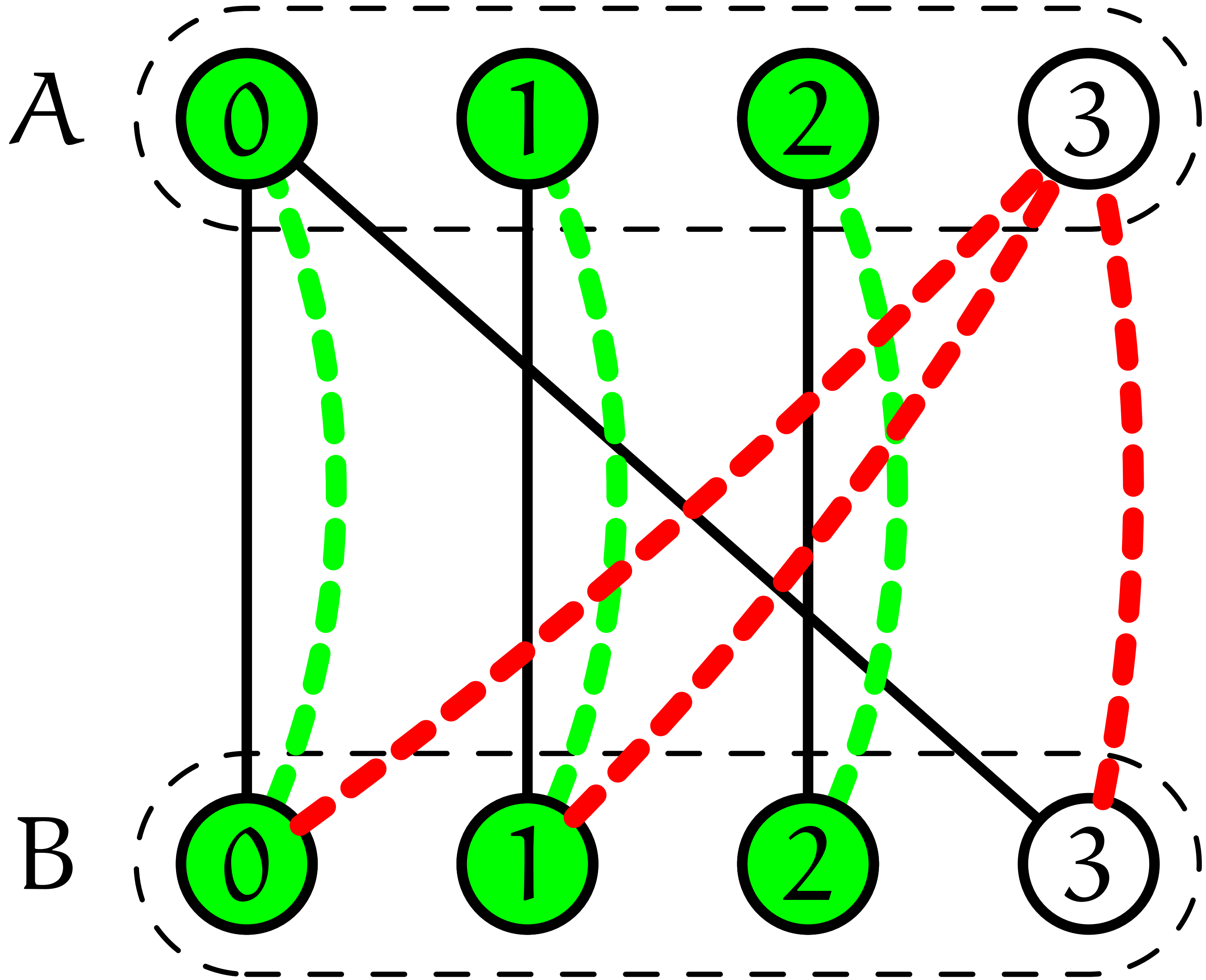
AC-3$_b$ #CC (3)



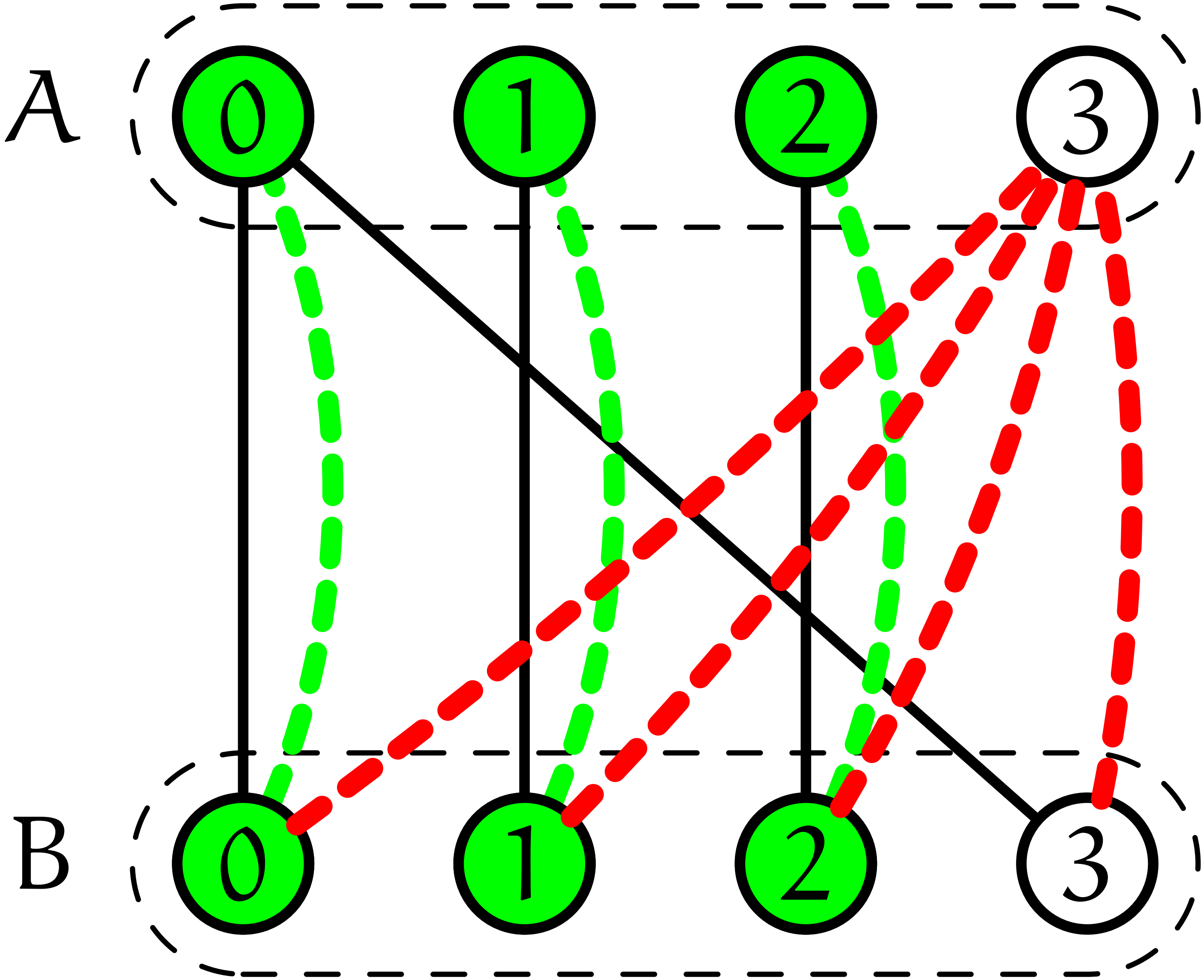
AC-3$_b$ #CC (4)



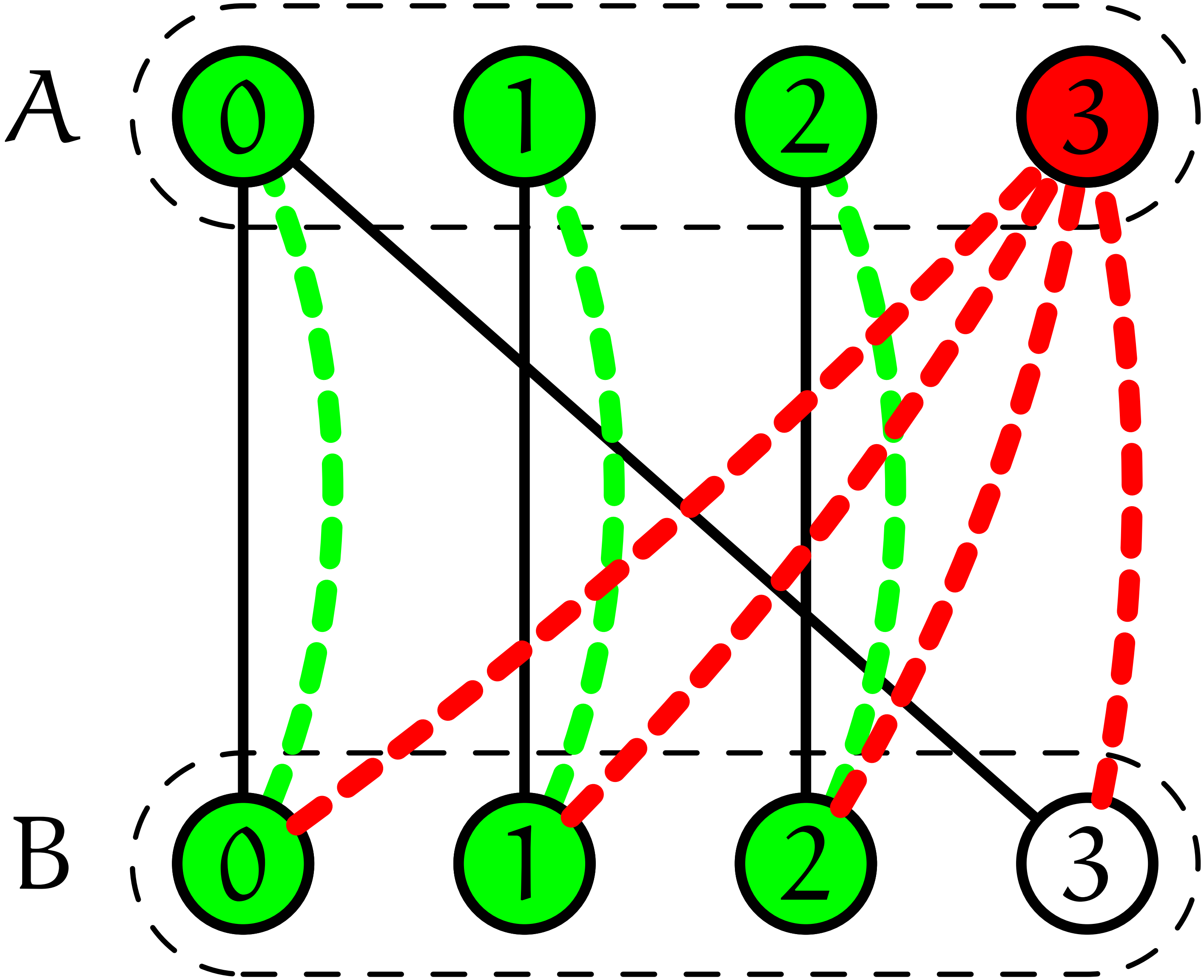
AC-3$_b$ #CC (5)



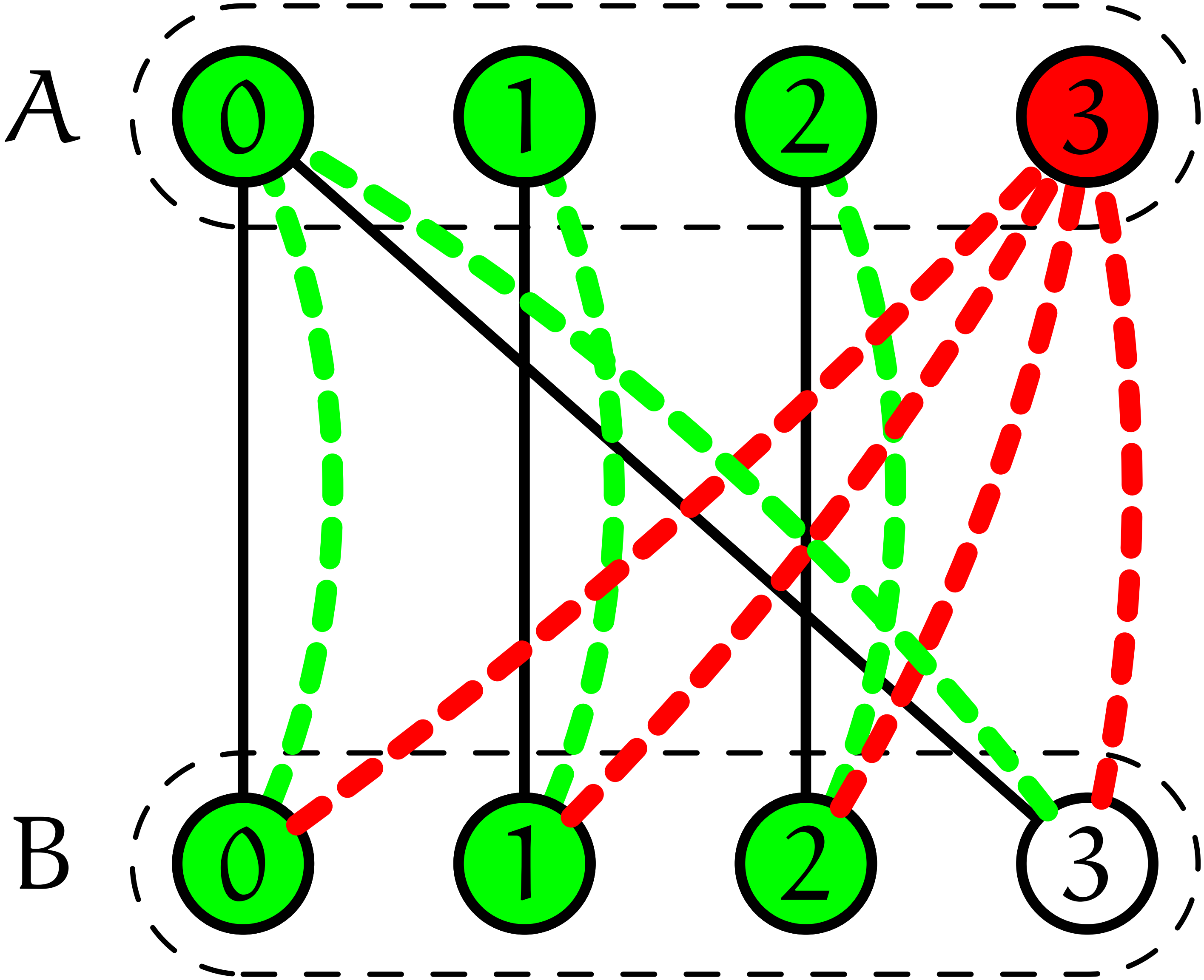
AC-3$_b$ #CC (6)



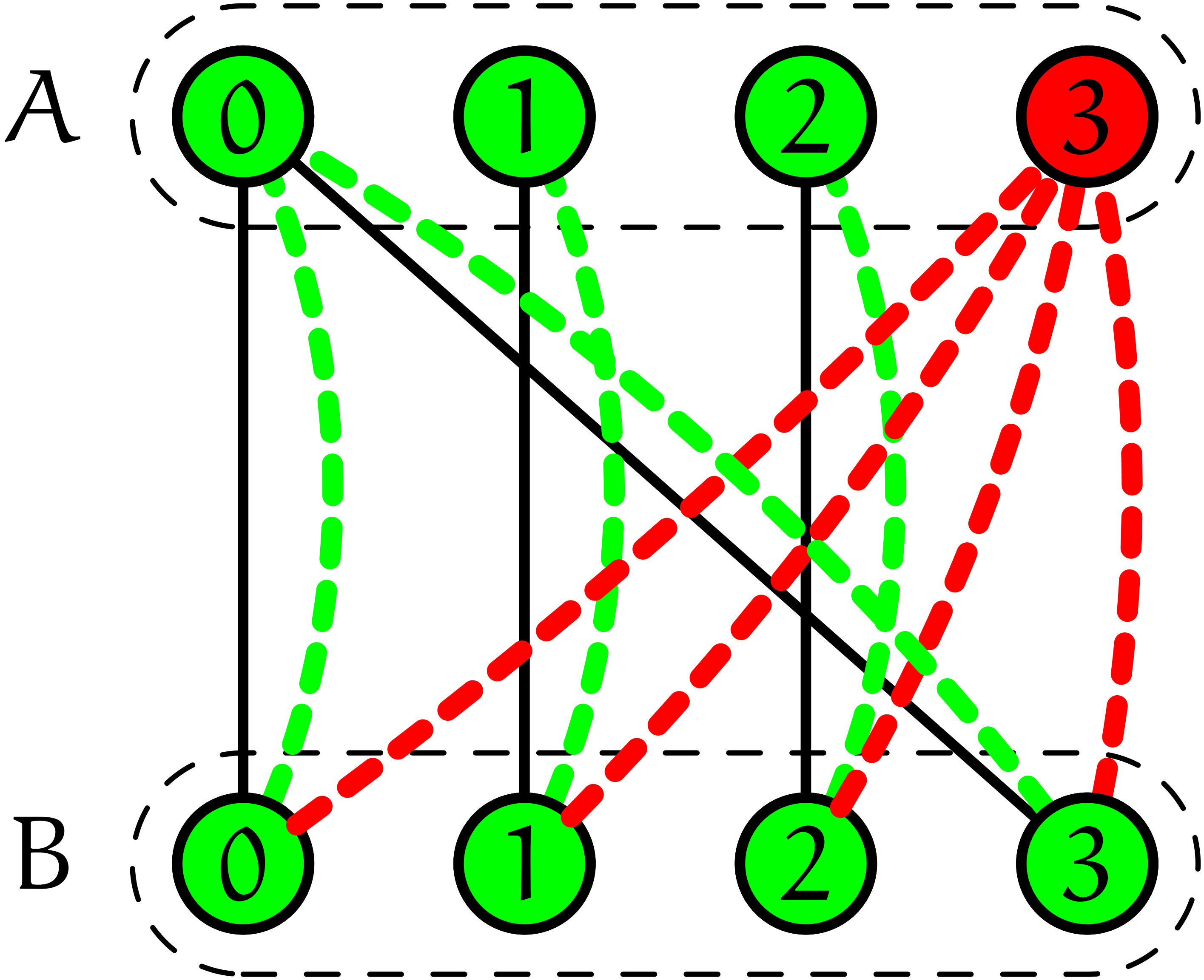
AC-3$_b$ #CC (7)



AC-3$_b$ #CC (7)



AC-3 b #CC (8)



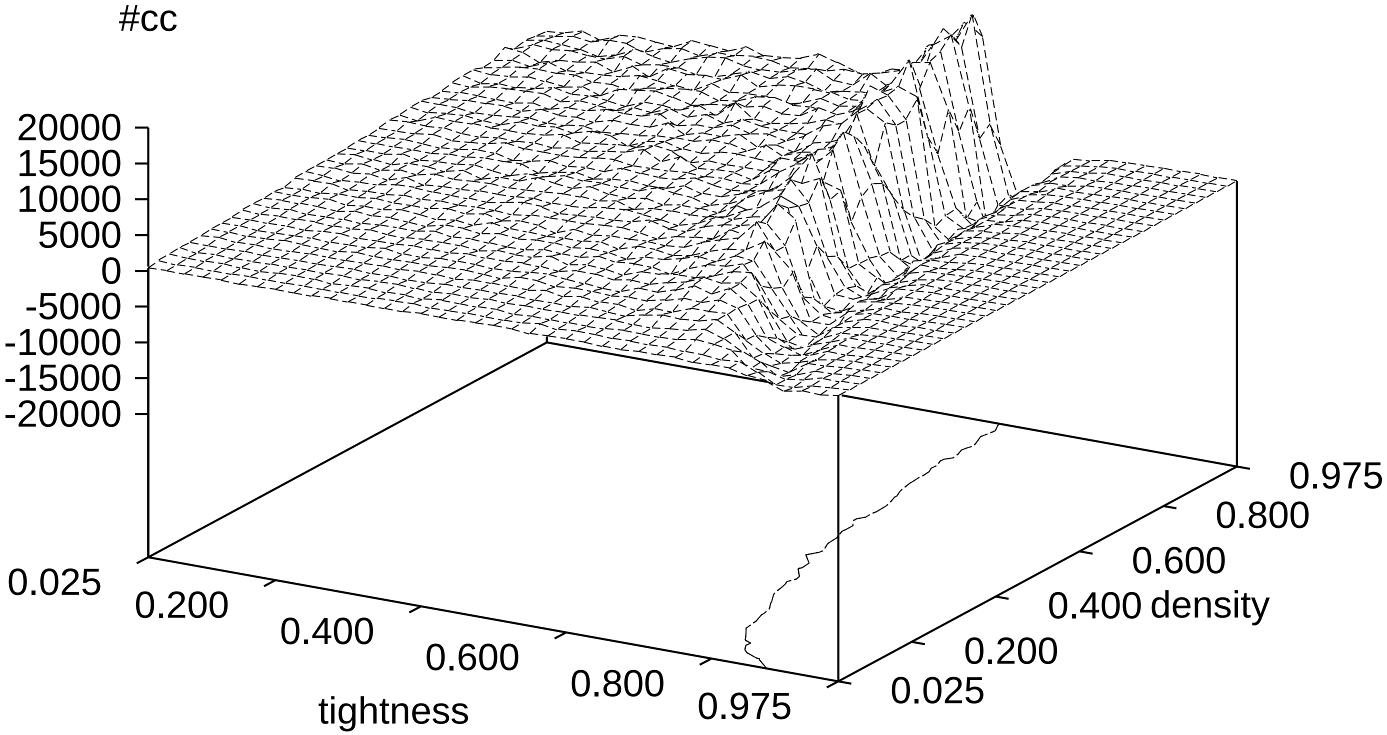
AC-3$_b$ #CC (8)

Experimental Results

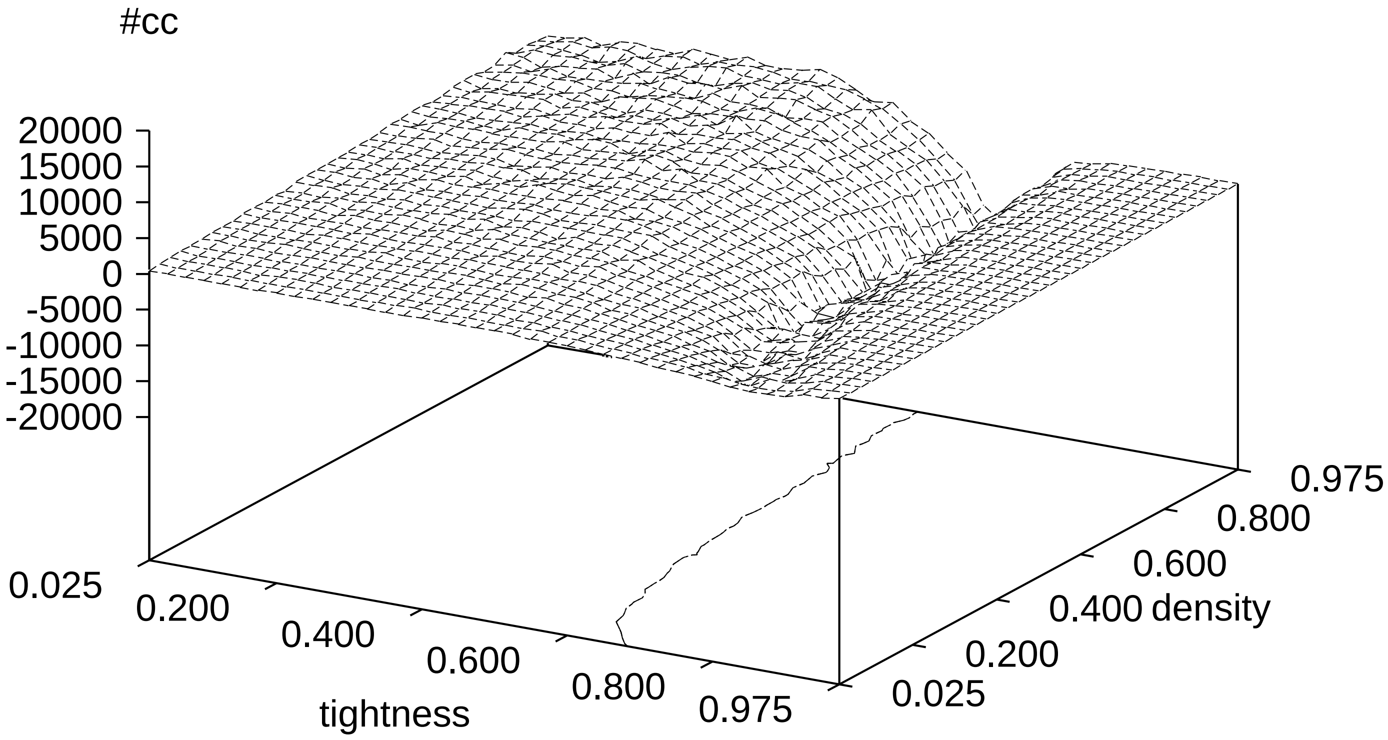
For each combination of (density,tightness) in $\{(d/40, t/40)|d \in \{1, 2, \ldots 39\}, t \in \{1, 2, \ldots 39\}\}$ 20 random connected CSPs were generated (30,420 in total).

	DEE	AC-3	AC-3_b	AC-7
$\#cc$	7311	7261	5077	5319

Average Number of Consistency-Checks



$\#\mathrm{cc}(\mathsf{AC\text{-}3}) - \#\mathrm{cc}(\mathsf{AC\text{-}3}_{\mathrm{b}})$



$\#\mathrm{cc}(\text{AC-7}) - \#\mathrm{cc}(\text{AC-3}_\mathrm{b})$

Discussion

- To minimise the number of consistency–checks, the number of successful double–support checks has to be maximised.
- For the problem set under consideration and the “usual” ordering heuristics AC-3_b outperforms AC-7.
- Trying to maximise the number of successful double-support checks seems to improve arc–consistency algorithms.
- Don’t be too eager!