User Guide of CABEAN Version 1.0

Cui Su

April 17, 2020

Contents

1 Introduction **2**

2 Installation Guide **2**
2.1 System Requirement 2
2.2 Installation 2

3 Syntax of the Input Files **2**
3.1 Model File 2
3.2 Specification Files 4

4 Execution of CABEAN with Command Line **4**
4.1 Command Options 4
4.2 Examples of CABEAN Command Lines 5

5 Case Studies **7**
5.1 Minimal One-step Temporary Source-target Control 7
5.2 Attractor-based Sequential Instantaneous Control 9

1 Introduction

CABEAN is a software tool for the control of large asynchronous Boolean networks, which are often used to model biological systems. The control of a Boolean network aims to identify effective perturbations that can drive the dynamics of the network from a source attractor to the desired attractor. Currently, CABEAN provides several control methods:

- the minimal one-step instantaneous source-target control (OI) [3, 4];
- the minimal one-step temporary source-target control (OT) [6];
- the minimal one-step permanent source-target control (OP) [6];
- the attractor-based sequential instantaneous source-target control (ASI) [1].
- the attractor-based sequential temporary source-target control (AST) [5].
- the attractor-based sequential permanent source-target control (ASP) [5].

2 Installation Guide

2.1 System Requirement

1. Platform: x86 compatible 32 bit or 64 bit processor;
2. Operating system: Windows, Linux and Mac OSX (10.4 and above);
3. Required compilers: flex 2.6.4 or higher, GNU bison 3.0.4 or higher, GNU g++ 4.0.1 or higher, and Windows subsystem for Linux (Windows platform only).

2.2 Installation

The executable binary of CABEAN can be downloaded at `http://satoss.uni.lu/software/CABEAN/`. With the required compiler, the binary of CABEAN can be executed directly with command line.

3 Syntax of the Input Files

CABEAN accepts three types of input files, including the model file, files for the specification of undesired attractors and undesired perturbations.

3.1 Model File

The model file is under ISPL syntax. We refer details of ISPL format to `https://vas.doc.ic.ac.uk/software/mcmas/manual.pdf`. Here we use an example to introduce the basics of ISPL format. The Boolean functions and attractors of the example are given in Table 1. The model file of the example is given in Figure 1. It contains two parts: 'Agent M' and 'InitStates'.

1. Section 'Agent M' defines the Boolean variables and their Boolean functions. It can be further split into four subsections:
   (a) subsection 'Vars' defines Boolean variables of the network;
   (b) subsections 'Actions' and 'Protocol' are mandatory but we define them as 'none' as shown in Figure 1;
   (c) subsection 'Evolution' defines Boolean functions of the variables.

nodes	functions	indices of attractors	attractors
x_1	$f_1 = x_2$	A_1	001100
x_2	$f_2 = x_1$	A_2	001111
x_3	$f_3 = (\sim x_2) \vee x_4$	A_3	110000
x_4	$f_4 = x_3$	A_4	110010
x_5	$f_5 = (x_1 \vee x_6) \wedge x_5$	A_5	111100
x_6	$f_6 = x_4 \wedge x_5$	A_6	111111

Table 1: Boolean functions and attractors of the example.

```
Agent M
   Vars:
      x1: boolean;
      x2: boolean;
      x3: boolean;
      x4: boolean;
      x5: boolean;
      x6: boolean;
   end Vars
   Actions = none;
   Protocol:
      Other: none;
   end Protocol
   Evolution:
      x1=true  if x2=true;
      x1=false if x2=false;
      x2=true  if x1=true;
      x2=false if x1=false;
      x3=true  if (~ x2)|x4=true;
      x3=false if (~ x2)|x4=false;
      x4=true  if x3=true;
      x4=false if x3=false;
      x5=true  if (x1|x6)&x5=true;
      x5=false if (x1|x6)&x5=false;
      x6=true  if x4&x5=true;
      x6=false if x4&x5=false;
   end Evolution
end Agent

InitStates
      M.x1=true or M.x1=false;
end InitStates
```

Figure 1: The model file of the example.

2. Section 'InitStates' defines the initial state. A prefix "M." is added to each node x to indicate that x is a variable defined in agent M. Note that the initialisation of an input node in "InitStates" will shrink the state space to the states, in which the input node has the initial value. Thus, it will influence the attractors and the control results. However, the initialisation of a non-input node has no influence on the attractors or the control results.

3.2 Specification Files

CABEAN integrates practical constraints on the attractors and perturbations during the computation. User can define a list of undesired attractors that can not play the role of source, target or intermediate attractors. The indices of undesired attractors are specified in a text file and the indices are separated by comma. Suppose attractors A_2 and A_5 are undesired attractors. The specification file is shown below.

```
2, 5
```

User can define three types of undesired perturbations:

- `R0`: nodes that can not be perturbed from 'off' to 'on';
- `R1`: nodes that can not be perturbed from 'on' to 'off';
- `R`: nodes that can not be perturbed in either way.

For each type, the nodes are separated by comma as shown below.

```
R0:
R1:  x5
R: x4, x6
```

4 Execution of CABEAN with Command Line

4.1 Command Options

We summarise the command options as follows.

- `-steadystates`: the mandatory option.
- `-asynbn`: the mandatory option.
- `-newtarjan`: the mandatory option.
- `-newpred`: the mandatory option.
- `-compositional <number>`: choose the method for attractor detection and basin computation. Only options '0' and '2' are supported in the current version. Option '0' is the default option and it indicates that the global methods for the computation of attractors [2] and basins [3] are selected. Option '2' means the decomposition-based computation of attractors [2] and basins [3] will be used.
- `-control <control method>`: choose one of the control methods:
  - `OI`: the minimal one-step instantaneous source-target control;
  - `OT`: the minimal one-step temporary source-target control;
  - `OP`: the minimal one-step permanent source-target control;
  - `ASI`: the attractor-based sequential instantaneous source-target control;
  - `AST`: the attractor-based sequential temporary source-target control;
  - `ASP`: the attractor-based sequential permanent source-target control.

- `-sin <number> -tin <number>`: specify the indices of the source and target attractors and only compute the results for the specified source and target attractors.
- `-allpairs`: for source-target control methods, compute the results for all combinations of source and target attractors.
- `-rmID <file name>`: set the indices of undesired attractor. The cases where the source or the target is undesired will be skipped. The undesired attractors will also be avoided as intermediate attractors for ASI control.
- `-rmPert <file name>`: set undesired perturbations.
- `<model file>`: the name of the model file is always placed at the end of the command line.

4.2 Examples of CABEAN Command Lines

In this section, we summarise some general command lines. We use '<control>' to denote one of the control methods (OI, OT, OP, ASI, AST and ASP).

1. Computation of the control for one pair of source and target attractors:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -sin <index of the source> -tin <index of the target> <model file>
2. Computation of the control for one pair of source and target attractors with constraints on perturbations:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -sin <index of the source> -tin <index of the target> -rmPert <file name> <model file>
3. Computation of the control for one pair of source and target attractors with constraints on attractors:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -sin <index of the source> -tin <index of the target> -rmID <file name> <model file>
4. Computation of the control for one pair of source and target attractors with constraints on attractors and perturbations:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -sin <index of the source> -tin <index of the target> -rmID <file name1> -rmPert <file name2> <model file>
5. Computation of the control for all combinations of source and target attractors:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -allpairs <model file>
6. Computation of the control for all combinations of source and target attractors with constraints on perturbations:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -allpairs -rmPert <file name> <model file>
7. Computation of the control for all combinations of source and target attractors with constraints on attractors:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -allpairs -rmID <file name> <model file>

   Note that for ASI control, the undesired attractors specified will be avoided as source, target and intermediate attractors; for OI, OT and OP control, the cases where undesired attractors are the source or the target will be skipped.

8. Computation of the control for all combinations of source and target attractors with constraints on attractors and perturbations:
./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control <control> -allpairs -rmID <file name1> -rmPert <file name2> <model file>

   Similar to the previous command line, for all the control methods, the cases where undesired attractors are the source or the target will be skipped; and for ASI control, the undesired attractors will also be avoided as intermediate attractors.

```
 Command line:  ./cabean -steadystates -asynbn -newtarjan -newpred
-compositional 2 -control OT -sin 6 -tin 1 example.ispl
====== find attractor #1 :  1 states ======
:  7 nodes 1 leaves 1 minterms
0-0-1-1-0-0- 1
====== find attractor #2 :  1 states ======
:  7 nodes 1 leaves 1 minterms
0-0-1-1-1-1- 1
====== find attractor #3 :  1 states ======
:  7 nodes 1 leaves 1 minterms
1-1-0-0-0-0- 1
====== find attractor #4 :  1 states ======
:  7 nodes 1 leaves 1 minterms
1-1-0-0-1-0- 1
====== find attractor #5 :  1 states ======
:  7 nodes 1 leaves 1 minterms
1-1-1-1-0-0- 1
====== find attractor #6 :  1 states ======
:  7 nodes 1 leaves 1 minterms
1-1-1-1-1-1- 1
number of attractors = 6
time for attractor detection=0.002 seconds
=== ONE-STEP TEMPORARY SOURCE-TARGET CONTROL (DECOMP) ===
source - 6 target - 1
PATH 1 - #perturbations:  2
Control set:  x2=0 x6=0
PATH 2 - #perturbations:  2
Control set:  x2=0 x5=0
PATH 3 - #perturbations:  2
Control set:  x2=0 x4=0
PATH 4 - #perturbations:  2
Control set:  x2=0 x3=0
PATH 5 - #perturbations:  2
Control set:  x1=0 x6=0
PATH 6 - #perturbations:  2
Control set:  x1=0 x5=0
PATH 7 - #perturbations:  2
Control set:  x1=0 x4=0
PATH 8 - #perturbations:  2
Control set:  x1=0 x3=0
execution time for control = 0.005 seconds
```

Figure 2: Computation of the minimal OT control from A_6 to A_1.

5 Case Studies

In this section, we use the example introduced in Section 3 to illustrate the basic usages of CABEAN. The model file is save as 'example.ispl'. This network has in total six attractors as shown in Table 1.

5.1 Minimal One-step Temporary Source-target Control

In this subsection, we compute the minimal OT control for (1) one pair of source and target attractors and (2) all combinations of source and target attractors.

1. **Minimal OT control from A_6 to A_1.**
   Let us assume the source and target attractors are A_6 and A_1, respectively. All the minimal OT control sets from A_6 to A_1 are computed using the following command line:
   "./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control OT -sin 6 -tin 1 example.ispl"
   The results are printed out in the terminal as shown in Figure 2. CABEAN first identifies all the exact attractors of the network, print them in lexicographical order and then compute all the minimal OT control. We can see that there are in total eight minimal OT control sets and they require at least two perturbations. Henceforth, we will omit the attractor part in the output.

2. **Minimal OT control from A_6 to A_1 with constraints on perturbations.**
   Suppose the node x_5 can not be perturbed from 1 to 0; the nodes x_4 and x_6 can not be perturbed either from 1 to 0 or the opposite. The file for specifying the constraints, 'rmPert.txt', has been given in Section 3.2. We compute the minimal OT control under these constraints using the following command:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control OT -sin 6 -tin 1 -rmPert rmPert.txt example.ispl
   The output is given in Figure 3. Compared to the results in Figure 2, we can see that CABEAN only computes the control sets without undesired perturbations given in the file 'rmPert.txt'.

```
 Command line:  ./cabean -steadystates -asynbn -newtarjan -newpred
-compositional 2 -control OT -sin 6 -tin 1 -rmPert rmPert.txt example.ispl

... omit the attractors ...

=== ONE-STEP TEMPORARY SOURCE-TARGET CONTROL (DECOMP) ===
source - 6 target - 1
PATH 1 - #perturbations:  2
Control set:  x2=0 x3=0
PATH 2 - #perturbations:  2
Control set:  x1=0 x3=0
execution time for control = 0.002 seconds
```

Figure 3: Computation of minimal OT from A_6 to A_1 with constraints on perturbations.

3. **Minimal OT control for all pairs of source and target attractors.**
   We use the following commands to compute the minimal OT control for all combinations of source and target attractors:

./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control OT -allpairs example.ispl'

The output of CABEAN is shown in Figure 4. CABEAN first performs attractor detection and then compute the minimal OT control for each pair of source and target attractors. After traversing all combinations, the execution time and the number of perturbations for each case are summarised.

```
 Command line:  ./cabean -steadystates -asynbn -newtarjan -newpred
-compositional 2 -control OT -allpairs example.ispl
... omit the attractors ...
=== ONE-STEP TEMPORARY SOURCE-TARGET CONTROL (DECOMP) ===
---------------------------------------
source - 1 target - 2
PATH 1 - #perturbations:  1
Control set:  x5=1
execution time for control = 0.001 seconds
-----------------------------------------
source - 1 target - 3
PATH 1 - #perturbations:  2
Control set:  x2=1 x4=0
PATH 2 - #perturbations:  2
Control set:  x2=1 x3=0
PATH 3 - #perturbations:  2
Control set:  x1=1 x4=0
PATH 4 - #perturbations:  2
Control set:  x1=1 x3=0
execution time for control = 0.002 seconds
... we omit some results ...
-----------------------------------------
source - 6 target - 5
PATH 1 - #perturbations:  1
Control set:  x5=0
execution time for control = 0.001 seconds
**********************************
print all the time costs:
0 0.001 0.002 0.002 0.001 0.001
0.001 0 0.003 0.002 0.001 0.001
0.001 0.002 0 0.001 0.001 0.001
0.005 0.004 0.001 0 0.001 0.001
0.002 0.001 0.001 0.001 0 0
0.003 0.001 0.001 0.001 0.001 0
print the number of driver nodes:
0 1 2 3 1 2
1 0 3 2 2 1
1 2 0 1 1 2
2 2 1 0 2 1
1 2 1 2 0 1
2 1 2 1 1 0
total time costs=0.046 seconds
```

Figure 4: Computation of the minimal OT control for all combinations of source and target attractors.

4. **Minimal OT control for all pairs of source and target attractors with constraints on attractors**
   Given some undesired attractors, the cases where the source and/or the target is undesired can be skipped by adding '-rmID ¡file name¿' to the command line. Suppose the undesired attractors are A_2 and A_5, the specification file ('rmID.txt') has been given in Section 3.2. The command line is:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control OT -allpairs -rmID rmID.txt example.ispl'
   All pairs of source and target attractors except for the cases, where the source or the target is A_2 or A_5, are computed.

In the command line, by replacing 'OT' with 'OI', 'OP', 'ASI', we can compute the minimal OI, OP, and ASI control for either one pair or all pairs of source and target attractors.

5.2 Attractor-based Sequential Instantaneous Control

In this subsection, we illustrate the computation of ASI control from A_6 to A_1 with and without constraints on the attractors. In the current version, the ASI control computes all the attractor-based sequential instantaneous paths with at most K perturbations and we use the minimal number of perturbations required by OI control as the threshold K.

1. **ASI control from A_6 to A_1.** The computation of ASI control without any constraints can be performed using the following command:
   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control ASI -sin 6 -tin 1 example.ispl'
   The output is given in Figure 5.

```
 Command line:  ./cabean -steadystates -asynbn -newtarjan -newpred
-compositional 2 -control ASI -sin 6 -tin 1 example.ispl
... omit the attractors ...
=== ATTRACTOR-BASED SEQUENTIAL INSTANTANEOUS SOURCE-TARGET CONTROL (DECOMP)
===
source - 6 target - 1
PATH 1 - #perturbations:  3
Sequence of the attractors:  6 -> 1
   STEP 1
     Control set 1:  x1=0 x2=0 x5=0
PATH 2 - #perturbations:  3
Sequence of the attractors:  6 -> 2 -> 1
   STEP 1
     Control set 1:  x1=0 x2=0
   STEP 2
     Control set 1:  x5=0
PATH 3 - #perturbations:  3
Sequence of the attractors:  6 -> 5 -> 1
   STEP 1
     Control set 1:  x5=0
   STEP 2
     Control set 1:  x1=0 x2=0
execution time of control=0.004 seconds
```

Figure 5: Computation of the ASI control from A_6 to A_1.

2. **ASI control from A_6 to A_1 with constraints on perturbations.** Suppose the node x_5 can not be perturbed from 1 to 0; the nodes x_4 and x_6 can not be perturbed either from 1 to 0 or the opposite. We can compute ASI control without undesired perturbations using the following command line: ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control ASI -sin 6 -tin 1 -rmPert rmPert.txt example.ispl'
   The results are given in Figure 6. We can see that there does not exist any one-step instantaneous control paths satisfying the constraints and thus the threshold is set to 0.

```
 Command line:  ./cabean -steadystates -asynbn -newtarjan -newpred
-compositional 2 -control ASI -sin 6 -tin 1 -rmPert rmPert.txt example.ispl
... omit the attractors ...
=== ATTRACTOR-BASED SEQUENTIAL INSTANTANEOUS SOURCE-TARGET CONTROL (DECOMP)
===
source - 6 target - 1
There does not exist any direct path satisfying the constraints.
The threshold on the number of perturbations is 0.
```

Figure 6: Computation of the ASI control from A_6 to A_1 with constraints on perturbations.

3. **ASI control from A_6 to A_1 with constraints on attractors.** Suppose attractors A_2 and A_5 are undesired attractors as defined in Section 3.2 and the file is saved as 'rmID.txt'. We compute the ASI control bypassing undesired attractors using the following command:

   ./cabean -steadystates -asynbn -newtarjan -newpred -compositional 2 -control ASI -sin 6 -tin 1 -rmID rmID.txt example.ispl

   The output is given in Figure 7.

```
 Command line:  ./cabean -steadystates -asynbn -newtarjan -newpred
-compositional 2 -control ASI -sin 6 -tin 1 -rmID rmID.txt example.ispl
... omit the attractors ...
=== ONE-STEP TEMPORARY SOURCE-TARGET CONTROL (DECOMP) ===
IDs of undesired attractors
2 5
-----------------------------------------
source - 6 target - 1
PATH 1 - #perturbations:  3
Sequence of the attractors:  6 -> 1
   STEP 1
     Control set 1:  x1=0 x2=0 x5=0
execution time of control=0.002 seconds
```

Figure 7: Computation of ASI from A_6 to A_1 with constraints on attractors.

References

[1] Hugues Mandon, Cui Su, Stefan Haar, Jun Pang, and Loïc Paulevé. Sequential reprogramming of Boolean networks made practical. In *Proc. 17th International Conference on Computational Methods in Systems Biology*, volume 11773 of *LNCS*, pages 3–19. Springer, 2019.

[2] A. Mizera, J. Pang, H. Qu, and Q. Yuan. Taming asynchrony for attractor detection in large Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1):31–42, 2019.

[3] S. Paul, C. Su, J. Pang, and A. Mizera. A decomposition-based approach towards the control of Boolean networks. In *Proc. 9th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 11–20. ACM Press, 2018.

[4] Soumya Paul, Cui Su, Jun Pang, and Andrzej Mizera. An efficient approach towards the source-target control of Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019. accepted.

[5] Cui Su and Jun Pang. Sequential control of Boolean networks with temporary and permanent perturbations. *ArXiv e-prints*, 2020. https://arxiv.org/abs/2004.07184.

[6] Cui Su, Soumya Paul, and Jun Pang. Controlling large Boolean networks with temporary and permanent perturbations. In *Proc. 23rd International Symposium on Formal Methods*, volume 11800 of *LNCS*, pages 707–724. Springer-Verlag, 2019.