

## The Gerardium Rush

Generated by Doxygen 1.8.17

<b>1 LICENSE</b>	<b>1</b>
<b>2 Gormanium rush</b>	<b>2</b>
2.1 Installation	2
2.2 File generation	2
2.3 Macros	2
2.3.1 PARALLEL_ENABLE	2
2.3.2 OUT_RESULT_TO_FILE	2
2.3.3 PERFORMANCE_ANALYSIS_MACRO	3
2.4 Test	3
2.5 Post-processing	3
2.6 Documentation	3
2.7 License	3
<b>3 Namespace Index</b>	<b>3</b>
3.1 Namespace List	3
<b>4 Class Index</b>	<b>3</b>
4.1 Class List	3
<b>5 File Index</b>	<b>4</b>
5.1 File List	4
<b>6 Namespace Documentation</b>	<b>5</b>
6.1 graph Namespace Reference	5
6.1.1 Function Documentation	5
6.1.2 Variable Documentation	6
6.2 graph_online Namespace Reference	7
6.2.1 Function Documentation	8
6.2.2 Variable Documentation	8
<b>7 Class Documentation</b>	<b>10</b>
7.1 Algorithm_Parameters Struct Reference	10
7.1.1 Detailed Description	10
7.1.2 Member Data Documentation	11
7.2 CCircuit Class Reference	12
7.2.1 Detailed Description	13
7.2.2 Constructor & Destructor Documentation	13
7.2.3 Member Function Documentation	13
7.3 CSimulator Class Reference	14
7.3.1 Constructor & Destructor Documentation	15
7.3.2 Member Function Documentation	15
7.4 CUnit Class Reference	17
7.4.1 Detailed Description	17

7.4.2 Constructor & Destructor Documentation . . . . .	18
7.4.3 Member Function Documentation . . . . .	18
7.4.4 Member Data Documentation . . . . .	18
7.5 GA Class Reference . . . . .	20
7.5.1 Constructor & Destructor Documentation . . . . .	22
7.5.2 Member Function Documentation . . . . .	23
7.5.3 Member Data Documentation . . . . .	28
7.6 SimulationParameters Struct Reference . . . . .	31
7.6.1 Detailed Description . . . . .	31
7.6.2 Member Data Documentation . . . . .	31
<b>8 File Documentation</b>	<b>33</b>
8.1 build/CMakeCache.txt File Reference . . . . .	33
8.1.1 Variable Documentation . . . . .	33
8.2 build/CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c File Reference . . . . .	33
8.2.1 Macro Definition Documentation . . . . .	34
8.2.2 Function Documentation . . . . .	35
8.2.3 Variable Documentation . . . . .	35
8.3 CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c File Reference . . . . .	35
8.3.1 Macro Definition Documentation . . . . .	36
8.3.2 Function Documentation . . . . .	37
8.3.3 Variable Documentation . . . . .	37
8.4 build/CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference . . . . .	38
8.4.1 Macro Definition Documentation . . . . .	38
8.4.2 Function Documentation . . . . .	39
8.4.3 Variable Documentation . . . . .	39
8.5 CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference . . . . .	40
8.5.1 Macro Definition Documentation . . . . .	40
8.5.2 Function Documentation . . . . .	41
8.5.3 Variable Documentation . . . . .	42
8.6 build/CMakeFiles/CMakeRuleHashes.txt File Reference . . . . .	42
8.7 CMakeFiles/CMakeRuleHashes.txt File Reference . . . . .	42
8.8 build/CMakeFiles/FindMPI/test_mpi.cpp File Reference . . . . .	42
8.8.1 Function Documentation . . . . .	43
8.9 CMakeFiles/FindMPI/test_mpi.cpp File Reference . . . . .	43
8.9.1 Function Documentation . . . . .	43
8.10 build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference . . . . .	44
8.10.1 Function Documentation . . . . .	44
8.10.2 Variable Documentation . . . . .	44
8.11 CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference . . . . .	45
8.11.1 Function Documentation . . . . .	45
8.11.2 Variable Documentation . . . . .	45

8.12 build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference . . . . .	46
8.12.1 Function Documentation . . . . .	46
8.12.2 Variable Documentation . . . . .	46
8.13 CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference . . . . .	47
8.13.1 Function Documentation . . . . .	47
8.13.2 Variable Documentation . . . . .	47
8.14 build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c File Reference . . . . .	48
8.14.1 Function Documentation . . . . .	48
8.15 CMakeFiles/FindOpenMP/OpenMPTryFlag.c File Reference . . . . .	48
8.15.1 Function Documentation . . . . .	49
8.16 build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp File Reference . . . . .	49
8.16.1 Function Documentation . . . . .	49
8.17 CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp File Reference . . . . .	50
8.17.1 Function Documentation . . . . .	50
8.18 build/CMakeFiles/TargetDirectories.txt File Reference . . . . .	50
8.19 CMakeFiles/TargetDirectories.txt File Reference . . . . .	50
8.20 build/DartConfiguration.tcl File Reference . . . . .	50
8.21 build/src/CMakeFiles/Circuit_Optimizer.dir/link.txt File Reference . . . . .	50
8.21.1 Variable Documentation . . . . .	50
8.22 build/src/CMakeFiles/circuitSimulator.dir/link.txt File Reference . . . . .	51
8.23 build/src/CMakeFiles/geneticAlgorithm.dir/link.txt File Reference . . . . .	51
8.24 build/tests/CMakeFiles/simulator_tests.dir/link.txt File Reference . . . . .	51
8.25 build/tests/CMakeFiles/test_GA.dir/link.txt File Reference . . . . .	51
8.25.1 Variable Documentation . . . . .	51
8.26 build/tests/CMakeFiles/test_simulator.dir/link.txt File Reference . . . . .	51
8.26.1 Variable Documentation . . . . .	51
8.27 build/tests/CMakeFiles/test_validity.dir/link.txt File Reference . . . . .	52
8.27.1 Variable Documentation . . . . .	52
8.28 src/CMakeFiles/Circuit_Optimizer.dir/link.txt File Reference . . . . .	52
8.28.1 Variable Documentation . . . . .	52
8.29 src/CMakeFiles/circuitSimulator.dir/link.txt File Reference . . . . .	53
8.30 src/CMakeFiles/geneticAlgorithm.dir/link.txt File Reference . . . . .	53
8.31 tests/CMakeFiles/simulator_tests.dir/link.txt File Reference . . . . .	53
8.32 tests/CMakeFiles/test_simulator.dir/link.txt File Reference . . . . .	53
8.33 tests/CMakeFiles/test_validity.dir/link.txt File Reference . . . . .	53
8.34 build/Testing/Temporary/CTestCostData.txt File Reference . . . . .	53
8.35 Testing/Temporary/CTestCostData.txt File Reference . . . . .	53
8.36 CMakeLists.txt File Reference . . . . .	53
8.37 src/CMakeLists.txt File Reference . . . . .	53
8.37.1 Function Documentation . . . . .	53
8.38 tests/CMakeLists.txt File Reference . . . . .	53
8.38.1 Function Documentation . . . . .	53

8.39 include/CCircuit.h File Reference . . . . .	54
8.40 include/CSimulator.h File Reference . . . . .	54
8.40.1 Function Documentation . . . . .	55
8.41 include/CUnit.h File Reference . . . . .	56
8.42 include/Genetic_Algorithm.h File Reference . . . . .	57
8.43 LICENSE.md File Reference . . . . .	58
8.44 README.md File Reference . . . . .	58
8.45 src/CCircuit.cpp File Reference . . . . .	58
8.46 src/CSimulator.cpp File Reference . . . . .	59
8.46.1 Detailed Description . . . . .	60
8.46.2 Function Documentation . . . . .	60
8.47 src/Genetic_Algorithm.cpp File Reference . . . . .	61
8.47.1 Function Documentation . . . . .	62
8.48 src/graph.py File Reference . . . . .	63
8.49 src/graph_online.py File Reference . . . . .	64
8.50 src/main.cpp File Reference . . . . .	64
8.50.1 Function Documentation . . . . .	65
8.51 tests/test_GA.cpp File Reference . . . . .	65
8.51.1 Function Documentation . . . . .	66
8.52 tests/test_simulator.cpp File Reference . . . . .	67
8.52.1 Function Documentation . . . . .	67
8.53 tests/test_validity.cpp File Reference . . . . .	69
8.53.1 Function Documentation . . . . .	69
<b>Index</b>	<b>71</b>

## 1 LICENSE

MIT License

Copyright (c) 2023 Linear-Regression Group

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 2 Gormanium rush

"The Gerardium Rush" repository is dedicated to the implementation of a Genetic Algorithm (GA) optimized for the task of maximizing the recovery of a valuable mineral named "gerardium" through the process of separation. The GA simulates the operation of separation units such as flotation cells or spirals, which are arranged in circuits to extract valuable material while also separating waste. The challenge lies in designing these circuits for optimal recovery and purity of gerardium, balancing the trade-off between these two factors based on economic considerations. The repository confines its scope to two product streams - a valuable concentrate stream and a waste-dominated tailings stream. The success of the separation circuit is evaluated based on the purity and weight of the gerardium in the concentrate stream. Given the vast number of potential circuit configurations, the genetic algorithm is leveraged to find an optimal solution, proving its effectiveness in tackling discrete optimization problems.

### 2.1 Installation

To compile

```
mkdir build
cd build
cmake --toolchain ../toolchain.cmake ..
make
cd ..
```

Currently the toolchain file requires `g++-13` and `gcc-13` installed, but any c++/c compiler that supports openmp will work. Replace `g++-13` and `gcc-13` to the path to the respective supporting compiler in `toolchain.cmake`. To remake, clean up the build directory and rerun the compile commands.

To run, from the base directory run

```
./build/bin/Circuit_Optimizer
```

### 2.2 File generation

To write the result to file, from the base directory run

```
mkdir out
./build/bin/Circuit_Optimizer
```

### 2.3 Macros

The package defines several macros for user to use.

#### 2.3.1 PARALLEL\_ENABLE

Macro for MPI functionalities. If defined, the program will utilize MPI if `parallel_mpi` is set to 1. `MPI_Init()` should be called inside `main` before any GA object is created. `MPI_Finalize()` is handled in GA's destructor, meaning that only one GA object should be created and destroyed.

#### 2.3.2 OUT\_RESULT\_TO\_FILE

Macro for write result to file.

### 2.3.3 PERFORMANCE\_ANALYSIS\_MACRO

Macro for doing extra work to perform some analysis.

## 2.4 Test

To run the test

```
./build/tests/bin/test_validity  
./build/tests/bin/simulator_tests
```

## 2.5 Post-processing

The python script contained in [src/graph.py](#) can be used to generate a graph of the results. The script takes in a dat file containing the results of the simulation and generates a graph of the results.

## 2.6 Documentation

To generate configuration file

```
doxygen -g
```

To generate documentation

```
doxygen Doxyfile
```

## 2.7 License

Distributed under the MIT License. See LICENSE for more information.

# 3 Namespace Index

## 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">graph</a>	<a href="#">5</a>
<a href="#">graph_online</a>	<a href="#">7</a>

# 4 Class Index

## 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Algorithm_Parameters</a>	<a href="#">10</a>
--------------------------------------	--------------------

<a href="#">CCircuit</a>	12
<a href="#">CSimulator</a>	14
<a href="#">CUnit</a>	17
<a href="#">GA</a>	20
<a href="#">SimulationParameters</a>	31

## 5 File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">build/DartConfiguration.tcl</a>	50
<a href="#">build/CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c</a>	33
<a href="#">build/CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp</a>	38
<a href="#">build/CMakeFiles/FindMPI/test_mpi.cpp</a>	42
<a href="#">build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c</a>	44
<a href="#">build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp</a>	46
<a href="#">build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c</a>	48
<a href="#">build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp</a>	49
<a href="#">CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c</a>	35
<a href="#">CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp</a>	40
<a href="#">CMakeFiles/FindMPI/test_mpi.cpp</a>	43
<a href="#">CMakeFiles/FindOpenMP/OpenMPCheckVersion.c</a>	45
<a href="#">CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp</a>	47
<a href="#">CMakeFiles/FindOpenMP/OpenMPTryFlag.c</a>	48
<a href="#">CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp</a>	50
<a href="#">include/CCircuit.h</a>	54
<a href="#">include/CSimulator.h</a>	54
<a href="#">include/CUnit.h</a>	56
<a href="#">include/Genetic_Algorithm.h</a>	57
<a href="#">src/CCircuit.cpp</a>	58
<a href="#">src/CSimulator.cpp</a>	59
Main code file for the <a href="#">CSimulator</a> class	



<a href="#">src/Genetic_Algorithm.cpp</a>	61
<a href="#">src/graph.py</a>	63
<a href="#">src/graph_online.py</a>	64
<a href="#">src/main.cpp</a>	64
<a href="#">tests/test_GA.cpp</a>	65
<a href="#">tests/test_simulator.cpp</a>	67
<a href="#">tests/test_validity.cpp</a>	69

## 6 Namespace Documentation

### 6.1 graph Namespace Reference

#### Functions

- [def read\\_inputs](#) ([file\\_path](#))

#### Variables

- [graph](#) = [graphviz.Digraph\(\)](#)
- [rankdir](#)
- [shape](#)
- [int index](#) = 1
- [color](#)
- [headport](#)
- [tailport](#)
- [arrowhead](#)
- [width](#)
- [height](#)
- [copy\\_list](#) = [int\\_list.copy\(\)](#)
- [largest](#) = [max\(copy\\_list\)](#)
- [second\\_largest](#) = [max\(copy\\_list\)](#)
- [string node\\_name](#) = 'Unit ' + [str\(i\)](#)
- [string end\\_node](#) = 'Tailing'
- [label](#)
- [cleanup](#)
- [True](#)
- [format](#)

#### 6.1.1 Function Documentation

##### 6.1.1.1 [read\\_inputs\(\)](#)

```
def graph.read_inputs (
    file_path )
```

## 6.1.2 Variable Documentation

**6.1.2.1 arrowhead** `graph.arrowhead`

**6.1.2.2 cleanup** `graph.cleanup`

**6.1.2.3 color** `graph.color`

**6.1.2.4 copy\_list** `graph.copy_list = int_list.copy()`

**6.1.2.5 end\_node** `string graph.end_node = 'Tailing'`

**6.1.2.6 format** `graph.format`

**6.1.2.7 graph** `graph.graph = graphviz.Digraph()`

**6.1.2.8 headport** `graph.headport`

**6.1.2.9 height** `graph.height`

**6.1.2.10 index** `int graph.index = 1`

**6.1.2.11 label** `graph.label`

**6.1.2.12 largest** `graph.largest = max(copy_list)`

**6.1.2.13 node\_name** `graph.node_name = 'Unit ' + str(i)`

**6.1.2.14 rankdir** `graph.rankdir`

**6.1.2.15 second\_largest** `graph.second_largest = max(copy_list)`

**6.1.2.16 shape** `graph.shape`

**6.1.2.17 tailport** `graph.tailport`

**6.1.2.18 True** `graph.True`

**6.1.2.19 width** `graph.width`

## 6.2 graph\_online Namespace Reference

### Functions

- def `read_inputs` (file\_path)

## Variables

- `def int_list = read_inputs('./inputTest.dat')`
- `graph = graphviz.Digraph()`
- `rankdir`
- `shape`
- `int index = 1`
- `color`
- `headport`
- `tailport`
- `arrowhead`
- `width`
- `height`
- `def copy_list = int_list.copy()`
- `largest = max(copy_list)`
- `second_largest = max(copy_list)`
- `string node_name = 'Unit ' + str(i)`
- `string end_node = 'Tailing'`
- `cleanup`
- `True`
- `format`

### 6.2.1 Function Documentation

**6.2.1.1 read\_inputs()** `def graph_online.read_inputs (`  
`file_path )`

### 6.2.2 Variable Documentation

**6.2.2.1 arrowhead** `graph_online.arrowhead`

**6.2.2.2 cleanup** `graph_online.cleanup`

**6.2.2.3 color** `graph_online.color`

**6.2.2.4 copy\_list** `def graph_online.copy_list = int_list.copy()`

**6.2.2.5 end\_node** `string graph_online.end_node = 'Tailing'`

**6.2.2.6 format** `graph_online.format`

**6.2.2.7 graph** `graph_online.graph = graphviz.Digraph()`

**6.2.2.8 headport** `graph_online.headport`

**6.2.2.9 height** `graph_online.height`

**6.2.2.10 index** `int graph_online.index = 1`

**6.2.2.11 int\_list** `def graph_online.int_list = read\_inputs('./inputTest.dat')`

**6.2.2.12 largest** `graph_online.largest = max(copy\_list)`

**6.2.2.13 node\_name** `graph_online.node_name = 'Unit ' + str(i)`

**6.2.2.14 rankdir** `graph_online.rankdir`

**6.2.2.15 second\_largest** `graph_online.second_largest = max(copy\_list)`

**6.2.2.16 shape** `graph_online.shape`

**6.2.2.17 tailport** `graph_online.tailport`

**6.2.2.18 True** `graph_online.True`

**6.2.2.19 width** `graph_online.width`

## 7 Class Documentation

### 7.1 Algorithm\_Parameters Struct Reference

```
#include <Genetic_Algorithm.h>
```

#### Public Attributes

- int [max\\_iterations](#)
- double [tol](#)
- double [mutation\\_rate](#)
- int [mutation\\_size](#)
- int [parent\\_pool\\_size](#)
- int [population\\_size](#)
- int [circuit\\_size](#)
- int [selection\\_scheme](#)
- int [tournament\\_size](#)
- double [crossover\\_rate](#)
- int [parallel\\_mpi](#)
- int [communicate\\_interval](#)
- int [tournament\\_size\\_parallel](#)
- int [parent\\_comm\\_size\\_parallel](#)
- int [max\\_iter\\_without\\_progress](#)
- double [mutation\\_rate\\_increase\\_factor](#)
- int [max\\_iter\\_before\\_fail](#)
- int [write\\_interval](#)
- int [mutation\\_scheme](#)

#### 7.1.1 Detailed Description

Header for the Genetic Algorithm library

## 7.1.2 Member Data Documentation

**7.1.2.1 circuit\_size** `int Algorithm_Parameters::circuit_size`

**7.1.2.2 communicate\_interval** `int Algorithm_Parameters::communicate_interval`

**7.1.2.3 crossover\_rate** `double Algorithm_Parameters::crossover_rate`

**7.1.2.4 max\_iter\_before\_fail** `int Algorithm_Parameters::max_iter_before_fail`

**7.1.2.5 max\_iter\_without\_progress** `int Algorithm_Parameters::max_iter_without_progress`

**7.1.2.6 max\_iterations** `int Algorithm_Parameters::max_iterations`

**7.1.2.7 mutation\_rate** `double Algorithm_Parameters::mutation_rate`

**7.1.2.8 mutation\_rate\_increase\_factor** `double Algorithm_Parameters::mutation_rate_increase_factor`

**7.1.2.9 mutation\_scheme** `int Algorithm_Parameters::mutation_scheme`

**7.1.2.10 mutation\_size** `int Algorithm_Parameters::mutation_size`

**7.1.2.11 parallel\_mpi** int Algorithm\_Parameters::parallel\_mpi

**7.1.2.12 parent\_comm\_size\_parallel** int Algorithm\_Parameters::parent\_comm\_size\_parallel

**7.1.2.13 parent\_pool\_size** int Algorithm\_Parameters::parent\_pool\_size

**7.1.2.14 population\_size** int Algorithm\_Parameters::population\_size

**7.1.2.15 selection\_scheme** int Algorithm\_Parameters::selection\_scheme

**7.1.2.16 tol** double Algorithm\_Parameters::tol

**7.1.2.17 tournament\_size** int Algorithm\_Parameters::tournament\_size

**7.1.2.18 tournament\_size\_parallel** int Algorithm\_Parameters::tournament\_size\_parallel

**7.1.2.19 write\_interval** int Algorithm\_Parameters::write\_interval

The documentation for this struct was generated from the following file:

- include/[Genetic\\_Algorithm.h](#)

## 7.2 CCircuit Class Reference

```
#include <CCircuit.h>
```



## Public Member Functions

- [CCircuit](#) (int num\_units)  
*CCircuit Constructor.*
- [~CCircuit](#) ()  
*CCircuit Destructor.*
- bool [check\\_validity](#) (int \*circuit\_vector)  
*Checks the validity of the circuit.*

### 7.2.1 Detailed Description

Header for the circuit class

This header defines the circuit class and its associated functions

### 7.2.2 Constructor & Destructor Documentation

**7.2.2.1 CCircuit()** `CCircuit::CCircuit (int vector_size )`

[CCircuit](#) Constructor.

This function initializes the number of units in the circuit and dynamically allocates memory for marks array.

Parameters

<code>vector_size</code>	The size of the circuit vector.
--------------------------	---------------------------------

**7.2.2.2 ~CCircuit()** `CCircuit::~~CCircuit ( )`

[CCircuit](#) Destructor.

This function deletes the dynamically allocated memory for marks array.

### 7.2.3 Member Function Documentation

**7.2.3.1 check\_validity()** `bool CCircuit::check_validity (`  
`int * circuit_vector )`

Checks the validity of the circuit.

Check if the circuit is valid.

This function checks the validity of the circuit based on the following conditions:

1. Every unit must be accessible from the feed.
2. Every unit must have a route forward to both of the outlet streams.
3. There should be no self-recycle.
4. The destination for both products from a unit should not be the same unit.
5. The destination id of each unit must be between 0 to num\_units + 1. The first destination must not be tailings, the second destination must not be concentrate.

#### Returns

True if the circuit is valid, false otherwise.

This function checks for various conditions for circuit validity, including unit accessibility from the feed, route forward to both of the outlet streams, no self-recycle, different destinations, and appropriate destination ID.

#### Parameters

<code>circuit_vector</code>	A pointer to an array representing the circuit.
-----------------------------	---

#### Returns

true if the circuit is valid, false otherwise.

The documentation for this class was generated from the following files:

- [include/CCircuit.h](#)
- [src/CCircuit.cpp](#)

## 7.3 CSimulator Class Reference

```
#include <CSimulator.h>
```

#### Public Member Functions

- [CSimulator](#) ()
- [CSimulator](#) (int vector\_size, int \*circuit\_vector, const [SimulationParameters](#) &params)  
*CSimulator constructor.*
- bool [solve](#) ()  
*Solve the simulation.*

- double [calculate\\_reward](#) (bool is\_converge)  
*Calculate the reward.*
- double [get\\_total\\_concentrate\\_ger](#) ()
- double [get\\_total\\_concentrate\\_waste](#) ()
- double [get\\_total\\_tail\\_ger](#) ()
- double [get\\_total\\_tail\\_waste](#) ()
- std::vector< int > [get\\_final\\_concentrate\\_list](#) () const
- std::vector< int > [get\\_final\\_tail\\_list](#) () const
- std::vector< CUnit > [get\\_units\\_vector](#) ()
- void [write\\_vector\\_values\\_to\\_array](#) (double \*dest)  
*writing the values in CUnits to an array*

### 7.3.1 Constructor & Destructor Documentation

**7.3.1.1 CSimulator()** [1/2] `CSimulator::CSimulator ( ) [inline]`

**7.3.1.2 CSimulator()** [2/2] `CSimulator::CSimulator (`  
`int vector_size,`  
`int * circuit_vector,`  
`const SimulationParameters & params )`

[CSimulator](#) constructor.

#### Parameters

<i>vector_size</i>	Size of the circuit_vector.
<i>circuit_vector</i>	Vector representing the circuit.
<i>params</i>	Parameters for the simulation.

### 7.3.2 Member Function Documentation

**7.3.2.1 calculate\_reward()** `double CSimulator::calculate_reward (`  
`bool is_converge )`

Calculate the reward.

#### Parameters

<i>is_converge</i>	Boolean parameter determining if the simulator has converged.
--------------------	---

**Returns**

Reward for the simulation.

**7.3.2.2 get\_final\_concentrate\_list()** `std::vector<int> CSimulator::get_final_concentrate_list ( )`  
`const`

**7.3.2.3 get\_final\_tail\_list()** `std::vector<int> CSimulator::get_final_tail_list ( ) const`

**7.3.2.4 get\_total\_concentrate\_ges()** `double CSimulator::get_total_concentrate_ges ( ) [inline]`

**7.3.2.5 get\_total\_concentrate\_waste()** `double CSimulator::get_total_concentrate_waste ( ) [inline]`

**7.3.2.6 get\_total\_tail\_ges()** `double CSimulator::get_total_tail_ges ( ) [inline]`

**7.3.2.7 get\_total\_tail\_waste()** `double CSimulator::get_total_tail_waste ( ) [inline]`

**7.3.2.8 get\_units\_vector()** `std::vector<CUnit> CSimulator::get_units_vector ( ) [inline]`

**7.3.2.9 solve()** `bool CSimulator::solve ( )`

Solve the simulation.

**Returns**

True if the simulation was solved successfully, false otherwise.

**7.3.2.10 write\_vector\_values\_to\_array()** `void CSimulator::write_vector_values_to_array ( double * dest )`

writing the values in CUnits to an array

#### Returns

destination write buffer

The documentation for this class was generated from the following files:

- [include/CSimulator.h](#)
- [src/CSimulator.cpp](#)

## 7.4 CUnit Class Reference

```
#include <CUnit.h>
```

### Public Member Functions

- [CUnit \(\)](#)
- [CUnit \(int unit\\_id, bool is\\_feed=false\)](#)
- [~CUnit \(\)](#)
- [bool get\\_is\\_feed \(\) const](#)
- [int get\\_unit\\_id \(\) const](#)

### Public Attributes

- `std::vector< int >` [concentrate\\_list](#)
- `std::vector< int >` [tails\\_list](#)
- `std::vector< int >` [concentrate\\_feed\\_list](#)
- `std::vector< int >` [tails\\_feed\\_list](#)
- `double` [ger\\_recovery](#)
- `double` [waste\\_recovery](#)
- `double` [ger\\_flow\\_in](#)
- `double` [waste\\_flow\\_in](#)
- `double` [ger\\_flow\\_in\\_old](#)
- `double` [waste\\_flow\\_in\\_old](#)
- `double` [c\\_ger\\_flow\\_out](#)
- `double` [c\\_waste\\_flow\\_out](#)
- `double` [t\\_ger\\_flow\\_out](#)
- `double` [t\\_waste\\_flow\\_out](#)

### 7.4.1 Detailed Description

Header for the unit class

## 7.4.2 Constructor & Destructor Documentation

**7.4.2.1 CUnit()** [1/2] CUnit::CUnit ( ) [inline]

**7.4.2.2 CUnit()** [2/2] CUnit::CUnit (  
    int *unit\_id*,  
    bool *is\_feed* = *false* ) [inline]

**7.4.2.3 ~CUnit()** CUnit::~~CUnit ( ) [inline]

## 7.4.3 Member Function Documentation

**7.4.3.1 get\_is\_feed()** bool CUnit::get\_is\_feed ( ) const [inline]

**7.4.3.2 get\_unit\_id()** int CUnit::get\_unit\_id ( ) const [inline]

## 7.4.4 Member Data Documentation

**7.4.4.1 c\_ger\_flow\_out** double CUnit::c\_ger\_flow\_out

**7.4.4.2 c\_waste\_flow\_out** double CUnit::c\_waste\_flow\_out

**7.4.4.3 concentrate\_feed\_list** std::vector<int> CUnit::concentrate\_feed\_list

**7.4.4.4 concentrate\_list** `std::vector<int> CUnit::concentrate_list`

**7.4.4.5 ger\_flow\_in** `double CUnit::ger_flow_in`

**7.4.4.6 ger\_flow\_in\_old** `double CUnit::ger_flow_in_old`

**7.4.4.7 ger\_recovery** `double CUnit::ger_recovery`

**7.4.4.8 t\_ger\_flow\_out** `double CUnit::t_ger_flow_out`

**7.4.4.9 t\_waste\_flow\_out** `double CUnit::t_waste_flow_out`

**7.4.4.10 tails\_feed\_list** `std::vector<int> CUnit::tails_feed_list`

**7.4.4.11 tails\_list** `std::vector<int> CUnit::tails_list`

**7.4.4.12 waste\_flow\_in** `double CUnit::waste_flow_in`

**7.4.4.13 waste\_flow\_in\_old** `double CUnit::waste_flow_in_old`

**7.4.4.14 waste\_recovery** `double CUnit::waste_recovery`

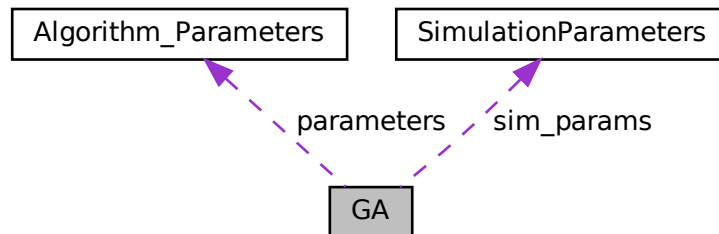
The documentation for this class was generated from the following file:

- [include/CUnit.h](#)

## 7.5 GA Class Reference

```
#include <Genetic_Algorithm.h>
```

Collaboration diagram for GA:



### Public Member Functions

- [GA \(\)](#)  
*Default constructor of the [GA](#) class. This constructor initializes the class member variables to default values.*
- [GA \(Algorithm\\_Parameters AP, SimulationParameters SP\)](#)  
*Constructor of the [GA](#) class. This constructor initializes the class member variables.*
- [~GA \(\)](#)  
*Destructor of the [GA](#) class. This destructor releases dynamically allocated memory.*
- void [setup \(\)](#)  
*This function sets up the genetic algorithm, initializing various parameters and data structures. The setup is based on the user passed parameters.*
- void [setup\\_parallel \(\)](#)
- void [optimize \(\)](#)  
*Main function to optimize the genetic algorithm.*
- void [optimize\\_without\\_mpi \(\)](#)  
*Executes the genetic algorithm optimization without MPI parallelism.*
- void [two\\_point\\_crossover](#) (int idx1, int idx2, int dest)  
*Performs the two point crossover.*
- void [generate\\_child \(\)](#)  
*This function generates all the children for next iteration in the genetic algorithm.*
- void [mutation](#) (int population\_idx)  
*Performs mutation operation on the population at a given index.*
- void [selection \(\)](#)  
*Performs the selection process on the current population.*
- void [rank\\_selection \(\)](#)  
*Performs the rank selection process on the current population.*
- void [roulette\\_selection \(\)](#)  
*Performs the roulette selection process on the current population.*
- void [tournament\\_selection \(\)](#)  
*Executes tournament selection.*
- double [rank\\_func](#) (int ranking)



- calculating a value based on ranking. in general, higher ranking will result in higher value
- void [set\\_best\\_to\\_parent](#) ()
  - Sets the best solution to the parent pool.
- void [in\\_place\\_sort](#) (int \*tailing\_val, int size, double \*tosort)
  - in-place sorting wrapper
- void [optimize\\_parallel](#) ()
- void [mpi\\_reduction\\_wrapper](#) ()
- void [tournament\\_selection\\_parallel](#) ()
- void [roulette\\_selection\\_parallel](#) ()
- int [get\\_idx](#) (int pos\_in\_array, int pos\_in\_vec)
  - This function returns an index into a one-dimensional array given two indices. The 1d array contains n elements, each one has size of size\_per\_cir.
- int [get\\_idx\\_node](#) (int pos, int node, int val)
  - This function returns an index into a one-dimensional array given three indices.
- int [get\\_node\\_num](#) (int pos)
  - get the node number
- void [copy\\_parent\\_to\\_child](#) (int parentid, int childid)
  - copy from parent to population
- void [generalized\\_copy](#) (int id1, int id2, int \*arr1, int \*arr2, int count)
  - generalized copy of two array of same structure
- void [reinitialize\\_pil](#) ()
  - reset the population\_idx\_list
- void [update\\_best](#) ()
  - update best result
- void [select\\_mate](#) (int &parent1, int &parent2)
- bool [check\\_same](#) (int \*p1, int \*p2)
  - This function checks if two arrays are the same.
- void [write\\_to\\_file](#) ()
  - This function writes the current best solution of the genetic algorithm to a file.

## Public Attributes

- [Algorithm\\_Parameters](#) parameters
- [SimulationParameters](#) sim\_params
- int [vec\\_size\\_parent](#)
  - Total size of parent-pool array.
- int [vec\\_size\\_population](#)
  - Total size of population-pool array.
- int [size\\_per\\_cir](#)
  - Size per circuit array.
- int [mate\\_count\\_all](#)
  - Parameters for performance analysis.
- int [same\\_mate](#)
  - Parameters for performance analysis.
- bool [nottouched](#)
  - Flag for updating the best fitness value.
- int \* [parents](#)
  - Parent pool.
- int \* [population](#)
  - Population pool.
- double [best\\_fitness](#)

- *Current best fitness.*
- int \* [best\\_circuit](#)
- *Current best circuit.*
- double \* [fitness\\_population](#)
- *Fitness of corresponding population.*
- double \* [fitness\\_parents](#)
- *Fitness of corresponding parent.*
- double \* [roulette\\_sum](#)
- *Rolling sum for roulette selection.*
- double \* [rank\\_sum](#)
- *Rolling sum for rank selection.*
- double \* [rank\\_val](#)
- *Ranking function values for each individual ranking.*
- int \* [population\\_idx\\_list](#)
- *Ranking of corresponding population.*
- double [old\\_best\\_fitness](#)
- *Last best fitness.*
- int [not\\_improving\\_count](#)
- *Iteration count during which the best fitness is not improving.*
- int [cur\\_iter](#)
- *Current iteration number of Genetic algorithm.*
- double \* [file\\_write\\_buffer](#)
- *Buffer to store the unit in/out flow information.*

### 7.5.1 Constructor & Destructor Documentation

#### 7.5.1.1 [GA\(\)](#) [1/2] `GA::GA ( )`

Default constructor of the [GA](#) class. This constructor initializes the class member variables to default values.

#### 7.5.1.2 [GA\(\)](#) [2/2] `GA::GA ( AlgorithmParameters AP, SimulationParameters SP )`

Constructor of the [GA](#) class. This constructor initializes the class member variables.

##### Parameters

<i>AP</i>	A structure containing the algorithm parameters.
<i>SP</i>	A structure containing the simulation parameters.

#### 7.5.1.3 [~GA\(\)](#) `GA::~~GA ( )`

Destructor of the [GA](#) class. This destructor releases dynamically allocated memory.

## 7.5.2 Member Function Documentation

**7.5.2.1 check\_same()** `bool GA::check_same (`  
`int * p1,`  
`int * p2 )`

This function checks if two arrays are the same.

### Parameters

<i>p1</i>	Pointer to the first array.
<i>p2</i>	Pointer to the second array.

### Returns

true if the two arrays are the same, false otherwise.

**7.5.2.2 copy\_parent\_to\_child()** `void GA::copy_parent_to_child (`  
`int parentid,`  
`int childid )`

copy from parent to population

### Parameters

<i>parentid</i>	index of parent to be copied
<i>childid</i>	index of population to be copied to

**7.5.2.3 generalized\_copy()** `void GA::generalized_copy (`  
`int id1,`  
`int id2,`  
`int * arr1,`  
`int * arr2,`  
`int count )`

generalized copy of two array of same structure

### Parameters

<i>id1</i>	idex in arr1 to copy to
------------	-------------------------

**Parameters**

<i>id2</i>	idex in arr2 to copy
<i>arr1</i>	destination array
<i>arr2</i>	source array
<i>count</i>	number of items to copy

**7.5.2.4 generate\_child()** `void GA::generate_child ( )`

This function generates all the children for next iteration in the genetic algorithm.

**7.5.2.5 get\_idx()** `int GA::get_idx (`  
`int pos_in_array,`  
`int pos_in_vec )`

This function returns an index into a one-dimensional array given two indices. The 1d array contains n elements, each one has size of size\_per\_cir.

**Parameters**

<i>pos_in_array</i>	The position in the array.
<i>pos_in_vec</i>	The position in the vector.

**Returns**

An index in a one-dimensional array.

**7.5.2.6 get\_idx\_node()** `int GA::get_idx_node (`  
`int pos,`  
`int node,`  
`int val )`

This function returns an index into a one-dimensional array given three indices.

**Parameters**

<i>pos</i>	The position in the array.
<i>node</i>	The node index.
<i>val</i>	The value index.

**Returns**

An index in a one-dimensional array.

**7.5.2.7 get\_node\_num()** `int GA::get_node_num (`  
    `int pos )`

get the node number

**Parameters**

<i>pos</i>	index in 1d array
------------	-------------------

**Returns**

the node number

**7.5.2.8 in\_place\_sort()** `void GA::in_place_sort (`  
    `int * tailing_val,`  
    `int size,`  
    `double * tosort )`

in-place sorting wrapper

**Parameters**

<i>tailing_val</i>	pointer to the array that should sort according to arr
<i>size</i>	size of array
<i>tosort</i>	pointer to the array to sort

**7.5.2.9 mpi\_reduction\_wrapper()** `void GA::mpi_reduction_wrapper ( )`

**7.5.2.10 mutation()** `void GA::mutation (`  
    `int population_idx )`

Performs mutation operation on the population at a given index.

**Parameters**

<i>population_idx</i>	Index of the population member to be mutated
-----------------------	--

The function randomly selects an index within the member of the population, then replaces the value at that index with a new random value.

#### 7.5.2.11 **optimize()** `void GA::optimize ( )`

Main function to optimize the genetic algorithm.

Depending on the defined macros, the function decides to execute parallel optimization or non-parallel optimization.

#### 7.5.2.12 **optimize\_parallel()** `void GA::optimize_parallel ( )`

#### 7.5.2.13 **optimize\_without\_mpi()** `void GA::optimize_without_mpi ( )`

Executes the genetic algorithm optimization without MPI parallelism.

The function iterates through a series of generations, each time generating a new population of possible solutions. The solutions are selected and modified to form the next generation.

#### 7.5.2.14 **rank\_func()** `double GA::rank_func (` `int ranking )`

calculating a value based on ranking. in general, higher ranking will result in higher value

##### Parameters

<i>ranking</i>	ranking
----------------	---------

##### Returns

the corresponding value of this ranking

#### 7.5.2.15 **rank\_selection()** `void GA::rank_selection ( )`

Performs the rank selection process on the current population.

#### 7.5.2.16 **reinitialize\_pil()** `void GA::reinitialize_pil ( )`

reset the `population_idx_list`

**7.5.2.17 roulette\_selection()** `void GA::roulette_selection ( )`

Performs the roulette selection process on the current population.

**7.5.2.18 roulette\_selection\_parallel()** `void GA::roulette_selection_parallel ( )`**7.5.2.19 select\_mate()** `void GA::select_mate (`  
`int & parent1,`  
`int & parent2 )`**7.5.2.20 selection()** `void GA::selection ( )`

Performs the selection process on the current population.

Depending on the specified selection scheme (rank, roulette or tournament), it selects the most suitable individuals from the population to be parents for the next generation.

**7.5.2.21 set\_best\_to\_parent()** `void GA::set_best_to_parent ( )`

Sets the best solution to the parent pool.

This is typically done after a new generation is created, and the best solution from the previous generation is kept for comparison.

**7.5.2.22 setup()** `void GA::setup ( )`

This function sets up the genetic algorithm, initializing various parameters and data structures. The setup is based on the user passed parameters.

**7.5.2.23 setup\_parallel()** `void GA::setup_parallel ( )`**7.5.2.24 tournament\_selection()** `void GA::tournament_selection ( )`

Executes tournament selection.

Selects the best solutions from a randomly chosen subset of the population. The chosen solutions are then used to form the parent pool for the next generation.

**7.5.2.25 tournament\_selection\_parallel()** `void GA::tournament_selection_parallel ( )`**7.5.2.26 two\_point\_crossover()** `void GA::two_point_crossover (`  
`int idx1,`  
`int idx2,`  
`int dest )`

Performs the two point crossover.

**Parameters**

<i>idx1</i>	index of the first parent
<i>idx2</i>	index of the second parent
<i>dest</i>	index of the population which will be overwritten

**7.5.2.27 update\_best()** `void GA::update_best ( )`

update best result

**7.5.2.28 write\_to\_file()** `void GA::write_to_file ( )`

This function writes the current best solution of the genetic algorithm to a file.

### 7.5.3 Member Data Documentation

**7.5.3.1 best\_circuit** `int* GA::best_circuit`

Current best circuit.

**7.5.3.2 best\_fitness** `double GA::best_fitness`

Current best fitness.

**7.5.3.3 cur\_iter** `int GA::cur_iter`

Current iteration number of Genetic algorithm.

**7.5.3.4 file\_write\_buffer** `double* GA::file_write_buffer`

Buffer to store the unit in/out flow information.



**7.5.3.5 fitness\_parents** `double* GA::fitness_parents`

Fitness of corresponding parent.

**7.5.3.6 fitness\_population** `double* GA::fitness_population`

Fitness of corresponding population.

**7.5.3.7 mate\_count\_all** `int GA::mate_count_all`

Parameters for performance analysis.

**7.5.3.8 not\_improving\_count** `int GA::not_improving_count`

Iteration count during which the best fitness is not improving.

**7.5.3.9 nottouched** `bool GA::nottouched`

Flag for updating the best fitness value.

**7.5.3.10 old\_best\_fitness** `double GA::old_best_fitness`

Last best fitness.

**7.5.3.11 parameters** `Algorithm_Parameters GA::parameters`

**7.5.3.12 parents** `int* GA::parents`

Parent pool.

**7.5.3.13 population** `int* GA::population`

Population pool.

**7.5.3.14 population\_idex\_list** `int* GA::population_idex_list`

Ranking of corresponding population.

**7.5.3.15 rank\_sum** `double* GA::rank_sum`

Rolling sum for rank selection.

**7.5.3.16 rank\_val** `double* GA::rank_val`

Ranking function values for each individual ranking.

**7.5.3.17 roulette\_sum** `double* GA::roulette_sum`

Rolling sum for roulette selection.

**7.5.3.18 same\_mate** `int GA::same_mate`

Parameters for performance analysis.

**7.5.3.19 sim\_params** `SimulationParameters GA::sim_params`

**7.5.3.20 size\_per\_cir** `int GA::size_per_cir`

Size per circuit array.

**7.5.3.21 vec\_size\_parent** `int GA::vec_size_parent`

Total size of parent-pool array.

**7.5.3.22 vec\_size\_population** `int GA::vec_size_population`

Total size of population-pool array.

The documentation for this class was generated from the following files:

- [include/Genetic\\_Algorithm.h](#)
- [src/Genetic\\_Algorithm.cpp](#)

## 7.6 SimulationParameters Struct Reference

```
#include <CSimulator.h>
```

### Public Attributes

- double [F0\\_ger](#)
- double [F0\\_waste](#)
- double [k\\_ger](#)
- double [k\\_waste](#)
- double [volume](#)
- double [phi](#)
- double [rho](#)
- double [ger\\_reward](#)
- double [waste\\_reward](#)
- double [min\\_flowrate](#)
- int [max\\_iter](#)

### 7.6.1 Detailed Description

header file for the circuit simulator

This header file defines the function that will be used to evaluate the circuit

### 7.6.2 Member Data Documentation

**7.6.2.1 F0\_ger** `double SimulationParameters::F0_ger`

**7.6.2.2 F0\_waste** double SimulationParameters::F0\_waste

**7.6.2.3 ger\_reward** double SimulationParameters::ger\_reward

**7.6.2.4 k\_ger** double SimulationParameters::k\_ger

**7.6.2.5 k\_waste** double SimulationParameters::k\_waste

**7.6.2.6 max\_iter** int SimulationParameters::max\_iter

**7.6.2.7 min\_flowrate** double SimulationParameters::min\_flowrate

**7.6.2.8 phi** double SimulationParameters::phi

**7.6.2.9 rho** double SimulationParameters::rho

**7.6.2.10 volume** double SimulationParameters::volume

**7.6.2.11 waste\_reward** double SimulationParameters::waste\_reward

The documentation for this struct was generated from the following file:

- [include/CSimulator.h](#)

## 8 File Documentation

### 8.1 build/CMakeCache.txt File Reference

#### Variables

- BUILD\_TESTING `__pad0__`
- mpi `MPI_CXX_LINK_FLAGS`
- pthread `OpenMP_C_FLAGS`
- pthread `OpenMP_gomp_LIBRARY`

#### 8.1.1 Variable Documentation

**8.1.1.1 `__pad0__`** `BUILD_TESTING __pad0__`

**8.1.1.2 `MPI_CXX_LINK_FLAGS`** `mpi MPI_CXX_LINK_FLAGS`

**8.1.1.3 `OpenMP_C_FLAGS`** `pthread OpenMP_C_FLAGS`

**8.1.1.4 `OpenMP_gomp_LIBRARY`** `pthread OpenMP_gomp_LIBRARY`

### 8.2 build/CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c File Reference

#### Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define C_DIALECT`

#### Functions

- `int main (int argc, char *argv[ ])`

## Variables

- char const \* [info\\_compiler](#) = "INFO" ":" "compiler[" COMPILER\_ID "]"
- char const \* [info\\_platform](#) = "INFO" ":" "platform[" PLATFORM\_ID "]"
- char const \* [info\\_arch](#) = "INFO" ":" "arch[" ARCHITECTURE\_ID "]"
- const char \* [info\\_language\\_dialect\\_default](#)

## 8.2.1 Macro Definition Documentation

### 8.2.1.1 ARCHITECTURE\_ID `#define ARCHITECTURE_ID`

### 8.2.1.2 C\_DIALECT `#define C_DIALECT`

### 8.2.1.3 COMPILER\_ID `#define COMPILER_ID ""`

### 8.2.1.4 DEC `#define DEC( n )`

#### Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

### 8.2.1.5 HEX `#define HEX( n )`

#### Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

### 8.2.1.6 PLATFORM\_ID `#define PLATFORM_ID`

**8.2.1.7 STRINGIFY** `#define STRINGIFY(  
X ) STRINGIFY_HELPER(X)`

**8.2.1.8 STRINGIFY\_HELPER** `#define STRINGIFY_HELPER(  
X ) #X`

## 8.2.2 Function Documentation

**8.2.2.1 main()** `int main (  
int argc,  
char * argv[] )`

## 8.2.3 Variable Documentation

**8.2.3.1 info\_arch** `char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

**8.2.3.2 info\_compiler** `char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

**8.2.3.3 info\_language\_dialect\_default** `const char* info_language_dialect_default`

Initial value:

```
=  
"INFO" ":" "dialect_default[" C_DIALECT "]"
```

**8.2.3.4 info\_platform** `char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

## 8.3 CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c File Reference

### Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define C_DIALECT`

## Functions

- int `main` (int argc, char \*argv[])

## Variables

- char const \* `info_compiler` = "INFO" ":" "compiler[" COMPILER\_ID "]"
- char const \* `info_platform` = "INFO" ":" "platform[" PLATFORM\_ID "]"
- char const \* `info_arch` = "INFO" ":" "arch[" ARCHITECTURE\_ID "]"
- const char \* `info_language_dialect_default`

### 8.3.1 Macro Definition Documentation

#### 8.3.1.1 ARCHITECTURE\_ID `#define ARCHITECTURE_ID`

#### 8.3.1.2 C\_DIALECT `#define C_DIALECT`

#### 8.3.1.3 COMPILER\_ID `#define COMPILER_ID ""`

#### 8.3.1.4 DEC `#define DEC( n )`

##### Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

#### 8.3.1.5 HEX `#define HEX( n )`

##### Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```



**8.3.1.6 PLATFORM\_ID** `#define PLATFORM_ID`

**8.3.1.7 STRINGIFY** `#define STRINGIFY(  
X ) STRINGIFY_HELPER(X)`

**8.3.1.8 STRINGIFY\_HELPER** `#define STRINGIFY_HELPER(  
X ) #X`

## 8.3.2 Function Documentation

**8.3.2.1 main()** `int main (  
int argc,  
char * argv[] )`

## 8.3.3 Variable Documentation

**8.3.3.1 info\_arch** `char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

**8.3.3.2 info\_compiler** `char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

**8.3.3.3 info\_language\_dialect\_default** `const char* info_language_dialect_default`

**Initial value:**

`=  
"INFO" ":" "dialect_default[" C_DIALECT "]"`

**8.3.3.4 info\_platform** `char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

## 8.4 build/CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

### Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD __cplusplus`

### Functions

- `int main (int argc, char *argv[ ])`

### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_dialect_default`

### 8.4.1 Macro Definition Documentation

#### 8.4.1.1 ARCHITECTURE\_ID `#define ARCHITECTURE_ID`

#### 8.4.1.2 COMPILER\_ID `#define COMPILER_ID ""`

#### 8.4.1.3 CXX\_STD `#define CXX_STD __cplusplus`

#### 8.4.1.4 DEC `#define DEC( n )`

##### Value:

```
( '0' + ((n) / 10000000) % 10 ), \
( '0' + ((n) / 1000000) % 10 ), \
( '0' + ((n) / 100000) % 10 ), \
( '0' + ((n) / 10000) % 10 ), \
( '0' + ((n) / 1000) % 10 ), \
( '0' + ((n) / 100) % 10 ), \
( '0' + ((n) / 10) % 10 ), \
( '0' + ((n) % 10) )
```

**8.4.1.5 HEX** `#define HEX(  
    n )`

**Value:**

```
('0' + ((n)>>28 & 0xF)), \
('0' + ((n)>>24 & 0xF)), \
('0' + ((n)>>20 & 0xF)), \
('0' + ((n)>>16 & 0xF)), \
('0' + ((n)>>12 & 0xF)), \
('0' + ((n)>>8  & 0xF)), \
('0' + ((n)>>4  & 0xF)), \
('0' + ((n)    & 0xF))
```

**8.4.1.6 PLATFORM\_ID** `#define PLATFORM_ID`

**8.4.1.7 STRINGIFY** `#define STRINGIFY(  
    X ) STRINGIFY\_HELPER(X)`

**8.4.1.8 STRINGIFY\_HELPER** `#define STRINGIFY_HELPER(  
    X ) #X`

## 8.4.2 Function Documentation

**8.4.2.1 main()** `int main (  
    int argc,  
    char * argv[] )`

## 8.4.3 Variable Documentation

**8.4.3.1 info\_arch** `char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

**8.4.3.2 info\_compiler** `char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

**8.4.3.3 info\_language\_dialect\_default** `const char* info_language_dialect_default`

**Initial value:**

```
= "INFO" ":" "dialect_default["  
  "98"  
"]"
```

**8.4.3.4 info\_platform** `char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

## 8.5 CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

### Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD __cplusplus`

### Functions

- `int main (int argc, char *argv[])`

### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_dialect_default`

### 8.5.1 Macro Definition Documentation

**8.5.1.1 ARCHITECTURE\_ID** `#define ARCHITECTURE_ID`

**8.5.1.2 COMPILER\_ID** `#define COMPILER_ID ""`

**8.5.1.3 CXX\_STD** `#define CXX_STD __cplusplus`**8.5.1.4 DEC** `#define DEC(  
    n )`**Value:**

```

('0' + (((n) / 100000000)%10)), \
('0' + (((n) / 1000000)%10)), \
('0' + (((n) / 100000)%10)), \
('0' + (((n) / 10000)%10)), \
('0' + (((n) / 1000)%10)), \
('0' + (((n) / 100)%10)), \
('0' + (((n) / 10)%10)), \
('0' + ((n) % 10))

```

**8.5.1.5 HEX** `#define HEX(  
    n )`**Value:**

```

('0' + ((n)>>28 & 0xF)), \
('0' + ((n)>>24 & 0xF)), \
('0' + ((n)>>20 & 0xF)), \
('0' + ((n)>>16 & 0xF)), \
('0' + ((n)>>12 & 0xF)), \
('0' + ((n)>>8 & 0xF)), \
('0' + ((n)>>4 & 0xF)), \
('0' + ((n) & 0xF))

```

**8.5.1.6 PLATFORM\_ID** `#define PLATFORM_ID`**8.5.1.7 STRINGIFY** `#define STRINGIFY(  
    X ) STRINGIFY\_HELPER(X)`**8.5.1.8 STRINGIFY\_HELPER** `#define STRINGIFY_HELPER(  
    X ) #X`**8.5.2 Function Documentation****8.5.2.1 main()** `int main (  
    int argc,  
    char * argv[] )`

### 8.5.3 Variable Documentation

**8.5.3.1 info\_arch** `char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

**8.5.3.2 info\_compiler** `char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

**8.5.3.3 info\_language\_dialect\_default** `const char* info_language_dialect_default`

**Initial value:**

```
= "INFO" ":" "dialect_default["  
  "98"  
"]"
```

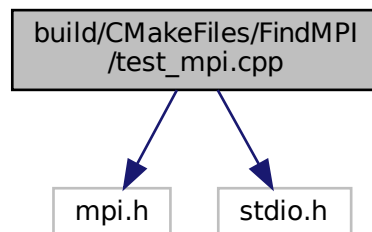
**8.5.3.4 info\_platform** `char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

## 8.6 build/CMakeFiles/CMakeRuleHashes.txt File Reference

## 8.7 CMakeFiles/CMakeRuleHashes.txt File Reference

## 8.8 build/CMakeFiles/FindMPI/test\_mpi.cpp File Reference

```
#include <mpi.h>  
#include <stdio.h>  
Include dependency graph for test_mpi.cpp:
```



## Functions

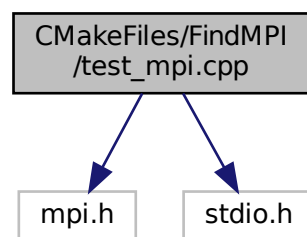
- int `main` (int argc, char \*argv[])

### 8.8.1 Function Documentation

**8.8.1.1 main()** `int main (`  
    `int argc,`  
    `char * argv[] )`

## 8.9 CMakeFiles/FindMPI/test\_mpi.cpp File Reference

```
#include <mpi.h>
#include <stdio.h>
Include dependency graph for test_mpi.cpp:
```



## Functions

- int `main` (int argc, char \*argv[])

### 8.9.1 Function Documentation

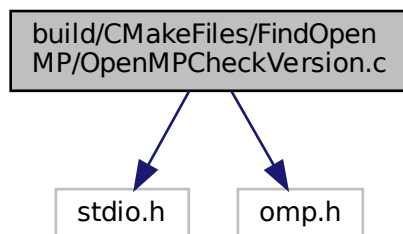
**8.9.1.1 main()** `int main (`  
    `int argc,`  
    `char * argv[] )`

## 8.10 build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference

```
#include <stdio.h>
```

```
#include <omp.h>
```

Include dependency graph for OpenMPCheckVersion.c:



### Functions

- int [main](#) (void)

### Variables

- const char [ompver\\_str](#) []

## 8.10.1 Function Documentation

**8.10.1.1 main()** int main (void )

## 8.10.2 Variable Documentation

**8.10.2.1 ompver\_str** const char ompver\_str[]

### Initial value:

```
= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'P', '-', 'd', 'a', 't', 'e', '[',
    ('0' + ((_OPENMP/100000)%10)),
    ('0' + ((_OPENMP/10000)%10)),
    ('0' + ((_OPENMP/1000)%10)),
    ('0' + ((_OPENMP/100)%10)),
    ('0' + ((_OPENMP/10)%10)),
    ('0' + ((_OPENMP/1)%10)),
    ']', '\0' }
```

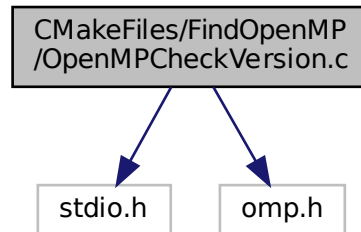


## 8.11 CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference

```
#include <stdio.h>
```

```
#include <omp.h>
```

Include dependency graph for OpenMPCheckVersion.c:



### Functions

- int [main](#) (void)

### Variables

- const char [ompver\\_str](#) []

#### 8.11.1 Function Documentation

**8.11.1.1 main()** int main (void )

#### 8.11.2 Variable Documentation

**8.11.2.1 ompver\_str** const char ompver\_str[]

##### Initial value:

```

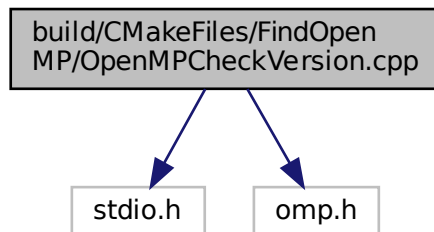
= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'P', '-', 'd', 'a', 't', 'e', '[',
    ('0' + ((_OPENMP/100000)%10)),
    ('0' + ((_OPENMP/10000)%10)),
    ('0' + ((_OPENMP/1000)%10)),
    ('0' + ((_OPENMP/100)%10)),
    ('0' + ((_OPENMP/10)%10)),
    ('0' + ((_OPENMP/1)%10)),
    ']', '\0' }
  
```

## 8.12 build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference

```
#include <stdio.h>
```

```
#include <omp.h>
```

Include dependency graph for OpenMPCheckVersion.cpp:



### Functions

- int [main](#) (void)

### Variables

- const char [ompver\\_str](#) []

## 8.12.1 Function Documentation

**8.12.1.1 main()** int main (void )

## 8.12.2 Variable Documentation

**8.12.2.1 ompver\_str** const char ompver\_str[]

### Initial value:

```

= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'P', '-', 'd', 'a', 't', 'e', '[',
    ('0' + ((_OPENMP/100000)%10)),
    ('0' + ((_OPENMP/10000)%10)),
    ('0' + ((_OPENMP/1000)%10)),
    ('0' + ((_OPENMP/100)%10)),
    ('0' + ((_OPENMP/10)%10)),
    ('0' + ((_OPENMP/1)%10)),
    ']', '\0' }

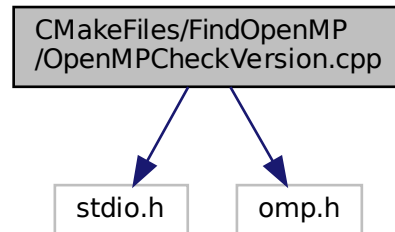
```

## 8.13 CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference

```
#include <stdio.h>
```

```
#include <omp.h>
```

Include dependency graph for OpenMPCheckVersion.cpp:



### Functions

- int `main` (void)

### Variables

- const char `ompver_str` []

#### 8.13.1 Function Documentation

**8.13.1.1 `main()`** int main (void )

#### 8.13.2 Variable Documentation

**8.13.2.1 `ompver_str`** const char ompver\_str[]

Initial value:

```

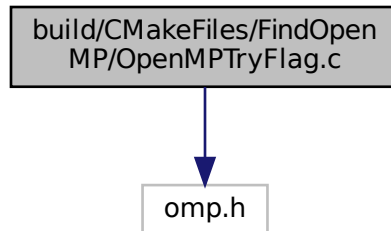
= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'P', '-', 'd', 'a', 't', 'e', '[',
    ('0' + ((__OPENMP/100000)%10)),
    ('0' + ((__OPENMP/10000)%10)),
    ('0' + ((__OPENMP/1000)%10)),
    ('0' + ((__OPENMP/100)%10)),
    ('0' + ((__OPENMP/10)%10)),
    ('0' + ((__OPENMP/1)%10)),
    ']', '\0' }

```

## 8.14 build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c File Reference

```
#include <omp.h>
```

Include dependency graph for OpenMPTryFlag.c:



### Functions

- int [main](#) (void)

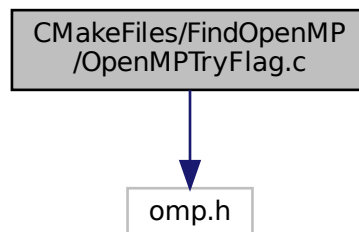
#### 8.14.1 Function Documentation

**8.14.1.1 main()** `int main (void )`

## 8.15 CMakeFiles/FindOpenMP/OpenMPTryFlag.c File Reference

```
#include <omp.h>
```

Include dependency graph for OpenMPTryFlag.c:



## Functions

- int [main](#) (void)

### 8.15.1 Function Documentation

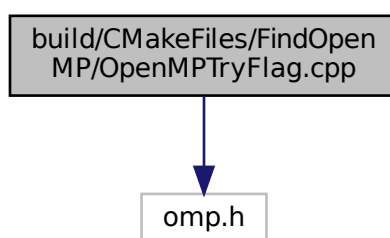
**8.15.1.1 main()**

```
int main (  
    void )
```

## 8.16 build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp File Reference

```
#include <omp.h>
```

Include dependency graph for OpenMPTryFlag.cpp:



## Functions

- int [main](#) (void)

### 8.16.1 Function Documentation

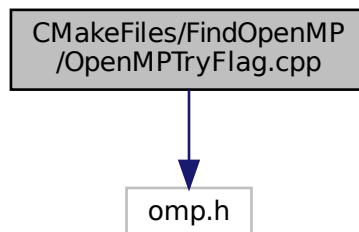
**8.16.1.1 main()**

```
int main (  
    void )
```

## 8.17 CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp File Reference

```
#include <omp.h>
```

Include dependency graph for OpenMPTryFlag.cpp:



### Functions

- int [main](#) (void)

#### 8.17.1 Function Documentation

**8.17.1.1 main()** `int main (`  
`void )`

## 8.18 build/CMakeFiles/TargetDirectories.txt File Reference

## 8.19 CMakeFiles/TargetDirectories.txt File Reference

## 8.20 build/DartConfiguration.tcl File Reference

## 8.21 build/src/CMakeFiles/Circuit\_Optimizer.dir/link.txt File Reference

### Variables

- `usr bin c fopenmp pthread CMakeFiles Circuit_Optimizer dir` [main](#) `cpp o o bin Circuit_Optimizer` [WI](#)
- `usr bin c fopenmp pthread CMakeFiles Circuit_Optimizer dir` [main](#) `cpp o o bin Circuit_Optimizer` [rpath](#)

#### 8.21.1 Variable Documentation

**8.21.1.1 rpath** usr bin c fopenmp pthread CMakeFiles Circuit\_Optimizer dir main cpp o o bin Circuit\_Optimizer rpath

**8.21.1.2 Wl** usr bin c fopenmp pthread CMakeFiles Circuit\_Optimizer dir main cpp o o bin Circuit\_Optimizer Wl

## 8.22 build/src/CMakeFiles/circuitSimulator.dir/link.txt File Reference

## 8.23 build/src/CMakeFiles/geneticAlgorithm.dir/link.txt File Reference

## 8.24 build/tests/CMakeFiles/simulator\_tests.dir/link.txt File Reference

## 8.25 build/tests/CMakeFiles/test\_GA.dir/link.txt File Reference

### Variables

- usr bin c fopenmp pthread CMakeFiles test\_GA dir test\_GA cpp o o bin test\_GA [Wl](#)
- usr bin c fopenmp pthread CMakeFiles test\_GA dir test\_GA cpp o o bin test\_GA [rpath](#)

### 8.25.1 Variable Documentation

**8.25.1.1 rpath** usr bin c fopenmp pthread CMakeFiles test\_GA dir test\_GA cpp o o bin test\_GA rpath

**8.25.1.2 Wl** usr bin c fopenmp pthread CMakeFiles test\_GA dir test\_GA cpp o o bin test\_GA Wl

## 8.26 build/tests/CMakeFiles/test\_simulator.dir/link.txt File Reference

### Variables

- usr bin c fopenmp pthread CMakeFiles test\_simulator dir test\_simulator cpp o o bin test\_simulator [Wl](#)
- usr bin c fopenmp pthread CMakeFiles test\_simulator dir test\_simulator cpp o o bin test\_simulator [rpath](#)

### 8.26.1 Variable Documentation

**8.26.1.1 rpath** `usr bin c fopenmp pthread CMakeFiles test_simulator dir test_simulator cpp o o bin test_simulator rpath`

**8.26.1.2 Wl** `usr bin c fopenmp pthread CMakeFiles test_simulator dir test_simulator cpp o o bin test_simulator Wl`

## 8.27 build/tests/CMakeFiles/test\_validity.dir/link.txt File Reference

### Variables

- `usr bin c fopenmp pthread CMakeFiles test_validity dir test_validity cpp o o bin test_validity` [Wl](#)
- `usr bin c fopenmp pthread CMakeFiles test_validity dir test_validity cpp o o bin test_validity` [rpath](#)

### 8.27.1 Variable Documentation

**8.27.1.1 rpath** `usr bin c fopenmp pthread CMakeFiles test_validity dir test_validity cpp o o bin test_validity rpath`

**8.27.1.2 Wl** `usr bin c fopenmp pthread CMakeFiles test_validity dir test_validity cpp o o bin test_validity Wl`

## 8.28 src/CMakeFiles/Circuit\_Optimizer.dir/link.txt File Reference

### Variables

- `usr bin c fopenmp pthread CMakeFiles Circuit_Optimizer dir` [main](#) `cpp o o bin Circuit_Optimizer` [Wl](#)
- `usr bin c fopenmp pthread CMakeFiles Circuit_Optimizer dir` [main](#) `cpp o o bin Circuit_Optimizer` [rpath](#)

### 8.28.1 Variable Documentation

**8.28.1.1 rpath** `usr bin c fopenmp pthread CMakeFiles Circuit_Optimizer dir` [main](#) `cpp o o bin Circuit_Optimizer rpath`

**8.28.1.2 Wl** `usr bin c fopenmp pthread CMakeFiles Circuit_Optimizer dir` [main](#) `cpp o o bin Circuit_Optimizer Wl`



**8.29 src/CMakeFiles/circuitSimulator.dir/link.txt File Reference****8.30 src/CMakeFiles/geneticAlgorithm.dir/link.txt File Reference****8.31 tests/CMakeFiles/simulator\_tests.dir/link.txt File Reference****8.32 tests/CMakeFiles/test\_simulator.dir/link.txt File Reference****8.33 tests/CMakeFiles/test\_validity.dir/link.txt File Reference****8.34 build/Testing/Temporary/CTestCostData.txt File Reference****8.35 Testing/Temporary/CTestCostData.txt File Reference****8.36 CMakeLists.txt File Reference****8.37 src/CMakeLists.txt File Reference****Functions**

- [add\\_library](#) (geneticAlgorithm Genetic\_Algorithm.cpp) set\_target\_properties(geneticAlgorithm PROPERTIES CXX\_STANDARD 17 ARCHIVE\_OUTPUT\_DIRECTORY "\$

**8.37.1 Function Documentation**

**8.37.1.1 add\_library()** `add_library (geneticAlgorithm Genetic_Algorithm. cpp )`

**8.38 tests/CMakeLists.txt File Reference****Functions**

- [project](#) (tests) list(APPEND Tests test\_validity test\_simulator test\_GA) foreach(TEST IN LISTS Tests) add\_executable(\$

**8.38.1 Function Documentation**

**8.38.1.1 project()** `project (tests )`

```
graph TD; include[include/CCircuit.h] --> CUnit[CUnit.h]; include --> vector; CUnit --> vector;
```

```

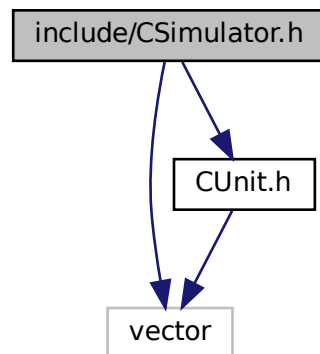
graph TD
    GA_H[include/Genetic_Algorithm.h] --> C_H[include/Circuit.h]
    C_CPP[src/Circuit.cpp] --> C_H
    S_CPP[src/Simulator.cpp] --> C_H
    TS_CPP[tests/test_simulator.cpp] --> C_H
    TV_CPP[tests/test_validity.cpp] --> C_H
    GA_CPP[src/Genetic_Algorithm.cpp] --> GA_H
    MC_CPP[src/main.cpp] --> GA_H
    MC_CPP --> TGA_CPP[tests/test_GA.cpp]
  
```

- class CCircuit

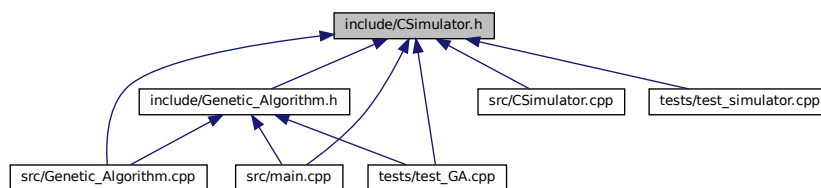
## 8.40 include/CSimulator.h File Reference

Generated by Doxygen

Include dependency graph for CSimulator.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [SimulationParameters](#)
- class [CSimulator](#)

## Functions

- double [evaluate\\_circuit](#) (int vector\_size, int \*circuit\_vector, const [SimulationParameters](#) &params, bool &is\_valid)  
*Evaluate the circuit.*
- double [evaluate\\_circuit\\_write](#) (int vector\_size, int \*circuit\_vector, double \*edge\_vector, const [SimulationParameters](#) &params)  
*Evaluate the circuit and save the circuit information in the given buffer.*

### 8.40.1 Function Documentation

**8.40.1.1 evaluate\_circuit()** `double evaluate_circuit (`  
    `int vector_size,`  
    `int * circuit_vector,`  
    `const SimulationParameters & params,`  
    `bool & is_valid )`

Evaluate the circuit.

#### Parameters

<i>vector_size</i>	Size of the circuit_vector.
<i>circuit_vector</i>	Vector representing the circuit.
<i>params</i>	Parameters for the evaluation.

#### Returns

Reward for the evaluation.

**8.40.1.2 evaluate\_circuit\_write()** `double evaluate_circuit_write (`  
    `int vector_size,`  
    `int * circuit_vector,`  
    `double * edge_vector,`  
    `const SimulationParameters & params )`

Evaluate the circuit and save the circuit information in the given buffer.

#### Parameters

<i>vector_size</i>	Size of the circuit_vector.
<i>edge_vector</i>	Save buffer
<i>circuit_vector</i>	Vector representing the circuit.
<i>params</i>	Parameters for the evaluation.

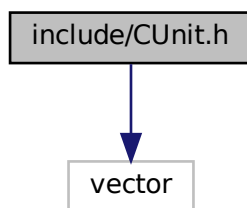
#### Returns

Reward for the evaluation.

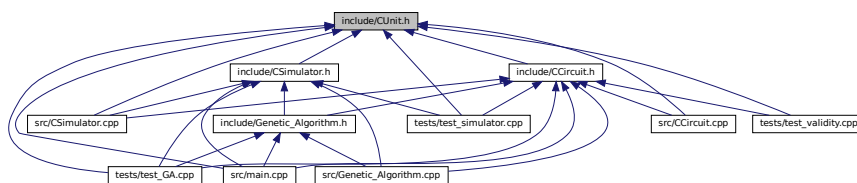
## 8.41 include/CUnit.h File Reference

```
#include <vector>
```

Include dependency graph for CUnit.h:



This graph shows which files directly or indirectly include this file:



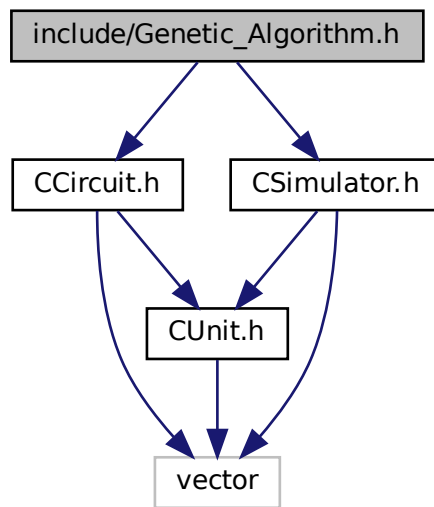
## Classes

- class [CUnit](#)

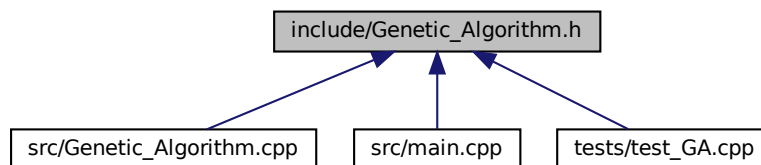
## 8.42 include/Genetic\_Algorithm.h File Reference

```
#include "CCircuit.h"  
#include "CSimulator.h"
```

Include dependency graph for Genetic\_Algorithm.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [Algorithm\\_Parameters](#)
- class [GA](#)

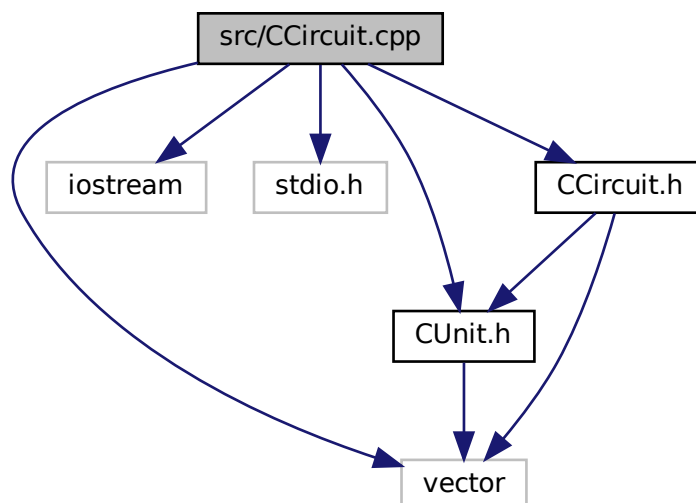
## 8.43 LICENSE.md File Reference

## 8.44 README.md File Reference

## 8.45 src/CCircuit.cpp File Reference

```
#include <vector>
#include <iostream>
```

```
#include <stdio.h>
#include "CUnit.h"
#include "CCircuit.h"
Include dependency graph for CCircuit.cpp:
```

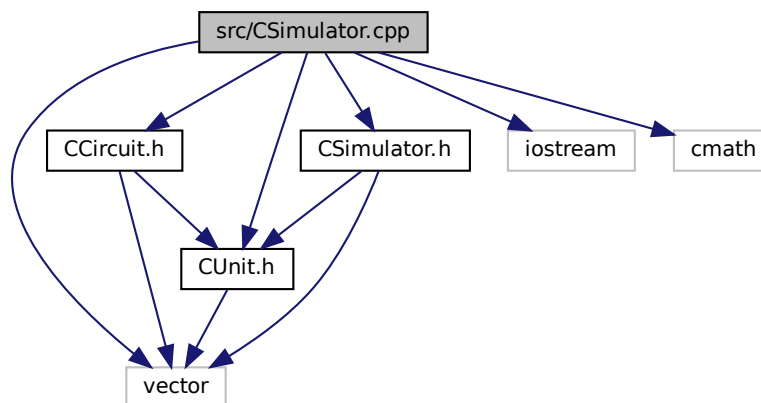


## 8.46 src/CSimulator.cpp File Reference

Main code file for the [CSimulator](#) class.

```
#include "CUnit.h"
#include "CCircuit.h"
#include "CSimulator.h"
#include <vector>
#include <iostream>
#include <cmath>
```

Include dependency graph for CSimulator.cpp:



## Functions

- double [evaluate\\_circuit](#) (int vector\_size, int \*circuit\_vector, const [SimulationParameters](#) &params, bool &is\_valid)  
*Evaluate the circuit.*
- double [evaluate\\_circuit\\_write](#) (int vector\_size, int \*circuit\_vector, double \*edge\_vector, const [SimulationParameters](#) &params)  
*Evaluate the circuit and save the circuit information in the given buffer.*

### 8.46.1 Detailed Description

Main code file for the [CSimulator](#) class.

### 8.46.2 Function Documentation

**8.46.2.1 [evaluate\\_circuit\(\)](#)** double [evaluate\\_circuit](#) (  
     int vector\_size,  
     int \* circuit\_vector,  
     const [SimulationParameters](#) & params,  
     bool & is\_valid )

Evaluate the circuit.

#### Parameters

<i>vector_size</i>	Size of the circuit_vector.
<i>circuit_vector</i>	Vector representing the circuit.
<i>params</i>	Parameters for the evaluation.



**Returns**

Reward for the evaluation.

**8.46.2.2 evaluate\_circuit\_write()** double evaluate\_circuit\_write (   
     int *vector\_size*,   
     int \* *circuit\_vector*,   
     double \* *edge\_vector*,   
     const [SimulationParameters](#) & *params* )

Evaluate the circuit and save the circuit information in the given buffer.

**Parameters**

<i>vector_size</i>	Size of the circuit_vector.
<i>edge_vector</i>	Save buffer
<i>circuit_vector</i>	Vector representing the circuit.
<i>params</i>	Parameters for the evaluation.

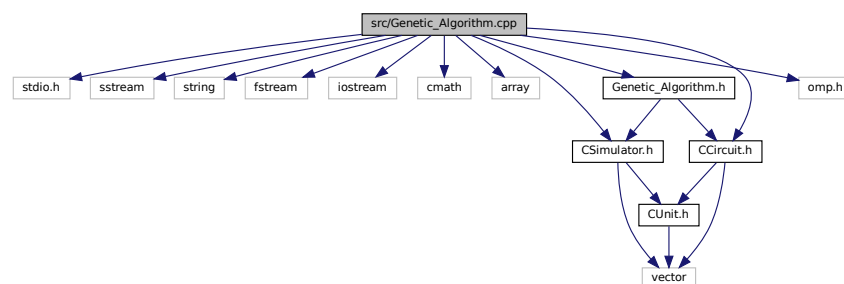
**Returns**

Reward for the evaluation.

**8.47 src/Genetic\_Algorithm.cpp File Reference**

```
#include <stdio.h>
#include <sstream>
#include <string>
#include <fstream>
#include <iostream>
#include <cmath>
#include <array>
#include "CSimulator.h"
#include "CCircuit.h"
#include "Genetic_Algorithm.h"
#include <omp.h>
```

Include dependency graph for Genetic\_Algorithm.cpp:



## Functions

- int [binary\\_search](#) (double \*list, double target, int size)  
*Perform a binary search.*
- int [partition](#) (double \*arr, int \*tailing\_val, int start, int end)  
*Helper function for sorting.*
- void [quickSort](#) (double \*arr, int \*tailing\_val, int start, int end)  
*Custom in-place sort, tailing\_val will be sort according to arr.*

### 8.47.1 Function Documentation

**8.47.1.1 [binary\\_search\(\)](#)** `int binary_search (`  
    `double * list,`  
    `double target,`  
    `int size )`

Perform a binary search.

#### Parameters

<i>list</i>	pointer of array of interest
<i>target</i>	the value wanted
<i>size</i>	size of the array

**8.47.1.2 [partition\(\)](#)** `int partition (`  
    `double * arr,`  
    `int * tailing_val,`  
    `int start,`  
    `int end )`

Helper function for sorting.

#### Parameters

<i>arr</i>	pointer to the array to sort
<i>tailing_val</i>	pointer to the array that should sort according to arr
<i>start</i>	starting index
<i>end</i>	ending index

#### Returns

pivoting index

**8.47.1.3 quickSort()** void quickSort (

```

    double * arr,
    int * tailing_val,
    int start,
    int end )

```

Custom in-place sort, tailing\_val will be sort according to arr.

#### Parameters

<i>arr</i>	pointer to the array to sort
<i>tailing_val</i>	pointer to the array that should sort according to arr
<i>start</i>	starting index
<i>end</i>	ending index

## 8.48 src/graph.py File Reference

### Namespaces

- [graph](#)

### Functions

- def [graph.read\\_inputs](#) (file\_path)

### Variables

- [graph.graph](#) = graphviz.Digraph()
- [graph.rankdir](#)
- [graph.shape](#)
- int [graph.index](#) = 1
- [graph.color](#)
- [graph.headport](#)
- [graph.tailport](#)
- [graph.arrowhead](#)
- [graph.width](#)
- [graph.height](#)
- [graph.copy\\_list](#) = int\_list.copy()
- [graph.largest](#) = max(copy\_list)
- [graph.second\\_largest](#) = max(copy\_list)
- string [graph.node\\_name](#) = 'Unit ' + str(i)
- string [graph.end\\_node](#) = 'Tailing'
- [graph.label](#)
- [graph.cleanup](#)
- [graph.True](#)
- [graph.format](#)

## 8.49 src/graph\_online.py File Reference

### Namespaces

- [graph\\_online](#)

### Functions

- [def graph\\_online.read\\_inputs](#) (file\_path)

### Variables

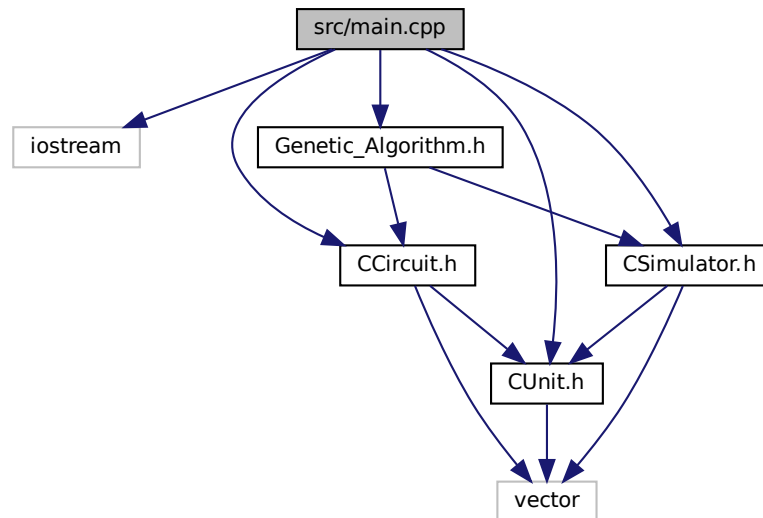
- [def graph\\_online.int\\_list](#) = [read\\_inputs](#)('./inputTest.dat')
- [graph\\_online.graph](#) = [graphviz.Digraph](#)()
- [graph\\_online.rankdir](#)
- [graph\\_online.shape](#)
- [int graph\\_online.index](#) = 1
- [graph\\_online.color](#)
- [graph\\_online.headport](#)
- [graph\\_online.tailport](#)
- [graph\\_online.arrowhead](#)
- [graph\\_online.width](#)
- [graph\\_online.height](#)
- [def graph\\_online.copy\\_list](#) = [int\\_list.copy](#)()
- [graph\\_online.largest](#) = [max](#)([copy\\_list](#))
- [graph\\_online.second\\_largest](#) = [max](#)([copy\\_list](#))
- [string graph\\_online.node\\_name](#) = 'Unit ' + [str](#)(i)
- [string graph\\_online.end\\_node](#) = 'Tailing'
- [graph\\_online.cleanup](#)
- [graph\\_online.True](#)
- [graph\\_online.format](#)

## 8.50 src/main.cpp File Reference

```
#include <iostream>
#include "CUnit.h"
#include "CCircuit.h"
#include "CSimulator.h"
```

```
#include "Genetic_Algorithm.h"
```

Include dependency graph for main.cpp:



## Functions

- int `main` (int argc, char \*argv[])

### 8.50.1 Function Documentation

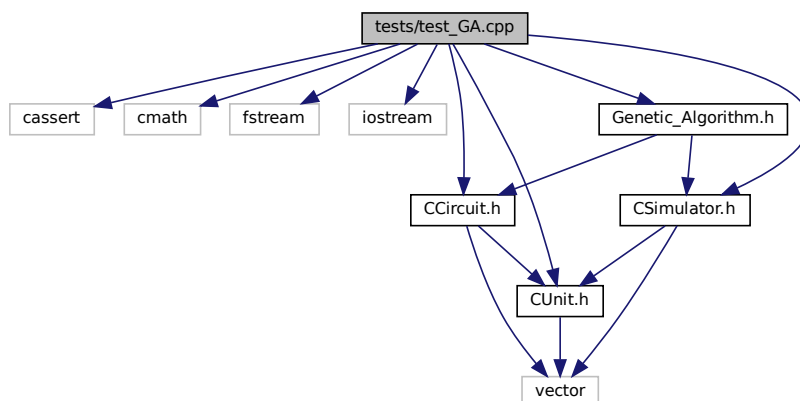
**8.50.1.1 main()** int main (  
    int argc,  
    char \* argv[] )

## 8.51 tests/test\_GA.cpp File Reference

```
#include <cassert>
#include <cmath>
#include <fstream>
#include <iostream>
#include "CUnit.h"
#include "CCircuit.h"
#include "CSimulator.h"
```

```
#include "Genetic_Algorithm.h"
```

Include dependency graph for test\_GA.cpp:



## Functions

- void [test\\_circuit5](#) ()  
Test [GA](#) performance on a circuit size of 5.
- void [test\\_circuit10](#) ()  
Test [GA](#) performance on a circuit size of 10.
- void [test\\_circuit20](#) ()  
Test [GA](#) performance on a circuit size of 20.
- int [main](#) ()  
Main function to run the tests.

### 8.51.1 Function Documentation

#### 8.51.1.1 `main()` `int main ( )`

Main function to run the tests.

This main function runs all the defined test functions.

#### 8.51.1.2 `test_circuit10()` `void test_circuit10 ( )`

Test [GA](#) performance on a circuit size of 10.

### 8.51.1.3 test\_circuit20() void test\_circuit20 ( )

Test GA performance on a circuit size of 20.

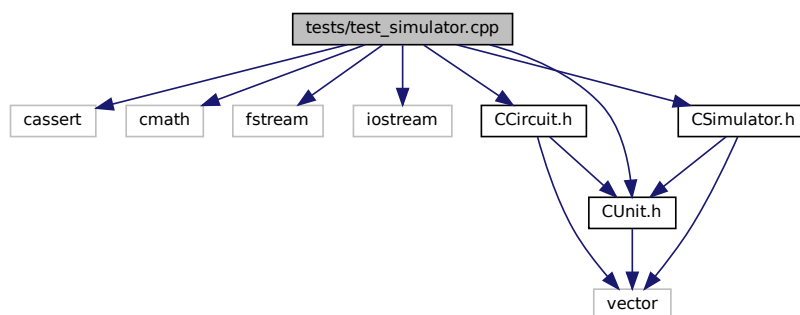
### 8.51.1.4 test\_circuit5() void test\_circuit5 ( )

Test GA performance on a circuit size of 5.

## 8.52 tests/test\_simulator.cpp File Reference

```
#include <cassert>
#include <cmath>
#include <fstream>
#include <iostream>
#include "CUnit.h"
#include "CCircuit.h"
#include "CSimulator.h"
```

Include dependency graph for test\_simulator.cpp:



## Functions

- void [test\\_instantiate\\_circuit\\_units](#) ( )  
*Test instantiation of circuit units.*
- bool [isClose](#) (double a, double b, double epsilon)  
*Check if two double values are close to each other within a certain epsilon.*
- void [test\\_calculate\\_reward](#) ( )  
*Test the reward calculation of the simulator.*
- void [test\\_small\\_circuit](#) ( )  
*Test a small circuit simulation.*
- int [main](#) ( )  
*Main function to run the tests.*

### 8.52.1 Function Documentation

**8.52.1.1 isClose()** `bool isClose (`  
    `double a,`  
    `double b,`  
    `double epsilon )`

Check if two double values are close to each other within a certain epsilon.

**Parameters**

<i>a</i>	First double value
<i>b</i>	Second double value
<i>epsilon</i>	Maximum allowed difference between a and b

**Returns**

True if  $|a - b| < \text{epsilon}$ , false otherwise

**8.52.1.2 main()** `int main ( )`

Main function to run the tests.

This main function runs all the defined test functions.

**8.52.1.3 test\_calculate\_reward()** `void test_calculate_reward ( )`

Test the reward calculation of the simulator.

This function initializes simulation parameters and creates a [CSimulator](#) instance to verify the correct calculation of the reward value.

**8.52.1.4 test\_instantiate\_circuit\_units()** `void test_instantiate_circuit_units ( )`

Test instantiation of circuit units.

This function sets up simulation parameters and creates a [CSimulator](#) instance to verify correct instantiation and behavior of the circuit units.

**8.52.1.5 test\_small\_circuit()** `void test_small_circuit ( )`

Test a small circuit simulation.

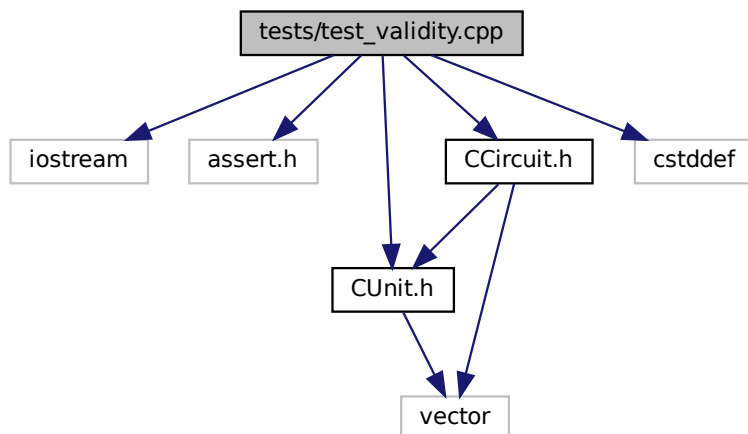
This function sets up simulation parameters for a small circuit, creates a [CSimulator](#) instance and verifies the correct calculation of total concentrate and tail flows.



## 8.53 tests/test\_validity.cpp File Reference

```
#include <iostream>
#include <assert.h>
#include "CUnit.h"
#include "CCircuit.h"
#include <cstdlib>
```

Include dependency graph for test\_validity.cpp:



### Functions

- `template<std::size_t N>`  
`void test_circuit (int(&vec)[N], bool expected_validity)`  
*Function to test circuit validity.*
- `int main (int argc, char *argv[])`  
*Main function.*

#### 8.53.1 Function Documentation

**8.53.1.1 main()** `int main (`  
`int argc,`  
`char * argv[] )`

Main function.

This function includes test cases to check the validity of a circuit. Tests are conducted on both valid and invalid circuit configurations.

**8.53.1.2 test\_circuit()** `template<std::size_t N>`  
`void test_circuit (`  
`int(&) vec[N],`  
`bool expected_validity )`

Function to test circuit validity.

**Parameters**

<i>vec</i>	Vector of integers representing a circuit.
<i>expected_validity</i>	Expected result of circuit validity check.

This function creates a [CCircuit](#) object, checks the circuit validity and validates the result using assert.

## Index

- \_\_pad0\_\_
  - CMakeCache.txt, [34](#)
- ~CCircuit
  - CCircuit, [13](#)
- ~CUnit
  - CUnit, [17](#)
- ~GA
  - GA, [22](#)
- add\_library
  - CMakeLists.txt, [54](#)
- Algorithm\_Parameters, [9](#)
  - circuit\_size, [10](#)
  - communicate\_interval, [10](#)
  - crossover\_rate, [10](#)
  - max\_iter\_before\_fail, [10](#)
  - max\_iter\_without\_progress, [10](#)
  - max\_iterations, [11](#)
  - mutation\_rate, [11](#)
  - mutation\_rate\_increase\_factor, [11](#)
  - mutation\_scheme, [11](#)
  - mutation\_size, [11](#)
  - parallel\_mpi, [11](#)
  - parent\_comm\_size\_parallel, [11](#)
  - parent\_pool\_size, [11](#)
  - population\_size, [11](#)
  - selection\_scheme, [11](#)
  - tol, [11](#)
  - tournament\_size, [12](#)
  - tournament\_size\_parallel, [12](#)
  - write\_interval, [12](#)
- ARCHITECTURE\_ID
  - CMakeCCompilerId.c, [35](#), [37](#)
  - CMakeCXXCompilerId.cpp, [39](#), [41](#)
- arrowhead
  - graph, [5](#)
  - graph\_online, [7](#)
- best\_circuit
  - GA, [29](#)
- best\_fitness
  - GA, [29](#)
- binary\_search
  - Genetic\_Algorithm.cpp, [63](#)
- build/CMakeCache.txt, [34](#)
- build/CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c,  
[34](#)
- build/CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp,  
[39](#)
- build/CMakeFiles/CMakeRuleHashes.txt, [43](#)
- build/CMakeFiles/FindMPI/test\_mpi.cpp, [43](#)
- build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c,  
[45](#)
- build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp,  
[47](#)
- build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c, [49](#)
- build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp,  
[50](#)
- build/CMakeFiles/TargetDirectories.txt, [51](#)
- build/DartConfiguration.tcl, [51](#)
- build/src/CMakeFiles/Circuit\_Optimizer.dir/link.txt, [51](#)
- build/src/CMakeFiles/circuitSimulator.dir/link.txt, [52](#)
- build/src/CMakeFiles/geneticAlgorithm.dir/link.txt, [52](#)
- build/Testing/Temporary/CTestCostData.txt, [54](#)
- build/tests/CMakeFiles/simulator\_tests.dir/link.txt, [52](#)
- build/tests/CMakeFiles/test\_GA.dir/link.txt, [52](#)
- build/tests/CMakeFiles/test\_simulator.dir/link.txt, [52](#)
- build/tests/CMakeFiles/test\_validity.dir/link.txt, [53](#)
- C\_DIALECT
  - CMakeCCompilerId.c, [35](#), [37](#)
- c\_gen\_flow\_out
  - CUnit, [17](#)
- c\_waste\_flow\_out
  - CUnit, [17](#)
- calculate\_reward
  - CSimulator, [15](#)
- CCircuit, [12](#)
  - ~CCircuit, [13](#)
  - CCircuit, [12](#)
  - check\_validity, [13](#)
- check\_same
  - GA, [22](#)
- check\_validity
  - CCircuit, [13](#)
- circuit\_size
  - Algorithm\_Parameters, [10](#)
- cleanup
  - graph, [5](#)
  - graph\_online, [8](#)
- CMakeCache.txt
  - \_\_pad0\_\_, [34](#)
  - MPI\_CXX\_LINK\_FLAGS, [34](#)
  - OpenMP\_C\_FLAGS, [34](#)
  - OpenMP\_gomp\_LIBRARY, [34](#)
- CMakeCCompilerId.c
  - ARCHITECTURE\_ID, [35](#), [37](#)
  - C\_DIALECT, [35](#), [37](#)
  - COMPILER\_ID, [35](#), [37](#)
  - DEC, [35](#), [37](#)
  - HEX, [35](#), [37](#)
  - info\_arch, [36](#), [38](#)
  - info\_compiler, [36](#), [38](#)
  - info\_language\_dialect\_default, [36](#), [38](#)
  - info\_platform, [36](#), [38](#)
  - main, [36](#), [38](#)
  - PLATFORM\_ID, [35](#), [37](#)
  - STRINGIFY, [35](#), [38](#)
  - STRINGIFY\_HELPER, [36](#), [38](#)
- CMakeCXXCompilerId.cpp
  - ARCHITECTURE\_ID, [39](#), [41](#)

- COMPILER\_ID, 39, 41
- CXX\_STD, 39, 41
- DEC, 39, 42
- HEX, 39, 42
- info\_arch, 40, 43
- info\_compiler, 40, 43
- info\_language\_dialect\_default, 40, 43
- info\_platform, 41, 43
- main, 40, 42
- PLATFORM\_ID, 40, 42
- STRINGIFY, 40, 42
- STRINGIFY\_HELPER, 40, 42
- CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c, 36
- CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp, 41
- CMakeFiles/CMakeRuleHashes.txt, 43
- CMakeFiles/FindMPI/test\_mpi.cpp, 44
- CMakeFiles/FindOpenMP/OpenMPCheckVersion.c, 46
- CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp, 48
- CMakeFiles/FindOpenMP/OpenMPTryFlag.c, 49
- CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp, 51
- CMakeFiles/TargetDirectories.txt, 51
- CMakeLists.txt, 54
  - add\_library, 54
  - project, 54
- color
  - graph, 5
  - graph\_online, 8
- communicate\_interval
  - Algorithm\_Parameters, 10
- COMPILER\_ID
  - CMakeCCompilerId.c, 35, 37
  - CMakeCXXCompilerId.cpp, 39, 41
- concentrate\_feed\_list
  - CUnit, 17
- concentrate\_list
  - CUnit, 18
- copy\_list
  - graph, 5
  - graph\_online, 8
- copy\_parent\_to\_child
  - GA, 22
- crossover\_rate
  - Algorithm\_Parameters, 10
- CSimulator, 14
  - calculate\_reward, 15
  - CSimulator, 14
  - get\_final\_concentrate\_list, 15
  - get\_final\_tail\_list, 15
  - get\_total\_concentrate\_ger, 15
  - get\_total\_concentrate\_waste, 15
  - get\_total\_tail\_ger, 15
  - get\_total\_tail\_waste, 15
  - get\_units\_vector, 15
  - solve, 15
  - write\_vector\_values\_to\_array, 16
- CSimulator.cpp
  - evaluate\_circuit, 61
  - evaluate\_circuit\_write, 62
- CSimulator.h
  - evaluate\_circuit, 56
  - evaluate\_circuit\_write, 57
- CUnit, 16
  - ~CUnit, 17
  - c\_ger\_flow\_out, 17
  - c\_waste\_flow\_out, 17
  - concentrate\_feed\_list, 17
  - concentrate\_list, 18
  - CUnit, 17
  - ger\_flow\_in, 18
  - ger\_flow\_in\_old, 18
  - ger\_recovery, 18
  - get\_is\_feed, 17
  - get\_unit\_id, 17
  - t\_ger\_flow\_out, 18
  - t\_waste\_flow\_out, 18
  - tails\_feed\_list, 18
  - tails\_list, 18
  - waste\_flow\_in, 18
  - waste\_flow\_in\_old, 18
  - waste\_recovery, 18
- cur\_iter
  - GA, 29
- CXX\_STD
  - CMakeCXXCompilerId.cpp, 39, 41
- DEC
  - CMakeCCompilerId.c, 35, 37
  - CMakeCXXCompilerId.cpp, 39, 42
- end\_node
  - graph, 5
  - graph\_online, 8
- evaluate\_circuit
  - CSimulator.cpp, 61
  - CSimulator.h, 56
- evaluate\_circuit\_write
  - CSimulator.cpp, 62
  - CSimulator.h, 57
- F0\_ger
  - SimulationParameters, 32
- F0\_waste
  - SimulationParameters, 32
- file\_write\_buffer
  - GA, 29
- fitness\_parents
  - GA, 29
- fitness\_population
  - GA, 30
- format
  - graph, 5
  - graph\_online, 8
- GA, 19

- ~GA, 22
- best\_circuit, 29
- best\_fitness, 29
- check\_same, 22
- copy\_parent\_to\_child, 22
- cur\_iter, 29
- file\_write\_buffer, 29
- fitness\_parents, 29
- fitness\_population, 30
- GA, 21
- generalized\_copy, 22
- generate\_child, 23
- get\_idx, 23
- get\_idx\_node, 23
- get\_node\_num, 24
- in\_place\_sort, 24
- mate\_count\_all, 30
- mpi\_reduction\_wrapper, 24
- mutation, 24
- not\_improving\_count, 30
- nottouched, 30
- old\_best\_fitness, 30
- optimize, 26
- optimize\_parallel, 26
- optimize\_without\_mpi, 26
- parameters, 30
- parents, 30
- population, 30
- population\_idx\_list, 31
- rank\_func, 26
- rank\_selection, 26
- rank\_sum, 31
- rank\_val, 31
- reinitialize\_pil, 26
- roulette\_selection, 27
- roulette\_selection\_parallel, 27
- roulette\_sum, 31
- same\_mate, 31
- select\_mate, 27
- selection, 27
- set\_best\_to\_parent, 27
- setup, 27
- setup\_parallel, 27
- sim\_params, 31
- size\_per\_cir, 31
- tournament\_selection, 27
- tournament\_selection\_parallel, 28
- two\_point\_crossover, 28
- update\_best, 29
- vec\_size\_parent, 31
- vec\_size\_population, 32
- write\_to\_file, 29
- generalized\_copy
  - GA, 22
- generate\_child
  - GA, 23
- Genetic\_Algorithm.cpp
  - binary\_search, 63
  - partition, 63
  - quickSort, 63
- ger\_flow\_in
  - CUnit, 18
- ger\_flow\_in\_old
  - CUnit, 18
- ger\_recovery
  - CUnit, 18
- ger\_reward
  - SimulationParameters, 33
- get\_final\_concentrate\_list
  - CSimulator, 15
- get\_final\_tail\_list
  - CSimulator, 15
- get\_idx
  - GA, 23
- get\_idx\_node
  - GA, 23
- get\_is\_feed
  - CUnit, 17
- get\_node\_num
  - GA, 24
- get\_total\_concentrate\_ger
  - CSimulator, 15
- get\_total\_concentrate\_waste
  - CSimulator, 15
- get\_total\_tail\_ger
  - CSimulator, 15
- get\_total\_tail\_waste
  - CSimulator, 15
- get\_unit\_id
  - CUnit, 17
- get\_units\_vector
  - CSimulator, 15
- graph, 4
  - arrowhead, 5
  - cleanup, 5
  - color, 5
  - copy\_list, 5
  - end\_node, 5
  - format, 5
  - graph, 6
  - graph\_online, 8
  - headport, 6
  - height, 6
  - index, 6
  - label, 6
  - largest, 6
  - node\_name, 6
  - rankdir, 6
  - read\_inputs, 5
  - second\_largest, 6
  - shape, 6
  - tailport, 6
  - True, 7
  - width, 7
- graph\_online, 7
  - arrowhead, 7

- cleanup, 8
  - color, 8
  - copy\_list, 8
  - end\_node, 8
  - format, 8
  - graph, 8
  - headport, 8
  - height, 8
  - index, 8
  - int\_list, 8
  - largest, 8
  - node\_name, 9
  - rankdir, 9
  - read\_inputs, 7
  - second\_largest, 9
  - shape, 9
  - tailport, 9
  - True, 9
  - width, 9
- headport
- graph, 6
  - graph\_online, 8
- height
- graph, 6
  - graph\_online, 8
- HEX
- CMakeCCompilerId.c, 35, 37
  - CMakeCXXCompilerId.cpp, 39, 42
- in\_place\_sort
- GA, 24
- include/CCircuit.h, 55
- include/CSimulator.h, 55
- include/CUnit.h, 57
- include/Genetic\_Algorithm.h, 58
- index
- graph, 6
  - graph\_online, 8
- info\_arch
- CMakeCCompilerId.c, 36, 38
  - CMakeCXXCompilerId.cpp, 40, 43
- info\_compiler
- CMakeCCompilerId.c, 36, 38
  - CMakeCXXCompilerId.cpp, 40, 43
- info\_language\_dialect\_default
- CMakeCCompilerId.c, 36, 38
  - CMakeCXXCompilerId.cpp, 40, 43
- info\_platform
- CMakeCCompilerId.c, 36, 38
  - CMakeCXXCompilerId.cpp, 41, 43
- int\_list
- graph\_online, 8
- isClose
- test\_simulator.cpp, 68
- k\_ger
- SimulationParameters, 33
- k\_waste
- SimulationParameters, 33
- label
- graph, 6
- largest
- graph, 6
  - graph\_online, 8
- LICENSE.md, 59
- link.txt
- rpath, 51–53
  - WI, 52, 53
- main
- CMakeCCompilerId.c, 36, 38
  - CMakeCXXCompilerId.cpp, 40, 42
  - main.cpp, 66
  - OpenMPCheckVersion.c, 45, 46
  - OpenMPCheckVersion.cpp, 47, 48
  - OpenMPTryFlag.c, 49, 50
  - OpenMPTryFlag.cpp, 50, 51
  - test\_GA.cpp, 67
  - test\_mpi.cpp, 44
  - test\_simulator.cpp, 69
  - test\_validity.cpp, 70
- main.cpp
- main, 66
- mate\_count\_all
- GA, 30
- max\_iter
- SimulationParameters, 33
- max\_iter\_before\_fail
- Algorithm\_Parameters, 10
- max\_iter\_without\_progress
- Algorithm\_Parameters, 10
- max\_iterations
- Algorithm\_Parameters, 11
- min\_flowrate
- SimulationParameters, 33
- MPI\_CXX\_LINK\_FLAGS
- CMakeCache.txt, 34
- mpi\_reduction\_wrapper
- GA, 24
- mutation
- GA, 24
- mutation\_rate
- Algorithm\_Parameters, 11
- mutation\_rate\_increase\_factor
- Algorithm\_Parameters, 11
- mutation\_scheme
- Algorithm\_Parameters, 11
- mutation\_size
- Algorithm\_Parameters, 11
- node\_name
- graph, 6
  - graph\_online, 9
- not\_improving\_count
- GA, 30
- nottouched

- GA, 30
- old\_best\_fitness
  - GA, 30
- ompver\_str
  - OpenMPCheckVersion.c, 45, 46
  - OpenMPCheckVersion.cpp, 47, 48
- OpenMP\_C\_FLAGS
  - CMakeCache.txt, 34
- OpenMP\_gomp\_LIBRARY
  - CMakeCache.txt, 34
- OpenMPCheckVersion.c
  - main, 45, 46
  - ompver\_str, 45, 46
- OpenMPCheckVersion.cpp
  - main, 47, 48
  - ompver\_str, 47, 48
- OpenMPTryFlag.c
  - main, 49, 50
- OpenMPTryFlag.cpp
  - main, 50, 51
- optimize
  - GA, 26
- optimize\_parallel
  - GA, 26
- optimize\_without\_mpi
  - GA, 26
- parallel\_mpi
  - Algorithm\_Parameters, 11
- parameters
  - GA, 30
- parent\_comm\_size\_parallel
  - Algorithm\_Parameters, 11
- parent\_pool\_size
  - Algorithm\_Parameters, 11
- parents
  - GA, 30
- partition
  - Genetic\_Algorithm.cpp, 63
- phi
  - SimulationParameters, 33
- PLATFORM\_ID
  - CMakeCCompilerId.c, 35, 37
  - CMakeCXXCompilerId.cpp, 40, 42
- population
  - GA, 30
- population\_idx\_list
  - GA, 31
- population\_size
  - Algorithm\_Parameters, 11
- project
  - CMakeLists.txt, 54
- quickSort
  - Genetic\_Algorithm.cpp, 63
- rank\_func
  - GA, 26
- rank\_selection
  - GA, 26
- rank\_sum
  - GA, 31
- rank\_val
  - GA, 31
- rankdir
  - graph, 6
  - graph\_online, 9
- read\_inputs
  - graph, 5
  - graph\_online, 7
- README.md, 59
- reinitialize\_pil
  - GA, 26
- rho
  - SimulationParameters, 33
- roulette\_selection
  - GA, 27
- roulette\_selection\_parallel
  - GA, 27
- roulette\_sum
  - GA, 31
- rpath
  - link.txt, 51–53
- same\_mate
  - GA, 31
- second\_largest
  - graph, 6
  - graph\_online, 9
- select\_mate
  - GA, 27
- selection
  - GA, 27
- selection\_scheme
  - Algorithm\_Parameters, 11
- set\_best\_to\_parent
  - GA, 27
- setup
  - GA, 27
- setup\_parallel
  - GA, 27
- shape
  - graph, 6
  - graph\_online, 9
- sim\_params
  - GA, 31
- SimulationParameters, 32
  - F0\_ger, 32
  - F0\_waste, 32
  - ger\_reward, 33
  - k\_ger, 33
  - k\_waste, 33
  - max\_iter, 33
  - min\_flowrate, 33
  - phi, 33
  - rho, 33
  - volume, 33

- waste\_reward, 33
- size\_per\_cir
  - GA, 31
- solve
  - CSimulator, 15
- src/CCircuit.cpp, 59
- src/CMakeFiles/Circuit\_Optimizer.dir/link.txt, 53
- src/CMakeFiles/circuitSimulator.dir/link.txt, 54
- src/CMakeFiles/geneticAlgorithm.dir/link.txt, 54
- src/CMakeLists.txt, 54
- src/CSimulator.cpp, 60
- src/Genetic\_Algorithm.cpp, 62
- src/graph.py, 64
- src/graph\_online.py, 65
- src/main.cpp, 65
- STRINGIFY
  - CMakeCCompilerId.c, 35, 38
  - CMakeCXXCompilerId.cpp, 40, 42
- STRINGIFY\_HELPER
  - CMakeCCompilerId.c, 36, 38
  - CMakeCXXCompilerId.cpp, 40, 42
- t\_get\_flow\_out
  - CUnit, 18
- t\_waste\_flow\_out
  - CUnit, 18
- tailport
  - graph, 6
  - graph\_online, 9
- tails\_feed\_list
  - CUnit, 18
- tails\_list
  - CUnit, 18
- test\_calculate\_reward
  - test\_simulator.cpp, 69
- test\_circuit
  - test\_validity.cpp, 70
- test\_circuit10
  - test\_GA.cpp, 67
- test\_circuit20
  - test\_GA.cpp, 67
- test\_circuit5
  - test\_GA.cpp, 68
- test\_GA.cpp
  - main, 67
  - test\_circuit10, 67
  - test\_circuit20, 67
  - test\_circuit5, 68
- test\_instantiate\_circuit\_units
  - test\_simulator.cpp, 69
- test\_mpi.cpp
  - main, 44
- test\_simulator.cpp
  - isClose, 68
  - main, 69
  - test\_calculate\_reward, 69
  - test\_instantiate\_circuit\_units, 69
  - test\_small\_circuit, 69
- test\_small\_circuit
  - test\_simulator.cpp, 69
- test\_validity.cpp
  - main, 70
  - test\_circuit, 70
- Testing/Temporary/CTestCostData.txt, 54
- tests/CMakeFiles/simulator\_tests.dir/link.txt, 54
- tests/CMakeFiles/test\_simulator.dir/link.txt, 54
- tests/CMakeFiles/test\_validity.dir/link.txt, 54
- tests/CMakeLists.txt, 54
- tests/test\_GA.cpp, 66
- tests/test\_simulator.cpp, 68
- tests/test\_validity.cpp, 70
- tol
  - Algorithm\_Parameters, 11
- tournament\_selection
  - GA, 27
- tournament\_selection\_parallel
  - GA, 28
- tournament\_size
  - Algorithm\_Parameters, 12
- tournament\_size\_parallel
  - Algorithm\_Parameters, 12
- True
  - graph, 7
  - graph\_online, 9
- two\_point\_crossover
  - GA, 28
- update\_best
  - GA, 29
- vec\_size\_parent
  - GA, 31
- vec\_size\_population
  - GA, 32
- volume
  - SimulationParameters, 33
- waste\_flow\_in
  - CUnit, 18
- waste\_flow\_in\_old
  - CUnit, 18
- waste\_recovery
  - CUnit, 18
- waste\_reward
  - SimulationParameters, 33
- width
  - graph, 7
  - graph\_online, 9
- WI
  - link.txt, 52, 53
- write\_interval
  - Algorithm\_Parameters, 12
- write\_to\_file
  - GA, 29
- write\_vector\_values\_to\_array
  - CSimulator, 16