

BGL evolution

Jeremy W. Murphy

A brief introduction to me

- Used BGL 2013-2016
 - Representing detailed residential fibre-optic network designs
- Started work as a maintainer in 2021
 - Primarily to help get bug fixes merged.
- Day job: automotive test rigs.
 - "Drive Toyota's transformation into a mobility company"

BGL's role

- The BGL has not changed in 25 years...

BGL's role



- The BGL has not changed in 25 years...
- but the landscape around it has:
 - New graph libraries: NW Graph, CXXGraph, Graaf, etc.
 - Proposal to the standard library underway.
 - Parallelism has become the norm

BGL's role

- The BGL has not changed in 25 years...
- but the landscape around it has:
 - New graph libraries: NW Graph, CXXGraph, Graaf, etc.
 - Proposal to the standard library underway.
 - Parallelism has become the norm
- How can the BGL stay relevant and provide great value?
 - Experienced vs new users

BGL's role

Compared to Boost Graph Library



	CXXGraph 	Boost Graph 
Setup time	Seconds	Minutes–Hours
API readability	High	Low
Boilerplate	Minimal	Heavy
Modern C++ style	Yes	Mixed/Legacy
Learning curve	Low	High

CXXGraph focuses on **developer experience first**.

<https://github.com/ZigRazor/CXXGraph>, 4a23155

BGL's role

Compared to Boost Graph Library

	CXXGraph 	Boost Graph 
Setup time	Seconds	Minutes–Hours
API readability	High	Low
Boilerplate	Minimal	Heavy
Modern C++ style	Yes	Mixed/Legacy
Learning curve	Low	High

CXXGraph focuses on **developer experience first**.

<https://github.com/ZigRazor/CXXGraph>, 4a23155

Can we improve on these metrics without compromising our values?

BGL's role

Compared to Boost Graph Library

	CXXGraph 	Boost Graph 
Setup time	Seconds	Minutes–Hours
API readability	High	Low
Boilerplate	Minimal	Heavy
Modern C++ style	Yes	Mixed/Legacy
Learning curve	Low	High

CXXGraph focuses on **developer experience first**.

<https://github.com/ZigRazor/CXXGraph>, 4a23155

Can we improve on these metrics without compromising our values?

- Flexible
- Generic
- Orthogonal
- Scalable
- Powerful

My Goals for the BGL

- Modernization
- Extension
- Collaboration

Modernization

Modernization

- Parallelization
 - Deprecate separate Parallel BGL library
 - Create parallel overloads of algorithms in the BGL, following the example of the standard.

Modernization

- Parallelization
 - Deprecate separate Parallel BGL library
 - Create parallel overloads of algorithms in the BGL, following the example of the standard.

before:

```
template <class Graph, class Buffer, class BFSVisitor, class ColorMap>
void breadth_first_search(const Graph& g, typename graph_traits<Graph>::vertex_descriptor s,
Buffer& Q, BFSVisitor vis, ColorMap color);
```

after:

```
template <class ExecutionPolicy, class Graph, class Buffer, class BFSVisitor, class ColorMap>
void breadth_first_search(ExecutionPolicy &&policy, const Graph& g, typename
graph_traits<Graph>::vertex_descriptor s, Buffer& Q, BFSVisitor vis, ColorMap color);
```

Modernization

- Parallelization
 - Deprecate separate Parallel BGL library
 - Create parallel overloads of algorithms in the BGL, following the example of the standard.

ExecutionPolicy:

- `std::threads`

Optional dependencies:

- HPX
- CUDA
- etc

Modernization

- Parallelization
- Allocators
- Replace named parameters with strong types

Extension

- Cardinal tree concept and structure
 - e.g. binary tree
- Nested namespaces for experiments.
- More algorithms
 - Variations on existing problems
 - Different problems (automorphism?)

Collaboration

Collaboration, Story #1: Bug in Stoer Wagner #286



etiennedeg opened on Feb 16, 2022

MWE:

Collaboration, Story #1: Bug in Stoer Wagner #286



etiennedeg opened on Feb 16, 2022

MWE:



jeremy-murphy on Feb 20, 2022

Thank you for finding and documenting this.

Could you open a pull request that adds this case as a unit test? Are you able to fix the implementation too?

Collaboration, Story #1: Bug in Stoer Wagner #286



etiennedeg opened on Feb 16, 2022

MWE:



jeremy-murphy on Feb 20, 2022

Thank you for finding and documenting this.

Could you open a pull request that adds this case as a unit test? Are you able to fix the implementation too?



etiennedeg on Jun 3, 2022

Sorry for the delay, I was a bit busy these days, I made a PR for a non-regression test.

Collaboration, Story #1: Bug in Stoer Wagner #286



etiennedeg opened on Feb 16, 2022

MWE:



jeremy-murphy on Feb 20, 2022

Thank you for finding and documenting this.

Could you open a pull request that adds this case as a unit test? Are you able to fix the implementation too?



etiennedeg on Jun 3, 2022

Sorry for the delay, I was a bit busy these days, I made a PR for a non-regression test.

Collaboration, Story #1: Bug in Stoer Wagner #286



etiennedeg opened on Feb 16, 2022

MWE:



jeremy-murphy on Feb 20, 2022

Thank you for finding and documenting this.

Could you open a pull request that adds this case as a unit test? Are you able to fix the implementation too?



etiennedeg on Jun 3, 2022

Sorry for the delay, I was a bit busy these days, I made a PR for a non-regression test.



sebrockm on Jul 12, 2022

Contributor ...

@etiennedeg

I was hunting down a bug in one of my project for weeks. Finally, I stumbled over this issue and it really is the root cause for my problem! Thank you so much for finding this!!!

@jeremy-murphy Based on **@etiennedeg**'s test in [#294](#) I created [#296](#) which passes that test. Unfortunately,

[http:](#)

Collaboration, Story #2: Boost.Geometry



Hermann-SW opened on Oct 5, 2024 · edited by Hermann-SW

Edits ▾

Contribu

I experienced "is_straight_line_drawing()" returning `false` for valid straight line drawing of a maximal planar graph with 10million vertices first:

Collaboration, Story #2: Boost.Geometry



Hermann-SW opened on Oct 5, 2024 · edited by Hermann-SW

Edits ▾

Contribu

I experienced "is_straight_line_drawing()" returning `false` for valid straight line drawing of a maximal planar graph with 10million vertices first:



jeremy-murphy commented on Dec 28, 2024 · edited ▾

Member

⋮

Fix the geometry bug uncovered and documented by [@Hermann-SW](#) by replacing the custom function with one from Boost.Geometry.

Collaboration, Story #2: Boost.Geometry



Hermann-SW opened on Oct 5, 2024 · edited by Hermann-SW

Edits ▾

Contribu

I experienced "is_straight_line_drawing()" returning `false` for valid straight line drawing of a maximal planar graph with 10million vertices first:



jeremy-murphy commented on Dec 28, 2024 · edited ▾

Member

⋮

Fix the geometry bug uncovered and documented by [@Hermann-SW](#) by replacing the custom function with one from Boost.Geometry.

Traversal rewrite #1366

 Closed

barendgehrels started this conversation in **Show and tell**



barendgehrels on Jan 12, 2025

Maintainer

Collaboration, Story #2: Boost.Geometry



tinko92 mentioned this on Nov 29, 2025



Robust, integer crosses for is_straight_line_drawing graph#442



jeremy-murphy on Dec 1, 2025

@grafikrobot FYI, this is now resolved thanks to @tinko92. Graph won't depend on Geometry.

Collaboration

Objectives:

Collaboration

Objectives:

1. User \longrightarrow Contributor

Collaboration

Objectives:


1. User \longrightarrow Contributor
2. Contributor \longrightarrow Maintainer

Collaboration

Objectives:

1. User → Contributor

2. Contributor → Maintainer

3.  `jeremy-murphy` →



Thank you

Thank you

- The BGL creators
 - For an awesome library.

Thank you

- The BGL creators
 - For an awesome library.
- Arno
 - For organizing this workshop.

Thank you

- The BGL creators
 - For an awesome library.
- Arno
 - For organizing this workshop.
- The C++ Alliance
 - For supporting C++.

Thank you

- The BGL creators
 - For an awesome library.
- Arno
 - For organizing this workshop.
- The C++ Alliance
 - For supporting C++.
- You
 - For participating!