

# JSSP Solver

iota

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	Ui Namespace Reference . . . . .	9
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	Arrow Class Reference . . . . .	11
6.1.1	Detailed Description . . . . .	12
6.1.2	Member Enumeration Documentation . . . . .	12
6.1.2.1	anonymous enum . . . . .	12
6.1.3	Constructor & Destructor Documentation . . . . .	13
6.1.3.1	Arrow() [1/2] . . . . .	13
6.1.3.2	Arrow() [2/2] . . . . .	13
6.1.4	Member Function Documentation . . . . .	14
6.1.4.1	boundingRect() . . . . .	14
6.1.4.2	endPoint() . . . . .	14

6.1.4.3	<a href="#">paint()</a>	14
6.1.4.4	<a href="#">setEndPoint()</a>	15
6.1.4.5	<a href="#">setStartPoint()</a>	15
6.1.4.6	<a href="#">shape()</a>	16
6.1.4.7	<a href="#">startPoint()</a>	16
6.1.4.8	<a href="#">type()</a>	16
6.1.5	<a href="#">Member Data Documentation</a>	16
6.1.5.1	<a href="#">private_arrowHead</a>	16
6.1.5.2	<a href="#">private_endPoint</a>	16
6.1.5.3	<a href="#">private_startPoint</a>	17
6.2	<a href="#">BLIST Struct Reference</a>	17
6.2.1	<a href="#">Detailed Description</a>	17
6.2.2	<a href="#">Member Data Documentation</a>	17
6.2.2.1	<a href="#">machine</a>	17
6.2.2.2	<a href="#">makespan</a>	17
6.2.2.3	<a href="#">order</a>	18
6.3	<a href="#">Fixer Struct Reference</a>	18
6.3.1	<a href="#">Detailed Description</a>	18
6.3.2	<a href="#">Member Data Documentation</a>	18
6.3.2.1	<a href="#">duration</a>	18
6.3.2.2	<a href="#">machine</a>	18
6.3.2.3	<a href="#">starttime</a>	19
6.4	<a href="#">GanttChartBase Class Reference</a>	19
6.4.1	<a href="#">Detailed Description</a>	20
6.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	20
6.4.2.1	<a href="#">GanttChartBase()</a>	20
6.4.3	<a href="#">Member Function Documentation</a>	20
6.4.3.1	<a href="#">boundingRect()</a>	21
6.4.3.2	<a href="#">machineOffset()</a>	21
6.4.3.3	<a href="#">operationPosition()</a>	21

6.4.3.4	<a href="#">paint()</a>	21
6.4.4	<a href="#">Member Data Documentation</a>	22
6.4.4.1	<a href="#">machineHeight</a>	22
6.4.4.2	<a href="#">machineHorizontalOffset</a>	22
6.4.4.3	<a href="#">makespan</a>	22
6.4.4.4	<a href="#">operationHeight</a>	22
6.4.4.5	<a href="#">widthUnit</a>	23
6.5	<a href="#">GanttChartMachine Class Reference</a>	23
6.5.1	<a href="#">Detailed Description</a>	24
6.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	24
6.5.2.1	<a href="#">GanttChartMachine()</a>	24
6.5.2.2	<a href="#">~GanttChartMachine()</a>	25
6.5.3	<a href="#">Member Function Documentation</a>	25
6.5.3.1	<a href="#">boundingRect()</a>	25
6.5.3.2	<a href="#">paint()</a>	25
6.5.3.3	<a href="#">setMakespan()</a>	26
6.5.4	<a href="#">Member Data Documentation</a>	26
6.5.4.1	<a href="#">machine_num</a>	26
6.5.4.2	<a href="#">makespan</a>	27
6.5.4.3	<a href="#">private_arrow</a>	27
6.6	<a href="#">GanttChartOperation Class Reference</a>	27
6.6.1	<a href="#">Detailed Description</a>	28
6.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	28
6.6.2.1	<a href="#">GanttChartOperation()</a>	28
6.6.3	<a href="#">Member Function Documentation</a>	28
6.6.3.1	<a href="#">paint()</a>	29
6.6.4	<a href="#">Member Data Documentation</a>	29
6.6.4.1	<a href="#">m_color</a>	29
6.6.4.2	<a href="#">m_id</a>	29
6.7	<a href="#">Input Class Reference</a>	30

6.7.1	Detailed Description	31
6.7.2	Constructor & Destructor Documentation	31
6.7.2.1	Input()	31
6.7.2.2	~Input()	31
6.7.3	Member Function Documentation	32
6.7.3.1	on_start_button_clicked	32
6.7.4	Member Data Documentation	32
6.7.4.1	jssp	32
6.7.4.2	ui	33
6.8	JOB Struct Reference	33
6.8.1	Detailed Description	33
6.8.2	Member Data Documentation	33
6.8.2.1	estime	33
6.8.2.2	magic	34
6.8.2.3	mhtime	34
6.8.2.4	next	34
6.8.2.5	order	34
6.8.2.6	prev	34
6.8.2.7	process_time	35
6.8.2.8	start	35
6.8.2.9	step	35
6.9	JOBMACHINEPAR Struct Reference	35
6.9.1	Detailed Description	35
6.9.2	Member Data Documentation	35
6.9.2.1	job	36
6.9.2.2	machine	36
6.10	JobShop Class Reference	36
6.10.1	Detailed Description	37
6.10.2	Constructor & Destructor Documentation	37
6.10.2.1	JobShop()	37

6.10.3	Member Function Documentation	37
6.10.3.1	generateGantt()	38
6.10.3.2	getJob()	38
6.10.3.3	getJobSize()	38
6.10.3.4	getMachineSize()	39
6.10.3.5	getProb()	39
6.10.3.6	runProb()	39
6.10.3.7	setJobSize()	40
6.10.3.8	setMachineSize()	41
6.10.4	Member Data Documentation	41
6.10.4.1	fixer	41
6.10.4.2	m_job	42
6.10.4.3	m_job_size	42
6.10.4.4	m_machine_size	42
6.11	MACHINEORDER Struct Reference	42
6.11.1	Detailed Description	42
6.11.2	Member Data Documentation	42
6.11.2.1	machines	43
6.11.2.2	size	43
6.12	NEIGHBOR Struct Reference	43
6.12.1	Detailed Description	43
6.12.2	Member Data Documentation	43
6.12.2.1	next	43
6.12.2.2	prev	44
6.13	ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT Struct Reference	44
6.13.1	Detailed Description	44
6.13.2	Member Data Documentation	44
6.13.2.1	active	44
6.13.2.2	bound	44
6.14	ONEMACHINestime Struct Reference	45

6.14.1 Detailed Description . . . . .	45
6.14.2 Member Data Documentation . . . . .	45
6.14.2.1 estime . . . . .	45
6.14.2.2 mhtime . . . . .	45
6.14.2.3 process_time . . . . .	45
6.15 pair Struct Reference . . . . .	46
6.15.1 Detailed Description . . . . .	46
6.15.2 Member Data Documentation . . . . .	46
6.15.2.1 endtime . . . . .	46
6.15.2.2 starttime . . . . .	46
6.16 Result Class Reference . . . . .	47
6.16.1 Detailed Description . . . . .	48
6.16.2 Constructor & Destructor Documentation . . . . .	48
6.16.2.1 Result() . . . . .	48
6.16.2.2 ~Result() . . . . .	49
6.16.3 Member Function Documentation . . . . .	49
6.16.3.1 Fix() . . . . .	49
6.16.3.2 on_fixButton_clicked . . . . .	50
6.16.4 Member Data Documentation . . . . .	50
6.16.4.1 jssp . . . . .	50
6.16.4.2 private_chart . . . . .	51
6.16.4.3 private_scene . . . . .	51
6.16.4.4 ui . . . . .	51
6.17 SEQUENCE Struct Reference . . . . .	51
6.17.1 Detailed Description . . . . .	51
6.17.2 Member Data Documentation . . . . .	51
6.17.2.1 job . . . . .	51



<b>7 File Documentation</b>	<b>53</b>
7.1 algorithm/bottle.cpp File Reference	53
7.1.1 Macro Definition Documentation	54
7.1.1.1 TRY_COUNT	54
7.1.2 Typedef Documentation	54
7.1.2.1 blist_t	54
7.1.2.2 mo_t	54
7.1.2.3 neighbor_t	55
7.1.3 Function Documentation	55
7.1.3.1 clear_and_backup_seq()	55
7.1.3.2 clear_seq()	55
7.1.3.3 cp_mo()	56
7.1.3.4 cp_neighbor()	56
7.1.3.5 cp_seq()	57
7.1.3.6 critical()	57
7.1.3.7 eval()	58
7.1.3.8 re_optimization_phase_1()	59
7.1.3.9 re_optimization_phase_2()	60
7.1.3.10 restore_neighbor()	61
7.1.3.11 save_times()	62
7.1.3.12 set_seq()	62
7.1.3.13 shifting_bottle_neck()	62
7.1.4 Variable Documentation	64
7.1.4.1 best_makespan	64
7.2 algorithm/common_definition.cpp File Reference	64
7.2.1 Variable Documentation	64
7.2.1.1 job	64
7.2.1.2 job_size	65
7.2.1.3 machine_size	65
7.2.1.4 terminate_flag	65

7.3	algorithm/eval.cpp File Reference	65
7.3.1	Typedef Documentation	66
7.3.1.1	job_machine_t	66
7.3.2	Function Documentation	66
7.3.2.1	eval()	66
7.3.3	Variable Documentation	67
7.3.3.1	magicnum	67
7.4	algorithm/onemachine.cpp File Reference	67
7.4.1	Macro Definition Documentation	68
7.4.1.1	ONEMACH_BBNODES	68
7.4.2	Typedef Documentation	68
7.4.2.1	onemach_bb_ass_t	68
7.4.3	Function Documentation	68
7.4.3.1	cp_onemach_time()	69
7.4.3.2	lower_bound()	69
7.4.3.3	make_span()	70
7.4.3.4	make_value()	70
7.4.3.5	mwr_schedule()	71
7.4.3.6	one_machine()	71
7.5	gantt/arrow.cpp File Reference	72
7.5.1	Detailed Description	73
7.5.2	Macro Definition Documentation	73
7.5.2.1	USE_MATH_DEFINES	73
7.5.3	Variable Documentation	74
7.5.3.1	arrowSize	74
7.5.3.2	Pi	74
7.6	gantt/ganttchartbase.cpp File Reference	74
7.6.1	Detailed Description	74
7.7	gantt/ganttchartmachine.cpp File Reference	75
7.7.1	Detailed Description	75

7.8	<a href="#">gant/ganttchartoperation.cpp File Reference</a>	75
7.9	<a href="#">include/arrow.h File Reference</a>	76
7.9.1	<a href="#">Detailed Description</a>	76
7.10	<a href="#">include/bottle.h File Reference</a>	77
7.10.1	<a href="#">Detailed Description</a>	78
7.10.2	<a href="#">Macro Definition Documentation</a>	78
7.10.2.1	<a href="#">ININITAS</a>	78
7.10.2.2	<a href="#">MAX</a>	78
7.10.2.3	<a href="#">MAXJOB</a>	78
7.10.2.4	<a href="#">MAXMACHINE</a>	79
7.10.3	<a href="#">Typedef Documentation</a>	79
7.10.3.1	<a href="#">job_t</a>	79
7.10.3.2	<a href="#">onemach_times_t</a>	79
7.10.3.3	<a href="#">sequence_t</a>	79
7.10.4	<a href="#">Function Documentation</a>	79
7.10.4.1	<a href="#">one_machine()</a>	79
7.10.4.2	<a href="#">prestissimo()</a>	80
7.10.4.3	<a href="#">run_bottle_neck()</a>	81
7.10.5	<a href="#">Variable Documentation</a>	81
7.10.5.1	<a href="#">job</a>	81
7.10.5.2	<a href="#">job_size</a>	82
7.10.5.3	<a href="#">machine_size</a>	82
7.10.5.4	<a href="#">terminate_flag</a>	82
7.11	<a href="#">include/ganttchartbase.h File Reference</a>	82
7.11.1	<a href="#">Detailed Description</a>	83
7.12	<a href="#">include/ganttchartmachine.h File Reference</a>	83
7.12.1	<a href="#">Detailed Description</a>	84
7.13	<a href="#">include/ganttchartoperation.h File Reference</a>	84
7.13.1	<a href="#">Detailed Description</a>	85
7.14	<a href="#">include/input.h File Reference</a>	85

7.14.1 Detailed Description . . . . .	86
7.15 include/jobshop.h File Reference . . . . .	86
7.15.1 Detailed Description . . . . .	87
7.16 include/result.h File Reference . . . . .	87
7.16.1 Detailed Description . . . . .	88
7.17 jobshop.cpp File Reference . . . . .	88
7.17.1 Detailed Description . . . . .	89
7.17.2 Variable Documentation . . . . .	89
7.17.2.1 best_makespan . . . . .	89
7.18 main.cpp File Reference . . . . .	89
7.18.1 Detailed Description . . . . .	90
7.18.2 Function Documentation . . . . .	90
7.18.2.1 main() . . . . .	90
7.19 ui/input.cpp File Reference . . . . .	90
7.19.1 Detailed Description . . . . .	91
7.20 ui/result.cpp File Reference . . . . .	91
7.20.1 Detailed Description . . . . .	92
7.20.2 Function Documentation . . . . .	92
7.20.2.1 starttime_cmp() . . . . .	92
7.20.3 Variable Documentation . . . . .	92
7.20.3.1 best_makespan . . . . .	92
<b>Index</b>	<b>93</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui . . . . .	9
--------------	---



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

BLIST . . . . .	17
Fixer . . . . .	18
JOB . . . . .	33
JOBMACHINEPAR . . . . .	35
MACHINEORDER . . . . .	42
NEIGHBOR . . . . .	43
ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT . . . . .	44
ONEMACHINestime . . . . .	45
pair . . . . .	46
QDialog	
Result . . . . .	47
QGraphicsItem	
GanttChartBase . . . . .	19
GanttChartMachine . . . . .	23
QGraphicsLineItem	
Arrow . . . . .	11
QGraphicsRectItem	
GanttChartOperation . . . . .	27
QMainWindow	
Input . . . . .	30
QObject	
JobShop . . . . .	36
SEQUENCE . . . . .	51





## Chapter 3

# Class Index

### 3.1 Class List

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

Arrow	11
BLIST	17
Fixer	18
GanttChartBase	19
GanttChartMachine	23
GanttChartOperation	27
Input	30
JOB	33
JOBMACHINEPAR	35
JobShop	36
MACHINEORDER	42
NEIGHBOR	43
ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT	44
ONEMACHINestime	45
pair	
The pair struct for sorting machines	46
Result	47
SEQUENCE	51



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">jobshop.cpp</a>	88
<a href="#">main.cpp</a>	89
<a href="#">algorithm/bottle.cpp</a>	53
<a href="#">algorithm/common_definition.cpp</a>	64
<a href="#">algorithm/eval.cpp</a>	65
<a href="#">algorithm/onemachine.cpp</a>	67
<a href="#">gantt/arrow.cpp</a>	72
<a href="#">gantt/ganttchartbase.cpp</a>	74
<a href="#">gantt/ganttchartmachine.cpp</a>	75
<a href="#">gantt/ganttchartoperation.cpp</a>	75
<a href="#">include/arrow.h</a>	76
<a href="#">include/bottle.h</a>	
Header file for the whole project	77
<a href="#">include/ganttchartbase.h</a>	82
<a href="#">include/ganttchartmachine.h</a>	83
<a href="#">include/ganttchartoperation.h</a>	84
<a href="#">include/input.h</a>	85
<a href="#">include/jobshop.h</a>	86
<a href="#">include/result.h</a>	87
<a href="#">ui/input.cpp</a>	90
<a href="#">ui/result.cpp</a>	91



## **Chapter 5**

# **Namespace Documentation**

### **5.1 Ui Namespace Reference**



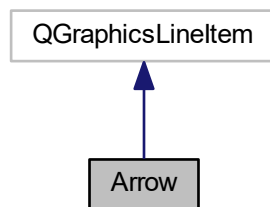
## Chapter 6

# Class Documentation

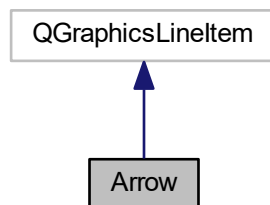
### 6.1 Arrow Class Reference

```
#include <arrow.h>
```

Inheritance diagram for Arrow:



Collaboration diagram for Arrow:



## Public Types

- enum { [Type](#) = UserType + 1 }

## Public Member Functions

- virtual int [type](#) () const
- [Arrow](#) (QGraphicsItem \*parent=0)  
*Creat an arrow.*
- [Arrow](#) (const QPointF &[startPoint](#), const QPointF &[endPoint](#), QGraphicsItem \*parent=0)  
*Creat an arrow : The harder version.*
- virtual QRectF [boundingRect](#) () const
- virtual QPainterPath [shape](#) () const
- QPointF [startPoint](#) () const
- void [setStartPoint](#) (const QPointF &[startPoint](#))
- QPointF [endPoint](#) () const
- void [setEndPoint](#) (const QPointF &[endPoint](#))

## Protected Member Functions

- virtual void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget=0)

## Private Attributes

- QPointF [private\\_startPoint](#)
- QPointF [private\\_endPoint](#)
- QPolygonF [private\\_arrowHead](#)

### 6.1.1 Detailed Description

Definition at line 22 of file arrow.h.

### 6.1.2 Member Enumeration Documentation

#### 6.1.2.1 anonymous enum

anonymous enum

#### Enumerator

Type	
------	--

Definition at line 25 of file arrow.h.



### 6.1.3 Constructor & Destructor Documentation

#### 6.1.3.1 Arrow() [1/2]

```
Arrow::Arrow (  
    QGraphicsItem * parent = 0 ) [explicit]
```

Creates an arrow.

##### Parameters

<i>parent</i>	
---------------	--

Definition at line 21 of file arrow.cpp.

#### 6.1.3.2 Arrow() [2/2]

```
Arrow::Arrow (  
    const QPointF & startPoint,  
    const QPointF & endPoint,  
    QGraphicsItem * parent = 0 )
```

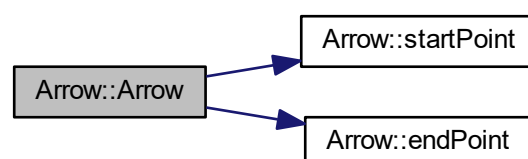
Creates an arrow : The harder version.

##### Parameters

<i>startPoint</i>	
<i>endPoint</i>	
<i>parent</i>	

Definition at line 33 of file arrow.cpp.

Here is the call graph for this function:



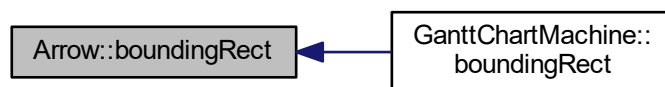
## 6.1.4 Member Function Documentation

### 6.1.4.1 boundingRect()

```
QRectF Arrow::boundingRect ( ) const [virtual]
```

Definition at line 42 of file arrow.cpp.

Here is the caller graph for this function:

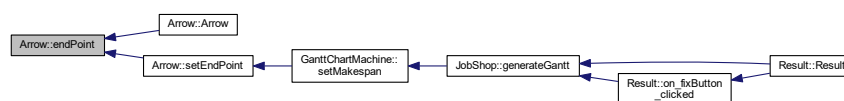


### 6.1.4.2 endPoint()

```
QPointF Arrow::endPoint ( ) const
```

Definition at line 87 of file arrow.cpp.

Here is the caller graph for this function:



### 6.1.4.3 paint()

```
void Arrow::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget = 0 ) [protected], [virtual]
```

Definition at line 57 of file arrow.cpp.

#### 6.1.4.4 setEndPoint()

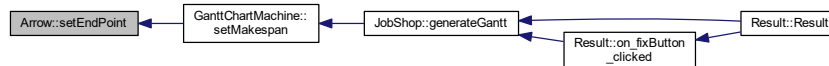
```
void Arrow::setEndPoint (
    const QPointF & endPoint )
```

Definition at line 91 of file arrow.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.1.4.5 setStartPoint()

```
void Arrow::setStartPoint (
    const QPointF & startPoint )
```

Definition at line 99 of file arrow.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.1.4.6 shape()

```
QPainterPath Arrow::shape ( ) const [virtual]
```

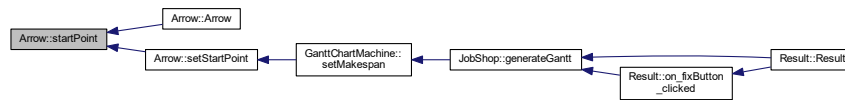
Definition at line 51 of file arrow.cpp.

#### 6.1.4.7 startPoint()

```
QPointF Arrow::startPoint ( ) const
```

Definition at line 95 of file arrow.cpp.

Here is the caller graph for this function:



#### 6.1.4.8 type()

```
virtual int Arrow::type ( ) const [inline], [virtual]
```

Definition at line 26 of file arrow.h.

### 6.1.5 Member Data Documentation

#### 6.1.5.1 private\_arrowHead

```
QPolygonF Arrow::private_arrowHead [private]
```

Definition at line 48 of file arrow.h.

#### 6.1.5.2 private\_endPoint

```
QPointF Arrow::private_endPoint [private]
```

Definition at line 47 of file arrow.h.

### 6.1.5.3 private\_startPoint

```
QPointF Arrow::private_startPoint [private]
```

Definition at line 46 of file arrow.h.

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

- [include/arrow.h](#)
- [gantt/arrow.cpp](#)

## 6.2 BLIST Struct Reference

### Public Attributes

- int [machine](#)
- int [makespan](#)
- int [order](#) [MAXJOB]

### 6.2.1 Detailed Description

Store the bottle information.

Definition at line 19 of file bottle.cpp.

### 6.2.2 Member Data Documentation

#### 6.2.2.1 machine

```
int BLIST::machine
```

Machine number of this bottle.

Definition at line 20 of file bottle.cpp.

#### 6.2.2.2 makespan

```
int BLIST::makespan
```

Makespan of this bottle.

Definition at line 21 of file bottle.cpp.

### 6.2.2.3 order

```
int BLIST::order[MAXJOB]
```

Job order of this bottle.

Definition at line 22 of file bottle.cpp.

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

- [algorithm/bottle.cpp](#)

## 6.3 Fixer Struct Reference

```
#include <jobshop.h>
```

### Public Attributes

- int [machine](#)
- int [starttime](#)
- int [duration](#)

### 6.3.1 Detailed Description

Definition at line 19 of file jobshop.h.

### 6.3.2 Member Data Documentation

#### 6.3.2.1 duration

```
int Fixer::duration
```

Definition at line 22 of file jobshop.h.

#### 6.3.2.2 machine

```
int Fixer::machine
```

Definition at line 20 of file jobshop.h.

### 6.3.2.3 starttime

```
int Fixer::starttime
```

Definition at line 21 of file jobshop.h.

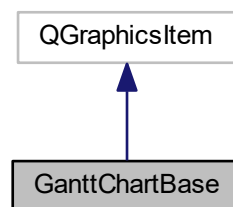
The documentation for this struct was generated from the following file:

- [include/jobshop.h](#)

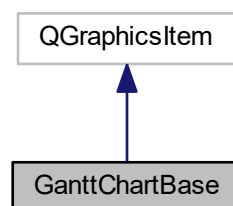
## 6.4 GanttChartBase Class Reference

```
#include <ganttchartbase.h>
```

Inheritance diagram for GanttChartBase:



Collaboration diagram for GanttChartBase:



### Public Member Functions

- [GanttChartBase](#) (int [makespan](#))  
*Construct a gantt chart.*
- virtual QRectF [boundingRect](#) () const

### Static Public Member Functions

- static QPointF [operationPosition](#) (int time)
- static QPointF [machineOffset](#) ()

### Static Public Attributes

- static const int [widthUnit](#) = 10
- static const int [operationHeight](#) = 20
- static const int [machineHeight](#) = [operationHeight](#) \* 3
- static const int [machineHorizontalOffset](#) = 35

### Protected Member Functions

- virtual void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget=0)  
*Paint this gantt chart.*

### Protected Attributes

- int [makespan](#)

## 6.4.1 Detailed Description

Definition at line 11 of file ganttchartbase.h.

## 6.4.2 Constructor & Destructor Documentation

### 6.4.2.1 GanttChartBase()

```
GanttChartBase::GanttChartBase (  
    int makespan )
```

Construct a gantt chart.

#### Parameters

<i>makespan</i>	The makespan of this project.
-----------------	-------------------------------

Definition at line 25 of file ganttchartbase.cpp.

## 6.4.3 Member Function Documentation



## 6.4.3.1 boundingRect()

```
QRectF GanttChartBase::boundingRect ( ) const [virtual]
```

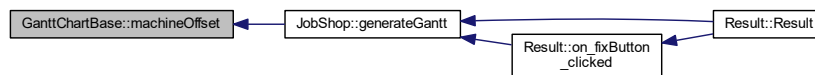
Definition at line 44 of file ganttchartbase.cpp.

## 6.4.3.2 machineOffset()

```
static QPointF GanttChartBase::machineOffset ( ) [inline], [static]
```

Definition at line 25 of file ganttchartbase.h.

Here is the caller graph for this function:

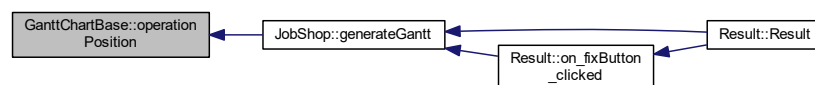


## 6.4.3.3 operationPosition()

```
static QPointF GanttChartBase::operationPosition (
    int time ) [inline], [static]
```

Definition at line 22 of file ganttchartbase.h.

Here is the caller graph for this function:



## 6.4.3.4 paint()

```
void GanttChartBase::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget = 0 ) [protected], [virtual]
```

Paint this gantt chart.

**Parameters**

<i>painter</i>	
<i>option</i>	
<i>widget</i>	

Definition at line 55 of file ganttchartbase.cpp.

## 6.4.4 Member Data Documentation

### 6.4.4.1 machineHeight

```
const int GanttChartBase::machineHeight = operationHeight * 3 [static]
```

Definition at line 19 of file ganttchartbase.h.

### 6.4.4.2 machineHorizontalOffset

```
const int GanttChartBase::machineHorizontalOffset = 35 [static]
```

Definition at line 20 of file ganttchartbase.h.

### 6.4.4.3 makespan

```
int GanttChartBase::makespan [protected]
```

Definition at line 33 of file ganttchartbase.h.

### 6.4.4.4 operationHeight

```
const int GanttChartBase::operationHeight = 20 [static]
```

Definition at line 18 of file ganttchartbase.h.

#### 6.4.4.5 widthUnit

```
const int GanttChartBase::widthUnit = 10 [static]
```

Definition at line 17 of file ganttchartbase.h.

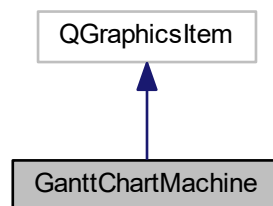
The documentation for this class was generated from the following files:

- include/[ganttchartbase.h](#)
- gantt/[ganttchartbase.cpp](#)

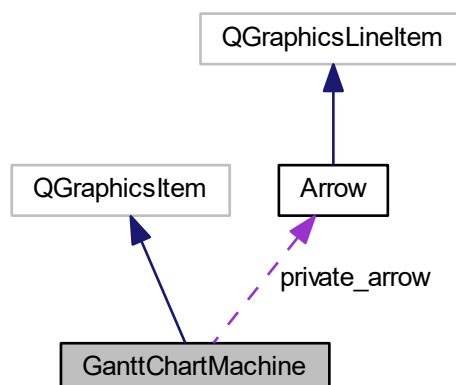
## 6.5 GanttChartMachine Class Reference

```
#include <ganttchartmachine.h>
```

Inheritance diagram for GanttChartMachine:



Collaboration diagram for GanttChartMachine:



## Public Member Functions

- [GanttChartMachine](#) (const QString &id, QGraphicsItem \*parent=0)  
*Construct a machine class.*
- [~GanttChartMachine](#) ()
- virtual QRectF [boundingRect](#) () const
- void [setMakespan](#) (int cMax)  
*Set makespan of this machine.*

## Protected Member Functions

- virtual void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget=0)  
*Paint this machine.*

## Private Attributes

- int [makespan](#)
- QString [machine\\_num](#)
- [Arrow](#) \* [private\\_arrow](#)

### 6.5.1 Detailed Description

Definition at line 15 of file ganttchartmachine.h.

### 6.5.2 Constructor & Destructor Documentation

#### 6.5.2.1 GanttChartMachine()

```
GanttChartMachine::GanttChartMachine (
    const QString & id,
    QGraphicsItem * parent = 0 )
```

Construct a machine class.

#### Parameters

<i>id</i>	The given machine id.
<i>parent</i>	

Definition at line 24 of file ganttchartmachine.cpp.

### 6.5.2.2 ~GanttChartMachine()

GanttChartMachine::~~GanttChartMachine ( )

Definition at line 42 of file ganttchartmachine.cpp.

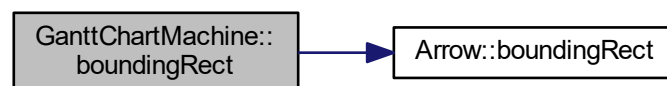
## 6.5.3 Member Function Documentation

### 6.5.3.1 boundingRect()

QRectF GanttChartMachine::boundingRect ( ) const [virtual]

Definition at line 46 of file ganttchartmachine.cpp.

Here is the call graph for this function:



### 6.5.3.2 paint()

```
void GanttChartMachine::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget = 0 ) [protected], [virtual]
```

Paint this machine.

#### Parameters

<i>painter</i>	
<i>option</i>	
<i>widget</i>	

Definition at line 58 of file ganttchartmachine.cpp.

### 6.5.3.3 setMakespan()

```
void GanttChartMachine::setMakespan (
    int cMax )
```

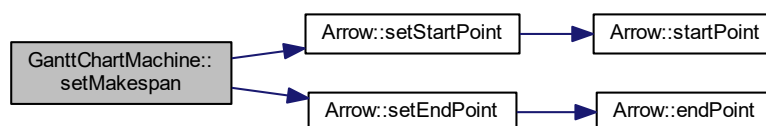
Set makespan of this machine.

#### Parameters

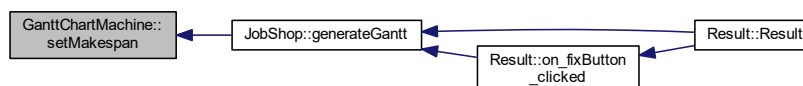
<i>cMax</i>	The makespan value.
-------------	---------------------

Definition at line 69 of file ganttchartmachine.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



## 6.5.4 Member Data Documentation

### 6.5.4.1 machine\_num

```
QString GanttChartMachine::machine_num [private]
```

Definition at line 32 of file ganttchartmachine.h.

#### 6.5.4.2 makespan

```
int GanttChartMachine::makespan [private]
```

Definition at line 31 of file ganttchartmachine.h.

#### 6.5.4.3 private\_arrow

```
Arrow* GanttChartMachine::private_arrow [private]
```

Definition at line 33 of file ganttchartmachine.h.

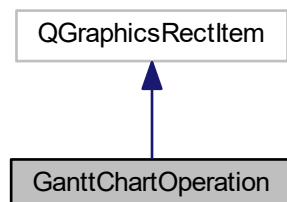
The documentation for this class was generated from the following files:

- include/[ganttchartmachine.h](#)
- gantt/[ganttchartmachine.cpp](#)

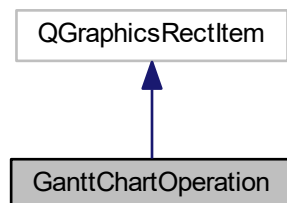
## 6.6 GanttChartOperation Class Reference

```
#include <ganttchartoperation.h>
```

Inheritance diagram for GanttChartOperation:



Collaboration diagram for GanttChartOperation:



## Public Member Functions

- [GanttChartOperation](#) (const QString &id, int time, QColor color)  
*Construct a [GanttChartOperation](#) class.*

## Protected Member Functions

- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*Paint this operation.*

## Protected Attributes

- QString [m\\_id](#)
- QColor [m\\_color](#)

### 6.6.1 Detailed Description

Definition at line 14 of file ganttchartoperation.h.

### 6.6.2 Constructor & Destructor Documentation

#### 6.6.2.1 GanttChartOperation()

```
GanttChartOperation::GanttChartOperation (
    const QString & id,
    int time,
    QColor color )
```

Construct a [GanttChartOperation](#) class.

#### Parameters

<i>id</i>	Job id.
<i>time</i>	Start time.
<i>color</i>	Color of this operation.

Definition at line 20 of file ganttchartoperation.cpp.

### 6.6.3 Member Function Documentation



### 6.6.3.1 paint()

```
void GanttChartOperation::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget ) [protected]
```

Paint this operation.

#### Parameters

<i>painter</i>	
<i>option</i>	
<i>widget</i>	

Definition at line 34 of file ganttchartoperation.cpp.

## 6.6.4 Member Data Documentation

### 6.6.4.1 m\_color

```
QColor GanttChartOperation::m_color [protected]
```

Definition at line 25 of file ganttchartoperation.h.

### 6.6.4.2 m\_id

```
QString GanttChartOperation::m_id [protected]
```

Definition at line 24 of file ganttchartoperation.h.

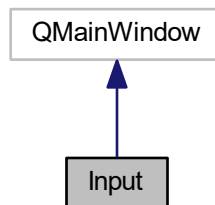
The documentation for this class was generated from the following files:

- [include/ganttchartoperation.h](#)
- [gantt/ganttchartoperation.cpp](#)

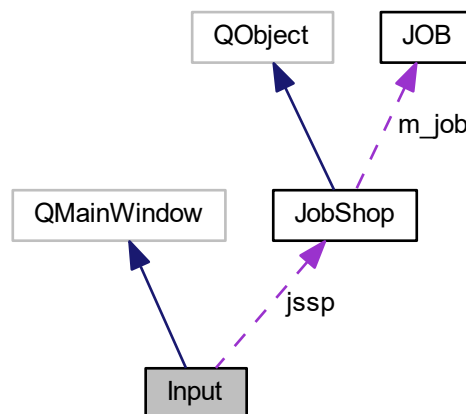
## 6.7 Input Class Reference

```
#include <input.h>
```

Inheritance diagram for Input:



Collaboration diagram for Input:



### Public Member Functions

- [Input](#) (QWidget \*parent=0)  
*Set the input window.*
- [~Input](#) ()  
*Delete the window.*

### Private Slots

- void [on\\_start\\_button\\_clicked](#) ()  
*Action of start button.*

## Private Attributes

- `Ui::Input * ui`
- `JobShop * jssp`

### 6.7.1 Detailed Description

Definition at line 18 of file input.h.

### 6.7.2 Constructor & Destructor Documentation

#### 6.7.2.1 Input()

```
Input::Input (
    QWidget * parent = 0 ) [explicit]
```

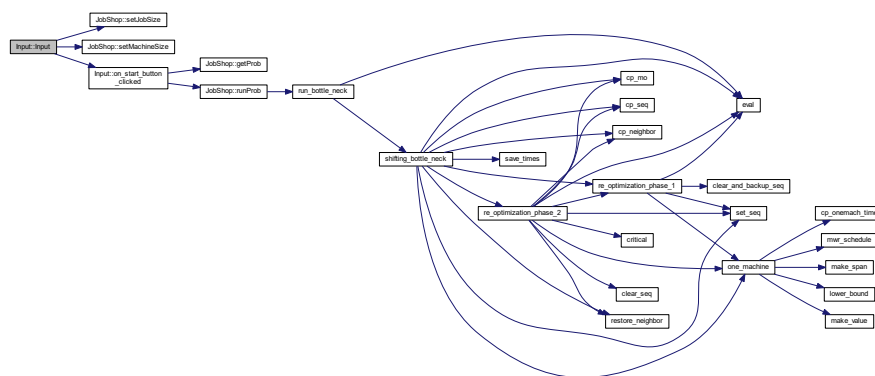
Set the input window.

#### Parameters

<i>parent</i>	
---------------	--

Definition at line 17 of file input.cpp.

Here is the call graph for this function:



#### 6.7.2.2 ~Input()

```
Input::~Input ( )
```

Delete the window.

Definition at line 31 of file input.cpp.

### 6.7.3 Member Function Documentation

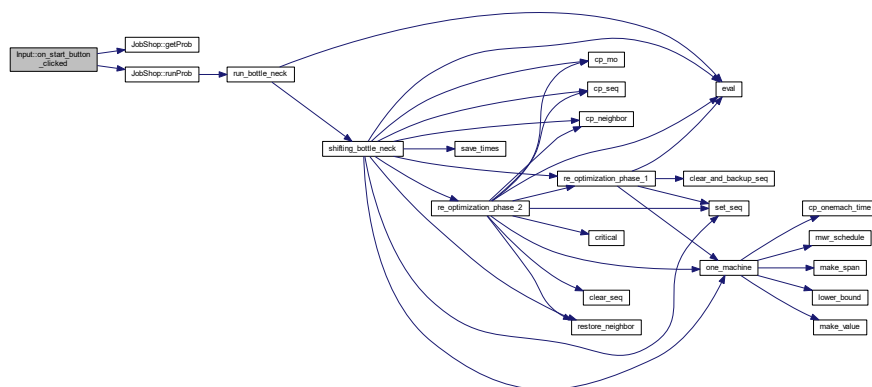
#### 6.7.3.1 on\_start\_button\_clicked

```
void Input::on_start_button_clicked ( ) [private], [slot]
```

Action of start button.

Definition at line 40 of file input.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.7.4 Member Data Documentation

#### 6.7.4.1 jssp

```
JobShop* Input::jssp [private]
```

Definition at line 31 of file input.h.

#### 6.7.4.2 ui

```
Ui::Input* Input::ui [private]
```

Definition at line 30 of file input.h.

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

- [include/input.h](#)
- [ui/input.cpp](#)

## 6.8 JOB Struct Reference

```
#include <bottle.h>
```

### Public Attributes

- int [etime](#) [MAXMACHINE]
- int [mhtime](#) [MAXMACHINE]
- int [magic](#) [MAXMACHINE]
- int [order](#) [MAXMACHINE]
- int [process\\_time](#) [MAXMACHINE]
- int [step](#) [MAXMACHINE]
- int [next](#) [MAXMACHINE]
- int [prev](#) [MAXMACHINE]
- int [start](#) [MAXMACHINE]

### 6.8.1 Detailed Description

Data representation for a job.

Definition at line 43 of file bottle.h.

### 6.8.2 Member Data Documentation

#### 6.8.2.1 etime

```
int JOB::etime [MAXMACHINE]
```

Earlist starting time of this job on each machine. Which is simply the sum of this job's processing times on the machine before [order[machine]] in this jobs prescribed ordering.

Definition at line 44 of file bottle.h.

#### 6.8.2.2 magic

```
int JOB::magic[MAXMACHINE]
```

The number generated and managed by the God in the computer. Every one who changed the name of this feild will be seen as an evil and will be cursed by the God.

Definition at line 46 of file bottle.h.

#### 6.8.2.3 mhtime

```
int JOB::mhtime[MAXMACHINE]
```

Minimum halting time of this job after [machine num]. Which is simply the sum of this job's processing times on the machine after [order[machine]] in this jobs prescribed ordering.

Definition at line 45 of file bottle.h.

#### 6.8.2.4 next

```
int JOB::next[MAXMACHINE]
```

Next job on machine [i].

Definition at line 50 of file bottle.h.

#### 6.8.2.5 order

```
int JOB::order[MAXMACHINE]
```

Required machine order for the job.

Definition at line 47 of file bottle.h.

#### 6.8.2.6 prev

```
int JOB::prev[MAXMACHINE]
```

Previous job blah blah.

Definition at line 51 of file bottle.h.

#### 6.8.2.7 process\_time

```
int JOB::process_time[MAXMACHINE]
```

Process time of each machine.

Definition at line 48 of file bottle.h.

#### 6.8.2.8 start

```
int JOB::start[MAXMACHINE]
```

Start time of this job on each machine.

Definition at line 52 of file bottle.h.

#### 6.8.2.9 step

```
int JOB::step[MAXMACHINE]
```

Solution step indexed by machine.

Definition at line 49 of file bottle.h.

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

- include/[bottle.h](#)

## 6.9 JOBMACHINEPAR Struct Reference

### Public Attributes

- int [job](#)
- int [machine](#)

#### 6.9.1 Detailed Description

Auxiliary struct to calculate the makespan.

Definition at line 13 of file eval.cpp.

#### 6.9.2 Member Data Documentation

### 6.9.2.1 job

```
int JOBMACHINEPAR::job
```

Definition at line 14 of file eval.cpp.

### 6.9.2.2 machine

```
int JOBMACHINEPAR::machine
```

Definition at line 15 of file eval.cpp.

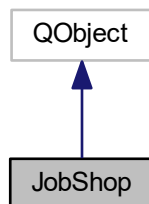
The documentation for this struct was generated from the following file:

- [algorithm/eval.cpp](#)

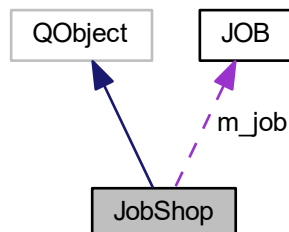
## 6.10 JobShop Class Reference

```
#include <jobshop.h>
```

Inheritance diagram for JobShop:



Collaboration diagram for JobShop:





## Public Member Functions

- [JobShop](#) ()  
*Get pointers of previous defined variables.*
- [job\\_t \\* getJob](#) ()
- [int getJobSize](#) ()
- [int getMachineSize](#) ()
- [void setJobSize](#) (int i)  
*Set job size using the given value.*
- [void setMachineSize](#) (int i)  
*Set machine size using the given value.*
- [void getProb](#) (QString)  
*Read innstance file from the given str.*
- [GanttChartBase \\* generateGantt](#) ()  
*Generate the gantt chart from our result.*
- [void runProb](#) ()  
*Run the awesome soler algorithm.*

## Public Attributes

- [QVector< Fixer \\* > fixer](#)

## Private Attributes

- [job\\_t \\* m\\_job](#)
- [int \\* m\\_job\\_size](#)
- [int \\* m\\_machine\\_size](#)

### 6.10.1 Detailed Description

Definition at line 26 of file jobshop.h.

### 6.10.2 Constructor & Destructor Documentation

#### 6.10.2.1 JobShop()

```
JobShop::JobShop ( )
```

Get pointers of previous defined variables.

Definition at line 21 of file jobshop.cpp.

### 6.10.3 Member Function Documentation

### 6.10.3.1 generateGantt()

```
GanttChartBase * JobShop::generateGantt ( )
```

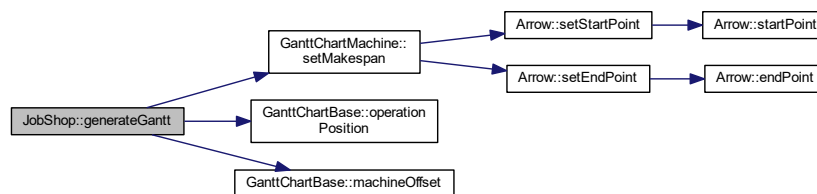
Generate the gantt chart from our result.

#### Returns

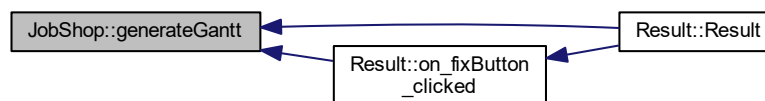
The gantt chart.

Definition at line 74 of file jobshop.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.10.3.2 getJob()

```
job_t* JobShop::getJob ( ) [inline]
```

Definition at line 31 of file jobshop.h.

### 6.10.3.3 getJobSize()

```
int JobShop::getJobSize ( ) [inline]
```

Definition at line 32 of file jobshop.h.

#### 6.10.3.4 getMachineSize()

```
int JobShop::getMachineSize ( ) [inline]
```

Definition at line 33 of file jobshop.h.

#### 6.10.3.5 getProb()

```
void JobShop::getProb (
    QString str )
```

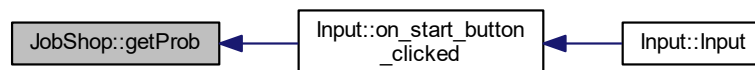
Read innstance file from the given str.

##### Parameters

<i>str</i>	The given string grom the QPlainTextEdit
------------	--

Definition at line 49 of file jobshop.cpp.

Here is the caller graph for this function:



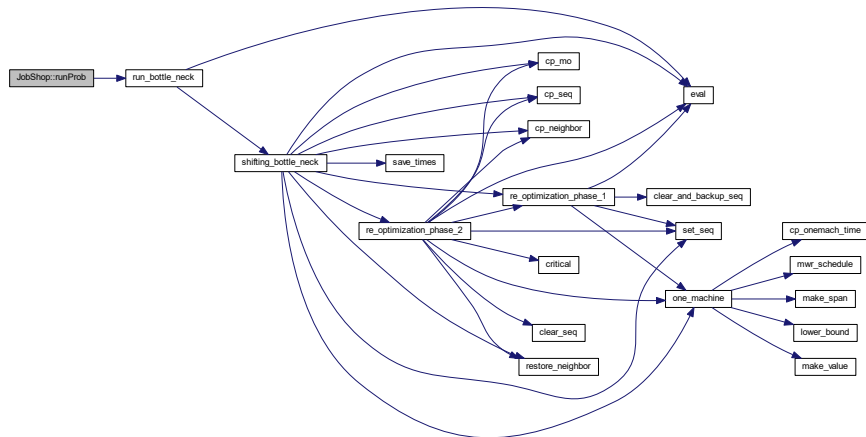
#### 6.10.3.6 runProb()

```
void JobShop::runProb ( )
```

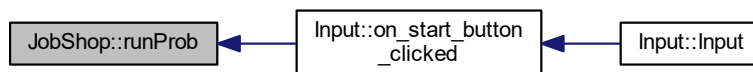
Run the awesome soler algorithm.

Definition at line 66 of file jobshop.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.10.3.7 setJobSize()

```
void JobShop::setJobSize (
    int i )
```

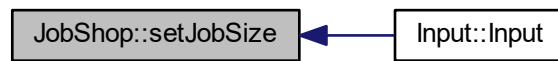
Set job size using the given value.

#### Parameters

<i>i</i>	Given value from the spin box
----------	-------------------------------

Definition at line 33 of file jobshop.cpp.

Here is the caller graph for this function:



#### 6.10.3.8 setMachineSize()

```
void JobShop::setMachineSize (  
    int i )
```

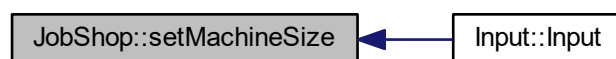
Set machine size using the given value.

##### Parameters

<i>i</i>	Given value from the spin box
----------	-------------------------------

Definition at line 41 of file jobshop.cpp.

Here is the caller graph for this function:



## 6.10.4 Member Data Documentation

### 6.10.4.1 fixer

```
QVector<Fixer*> JobShop::fixer
```

Definition at line 39 of file jobshop.h.

#### 6.10.4.2 m\_job

```
job_t* JobShop::m_job [private]
```

Definition at line 41 of file jobshop.h.

#### 6.10.4.3 m\_job\_size

```
int* JobShop::m_job_size [private]
```

Definition at line 42 of file jobshop.h.

#### 6.10.4.4 m\_machine\_size

```
int* JobShop::m_machine_size [private]
```

Definition at line 43 of file jobshop.h.

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

- [include/jobshop.h](#)
- [jobshop.cpp](#)

## 6.11 MACHINEORDER Struct Reference

### Public Attributes

- int [size](#)
- int [machines](#) [[MAXMACHINE](#)]

#### 6.11.1 Detailed Description

Machine order type.

Definition at line 28 of file bottle.cpp.

#### 6.11.2 Member Data Documentation

### 6.11.2.1 machines

```
int MACHINEORDER::machines [MAXMACHINE]
```

Sequenced machine list.

Definition at line 30 of file bottle.cpp.

### 6.11.2.2 size

```
int MACHINEORDER::size
```

Sequenced machine number.

Definition at line 29 of file bottle.cpp.

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

- [algorithm/bottle.cpp](#)

## 6.12 NEIGHBOR Struct Reference

### Public Attributes

- int [next](#) [MAXMACHINE]
- int [prev](#) [MAXMACHINE]

### 6.12.1 Detailed Description

A temporary struct to store a sequence.

Definition at line 37 of file bottle.cpp.

### 6.12.2 Member Data Documentation

#### 6.12.2.1 next

```
int NEIGHBOR::next [MAXMACHINE]
```

You will be either silly or able to understand the name's mean.

Definition at line 38 of file bottle.cpp.

#### 6.12.2.2 prev

```
int NEIGHBOR::prev[MAXMACHINE]
```

Same as the previous one.

Definition at line 39 of file [bottle.cpp](#).

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

- [algorithm/bottle.cpp](#)

## 6.13 ONEMACHINE\_BRANCH\_AND\_BOUND\_ASSISTANT Struct Reference

### Public Attributes

- int [active](#)
- int [bound](#)

#### 6.13.1 Detailed Description

Info of node of a branch and bound tree.

Definition at line 21 of file [onemachine.cpp](#).

#### 6.13.2 Member Data Documentation

##### 6.13.2.1 active

```
int ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT::active
```

Wether this node is active

Definition at line 22 of file [onemachine.cpp](#).

##### 6.13.2.2 bound

```
int ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT::bound
```

See "JOBTYPE" for more info.

Definition at line 23 of file [onemachine.cpp](#).

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

- [algorithm/onemachine.cpp](#)



## 6.14 ONEMACHINestime Struct Reference

```
#include <bottle.h>
```

### Public Attributes

- int [estime](#) [[MAXJOB](#)]
- int [mhtime](#) [[MAXJOB](#)]
- int [process\\_time](#) [[MAXJOB](#)]

### 6.14.1 Detailed Description

Store the time info for every job runs on the same machine.

Definition at line 65 of file bottle.h.

### 6.14.2 Member Data Documentation

#### 6.14.2.1 estime

```
int ONEMACHINestime::estime [MAXJOB]
```

See "JOBTYPE" for more info.

Definition at line 66 of file bottle.h.

#### 6.14.2.2 mhtime

```
int ONEMACHINestime::mhtime [MAXJOB]
```

See "JOBTYPE" for more info.

Definition at line 67 of file bottle.h.

#### 6.14.2.3 process\_time

```
int ONEMACHINestime::process_time [MAXJOB]
```

See "JOBTYPE" for more info.

Definition at line 68 of file bottle.h.

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

- [include/bottle.h](#)

## 6.15 pair Struct Reference

The pair struct for sorting machines.

### Public Attributes

- int [starttime](#)
- int [endtime](#)

### 6.15.1 Detailed Description

The pair struct for sorting machines.

Definition at line 20 of file result.cpp.

### 6.15.2 Member Data Documentation

#### 6.15.2.1 endtime

```
int pair::endtime
```

Definition at line 23 of file result.cpp.

#### 6.15.2.2 starttime

```
int pair::starttime
```

Definition at line 22 of file result.cpp.

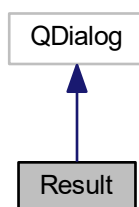
The documentation for this struct was generated from the following file:

- ui/[result.cpp](#)

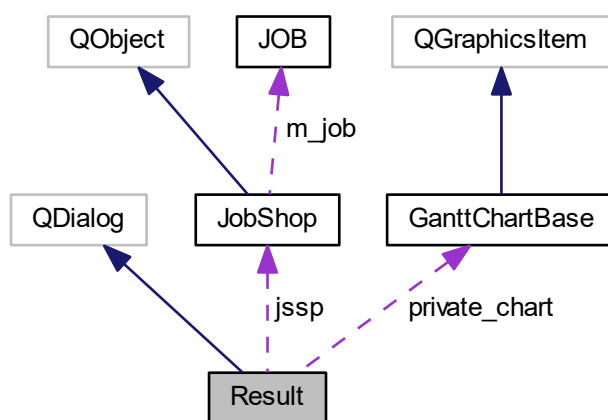
## 6.16 Result Class Reference

```
#include <result.h>
```

Inheritance diagram for Result:



Collaboration diagram for Result:



### Public Member Functions

- [Result](#) ([JobShop](#) \*instance, QWidget \*parent=0)  
*Construct the result window.*
- void [Fix](#) (int machine, int clock, int duration)  
*Handle the fix command.*
- [~Result](#) ()  
*Delete the window.*

## Private Slots

- void [on\\_fixButton\\_clicked](#) ()  
*The action of fixButton.*

## Private Attributes

- Ui::Result \* [ui](#)
- [JobShop](#) \* [jssp](#)
- [GanttChartBase](#) \* [private\\_chart](#)
- QGraphicsScene \* [private\\_scene](#)

### 6.16.1 Detailed Description

Definition at line 24 of file result.h.

### 6.16.2 Constructor & Destructor Documentation

#### 6.16.2.1 Result()

```
Result::Result (
    JobShop * instance,
    QWidget * parent = 0 )
```

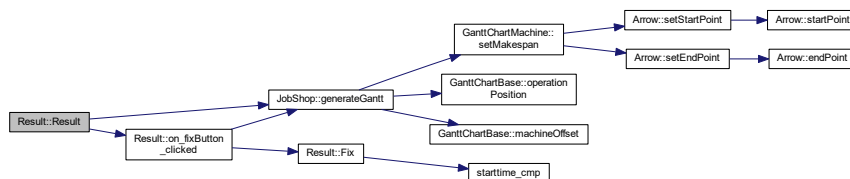
Construct the result window.

#### Parameters

<i>instance</i>	The instance class.
<i>parent</i>	...

Definition at line 42 of file result.cpp.

Here is the call graph for this function:



### 6.16.2.2 ~Result()

```
Result::~~Result ( )
```

Delete the window.

Definition at line 60 of file result.cpp.

## 6.16.3 Member Function Documentation

### 6.16.3.1 Fix()

```
void Result::Fix (
    int machine,
    int clock,
    int duration )
```

Handle the fix command.

#### Parameters

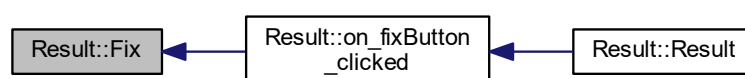
<i>machine</i>	
<i>clock</i>	
<i>duration</i>	

Definition at line 72 of file result.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



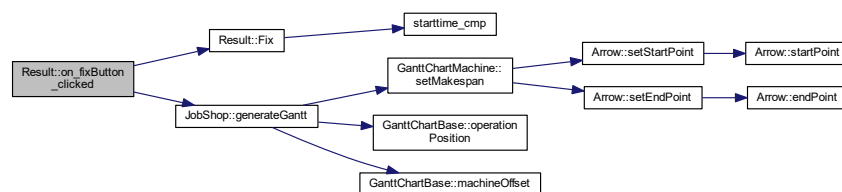
### 6.16.3.2 on\_fixButton\_clicked

```
void Result::on_fixButton_clicked ( ) [private], [slot]
```

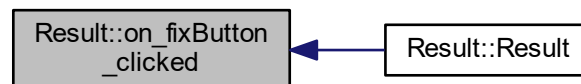
The action of fixButton.

Definition at line 136 of file result.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



## 6.16.4 Member Data Documentation

### 6.16.4.1 jssp

```
JobShop* Result::jssp [private]
```

Definition at line 38 of file result.h.

#### 6.16.4.2 private\_chart

```
GanttChartBase* Result::private_chart [private]
```

Definition at line 39 of file result.h.

#### 6.16.4.3 private\_scene

```
QGraphicsScene* Result::private_scene [private]
```

Definition at line 40 of file result.h.

#### 6.16.4.4 ui

```
Ui::Result* Result::ui [private]
```

Definition at line 37 of file result.h.

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

- include/[result.h](#)
- ui/[result.cpp](#)

## 6.17 SEQUENCE Struct Reference

```
#include <bottle.h>
```

### Public Attributes

- int [job](#) [[MAXJOB](#)]

### 6.17.1 Detailed Description

Job sequences on a machine.

Definition at line 58 of file bottle.h.

### 6.17.2 Member Data Documentation

#### 6.17.2.1 job

```
int SEQUENCE::job [MAXJOB]
```

Job sequences on a machine.

Definition at line 59 of file bottle.h.

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

- include/[bottle.h](#)





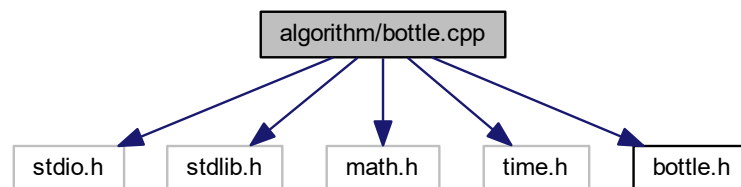
## Chapter 7

# File Documentation

### 7.1 algorithm/bottle.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <bottle.h>
```

Include dependency graph for bottle.cpp:



#### Classes

- struct [BLIST](#)
- struct [MACHINEORDER](#)
- struct [NEIGHBOR](#)

#### Macros

- `#define` [TRY\\_COUNT](#) 10

#### Typedefs

- typedef struct [BLIST](#) [blist\\_t](#)
- typedef struct [MACHINEORDER](#) [mo\\_t](#)
- typedef struct [NEIGHBOR](#) [neighbor\\_t](#)

## Functions

- static void `shifting_bottle_neck` (`sequence_t` \*seq, `mo_t` \*machine\_order, int \*try\_time\_set)  
*The major implementation of the Shifting Bottleneck Procedure, with a backtracing method.*
- static void `clear_and_backup_seq` (`sequence_t` \*seq, int mach, int \*save)
- static void `clear_seq` (`sequence_t` \*seq, int machine)
- static void `cp_mo` (`mo_t` \*mew, `mo_t` \*origin)
- static void `cp_seq` (`sequence_t` \*mew, `sequence_t` \*origin)
- static void `cp_neighbor` (`neighbor_t` \*mew)
- static void `re_optimization_phase_1` (`sequence_t` \*seq, `mo_t` \*mo, int \*makespan)
- static void `set_seq` (`sequence_t` \*seq, int mach, int \*order)
- static void `save_times` (void)
- static void `re_optimization_phase_2` (`sequence_t` \*seq, `mo_t` \*mo, int \*makespan)
- static void `restore_neighbor` (`neighbor_t` \*old)
- static int `critical` (int machine, int makespan)
- int `eval` (`sequence_t` \*seq)

## Variables

- int `best_makespan` = `INFINITAS`  
*Store the best makespan value.*

### 7.1.1 Macro Definition Documentation

#### 7.1.1.1 TRY\_COUNT

```
#define TRY_COUNT 10
```

Definition at line 14 of file bottle.cpp.

### 7.1.2 Typedef Documentation

#### 7.1.2.1 blist\_t

```
typedef struct BLIST blist_t
```

Store the bottle information.

#### 7.1.2.2 mo\_t

```
typedef struct MACHINEORDER mo_t
```

Machine order type.

### 7.1.2.3 neighbor\_t

```
typedef struct NEIGHBOR neighbor_t
```

A temporary struct to store a sequence.

## 7.1.3 Function Documentation

### 7.1.3.1 clear\_and\_backup\_seq()

```
static void clear_and_backup_seq (
    sequence_t * seq,
    int machine,
    int * save ) [inline], [static]
```

Store current sequence of machine N in the given address. Then just clear the sequence.

#### Parameters

<i>seq</i>	Sequence to be cleared.
<i>machine</i>	Current machine number.
<i>save</i>	Address to save the old sequence. A NULL address means the old sequence won't be stored.

Definition at line 366 of file bottle.cpp.

Here is the caller graph for this function:



### 7.1.3.2 clear\_seq()

```
static void clear_seq (
    sequence_t * seq,
    int machine ) [inline], [static]
```

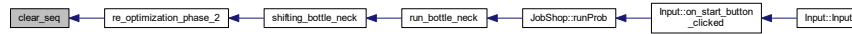
Clear the sequence.

#### Parameters

<i>seq</i>	Sequence to be cleared.
<i>machine</i>	Current machine number.

Definition at line 382 of file bottle.cpp.

Here is the caller graph for this function:



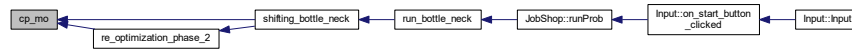
### 7.1.3.3 cp\_mo()

```
static void cp_mo (
    mo_t * mew,
    mo_t * origin ) [inline], [static]
```

Copy the origin machine order to mew.

Definition at line 352 of file bottle.cpp.

Here is the caller graph for this function:



### 7.1.3.4 cp\_neighbor()

```
static void cp_neighbor (
    neighbor_t * mew ) [inline], [static]
```

Store neighbor to a neighbor\_t variable.

Definition at line 421 of file bottle.cpp.

Here is the caller graph for this function:



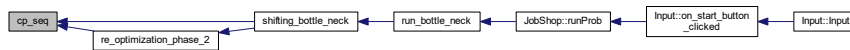
## 7.1.3.5 cp\_seq()

```
static void cp_seq (
    sequence_t * mew,
    sequence_t * origin ) [inline], [static]
```

Copy the origin sequence order to mew.

Definition at line 394 of file bottle.cpp.

Here is the caller graph for this function:



## 7.1.3.6 critical()

```
static int critical (
    int machine,
    int makespan ) [inline], [static]
```

Test whether the machine is the critical machine, which means the end of the procedure of this machine is also the end of all the operations.

## Parameters

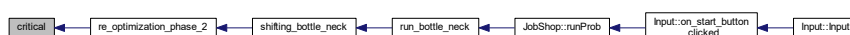
<i>machine</i>	Machine number to be tested.
<i>makespan</i>	The given makespan

## Returns

If the machine is the critical machine, return 1. Else return 0.

Definition at line 342 of file bottle.cpp.

Here is the caller graph for this function:



#### 7.1.3.7 eval()

```
int eval (
    sequence_t * seq )
```

Evaluate the makespan of the given sequence.

**Parameters**

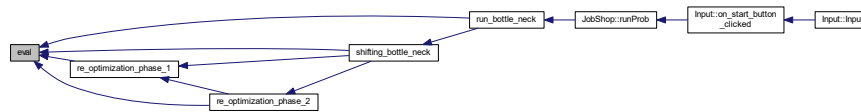
<i>seq</i>	The sequence of job.
------------	----------------------

**Returns**

The makespan of the sequence.

Definition at line 31 of file eval.cpp.

Here is the caller graph for this function:

**7.1.3.8 re\_optimization\_phase\_1()**

```

static void re_optimization_phase_1 (
    sequence_t * seq,
    mo_t * machine_order,
    int * makespan ) [inline], [static]

```

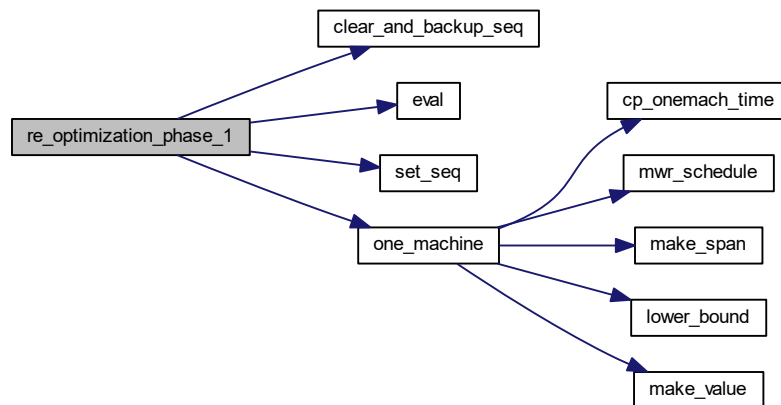
The re-optimization... Phase 1

**Parameters**

<i>seq</i>	The sequence
<i>machine_order</i>	Machine order
<i>makespan</i>	Current makespan

Definition at line 217 of file bottle.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.1.3.9 re\_optimization\_phase\_2()

```

static void re_optimization_phase_2 (
    sequence_t * seq,
    mo_t * machine_order,
    int * makespan ) [inline], [static]
  
```

The re-optimization... Phase 2

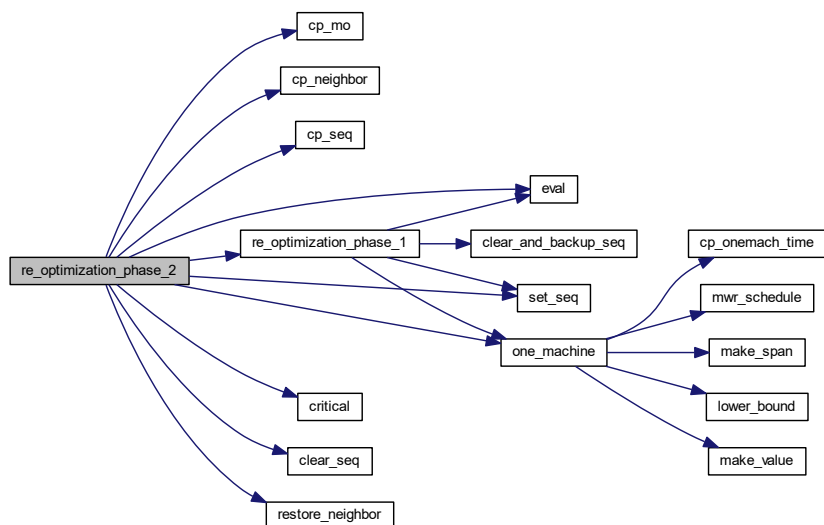
#### Parameters

<i>seq</i>	The sequence
<i>machine_order</i>	Machine order
<i>makespan</i>	Current makespan

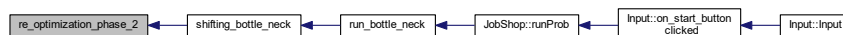
Definition at line 282 of file bottle.cpp.



Here is the call graph for this function:



Here is the caller graph for this function:



### 7.1.3.10 restore\_neighbor()

```
static void restore_neighbor (
    neighbor_t * old ) [inline], [static]
```

Load neighbor from a neighbor\_t variable.

Definition at line 432 of file bottle.cpp.

Here is the caller graph for this function:



### 7.1.3.11 save\_times()

```
static void save_times (
    void ) [inline], [static]
```

Save current start time of each operation.

Definition at line 444 of file bottle.cpp.

Here is the caller graph for this function:



### 7.1.3.12 set\_seq()

```
static void set_seq (
    sequence_t * seq,
    int machine,
    int * order ) [inline], [static]
```

Set sequence by the given order.

#### Parameters

<i>seq</i>	Sequence to be set.
<i>machine</i>	The machine which the sequence relies on.
<i>order</i>	The given order.

Definition at line 407 of file bottle.cpp.

Here is the caller graph for this function:



### 7.1.3.13 shifting\_bottle\_neck()

```
static void shifting_bottle_neck (
    sequence_t * seq,
```

```
mo_t * machine_order,
int * try_time_set ) [inline], [static]
```

The major implementation of the Shifting Bottleneck Procedure, with a backtracing method.

The basic idea of the algorithm can be described as follows: It sequences the machines one by one successively, taking each time the machine identified as a bottleneck among the machine not yet sequenced. Every time after a new machine is sequenced, all previously sequenced sequence will be locally re-optimized. Bottleneck identification and the local re-optimization are both based on solving a one machine scheduling problem, which is more easy than the JSSP. In this implementation a backtracing trick is introduced to improve the quality of the solution, which give us a method to use a slightly more time to run the basic shifting bottleneck procedure more times.

#### Parameters

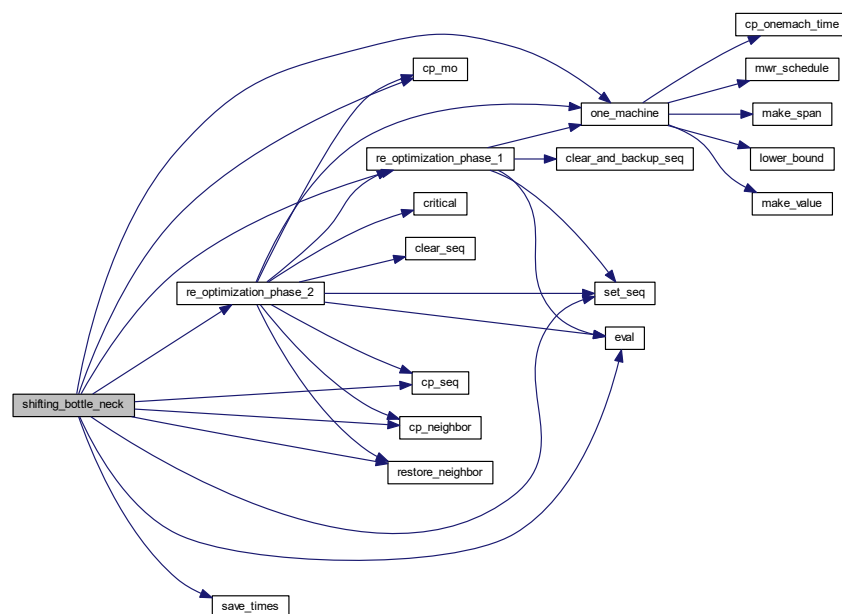
<i>seq</i>	The given sequence list. Will be updated when find a better makespan.
<i>machine_order</i>	Machine order.
<i>try_time_set</i>	Backtracing depth set.

#### Returns

When the procedure is done. You should find the start time of the solution at the "start" field of the struct array job.

Definition at line 117 of file bottle.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



## 7.1.4 Variable Documentation

### 7.1.4.1 best\_makespan

```
best_makespan = INFINITAS
```

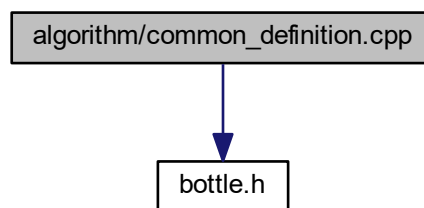
Store the best makespan value.

Definition at line 45 of file bottle.cpp.

## 7.2 algorithm/common\_definition.cpp File Reference

```
#include <bottle.h>
```

Include dependency graph for common\_definition.cpp:



### Variables

- `job_t job [MAXJOB]`
- `int job_size = 1`
- `int machine_size = 1`
- `int terminate_flag = 0`

## 7.2.1 Variable Documentation

### 7.2.1.1 job

`job`

Data representation of all the jobs. All operations runs on this variable.

Definition at line 5 of file common\_definition.cpp.

#### 7.2.1.2 job\_size

```
job_size = 1
```

Job number in this instance.

Definition at line 10 of file common\_definition.cpp.

#### 7.2.1.3 machine\_size

```
machine_size = 1
```

Machine number in this instance.

Definition at line 15 of file common\_definition.cpp.

#### 7.2.1.4 terminate\_flag

```
terminate_flag = 0
```

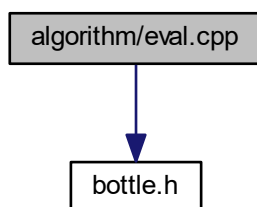
Should we stop???

Definition at line 20 of file common\_definition.cpp.

## 7.3 algorithm/eval.cpp File Reference

```
#include <bottle.h>
```

Include dependency graph for eval.cpp:



### Classes

- struct [JOBMACHINEPAR](#)

## Typedefs

- typedef struct JOBMACHINEPAR job\_machine\_t

## Functions

- int eval (sequence\_t \*seq)

## Variables

- int magicnum = 0

### 7.3.1 Typedef Documentation

#### 7.3.1.1 job\_machine\_t

```
typedef struct JOBMACHINEPAR job_machine_t
```

Auxiliary struct to calculate the makespan.

### 7.3.2 Function Documentation

#### 7.3.2.1 eval()

```
int eval (
    sequence_t * seq )
```

Evaluate the makespan of the given sequence.

#### Parameters

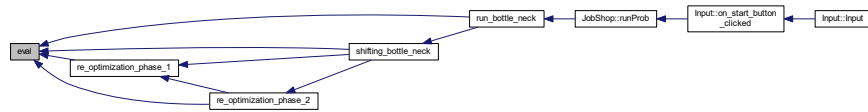
<i>seq</i>	The sequence of job.
------------	----------------------

#### Returns

The makespan of the sequence.

Definition at line 31 of file eval.cpp.

Here is the caller graph for this function:



### 7.3.3 Variable Documentation

#### 7.3.3.1 magicnum

```
magicnum = 0
```

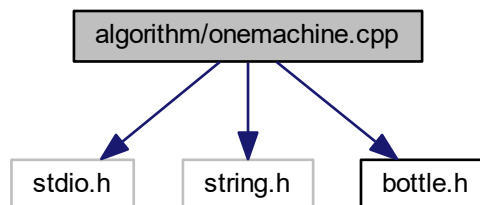
The number generated and managed by the God in the computer. Every one who changed the name of this feild will be seen as an evil and will be cursed by the God.

Definition at line 23 of file eval.cpp.

## 7.4 algorithm/onemachine.cpp File Reference

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

Include dependency graph for onemachine.cpp:



### Classes

- struct [ONEMACHINE\\_BRANCH\\_AND\\_BOUND\\_ASSISTANT](#)

## Macros

- `#define ONEMACH_BBNODES 300`

## Typedefs

- `typedef struct ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT onemach_bb_ass_t`

## Functions

- `static void cp_onemach_time (onemach_times_t *mew, onemach_times_t *origin)`
- `static void mwr_schedule (onemach_times_t one, int *order)`
- `static int lower_bound (onemach_times_t one, int *jset, int jset_size)`
- `static int make_span (onemach_times_t one, int *order, int *jset, int *jset_size, int *cjob, int *pjob, int *make)`
- `static int make_value (onemach_times_t one, int *order)`
- `int one_machine (onemach_times_t one, int *bestorder)`

### 7.4.1 Macro Definition Documentation

#### 7.4.1.1 ONEMACH\_BBNODES

```
#define ONEMACH_BBNODES 300
```

Nodes number of the branch and bound tree to solve the one machine sequencing problem.

Definition at line 17 of file onemachine.cpp.

### 7.4.2 Typedef Documentation

#### 7.4.2.1 onemach\_bb\_ass\_t

```
typedef struct ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT onemach_bb_ass_t
```

Info of node of a branch and bound tree.

### 7.4.3 Function Documentation



## 7.4.3.1 cp\_onemach\_time()

```
static void cp_onemach_time (
    onemach_times_t * mew,
    onemach_times_t * origin ) [inline], [static]
```

Copy origin onemach\_times struct to mew By practice, change from memcpy to just write what we want to do is very important...

Definition at line 159 of file onemachine.cpp.

Here is the caller graph for this function:



## 7.4.3.2 lower\_bound()

```
static int lower_bound (
    onemach_times_t one,
    int * job_set,
    int job_set_size ) [inline], [static]
```

Find the lower bound of the given machine on the given job order.

## Parameters

<i>one</i>	The representation of the given machine.
<i>job_set</i>	The set of job.
<i>job_set_size</i>	The size of job_set

## Returns

Lowerbound of the machine. Which is just the sum of minimum estimate and minimum mhtime and the sum of all the process time.

Definition at line 245 of file onemachine.cpp.

Here is the caller graph for this function:



### 7.4.3.3 make\_span()

```
static int make_span (
    onemach_times_t one,
    int * order,
    int * job_set,
    int * job_set_size,
    int * critical_job_order,
    int * terminate_job_order,
    int * make ) [inline], [static]
```

Test if the job order is feasible and compute the make\_span.

#### Parameters

<i>one</i>	The representation of the given machine.
<i>order</i>	The given job order.
<i>job_set</i>	The set of job on the machine.
<i>job_set_size</i>	The size of job_set.
<i>critical_job_order</i>	
<i>terminate_job_order</i>	
<i>make</i>	The make_span.

#### Returns

If the order is OK return 1,else return 0.

Definition at line 277 of file onemachine.cpp.

Here is the caller graph for this function:



### 7.4.3.4 make\_value()

```
static int make_value (
    onemach_times_t one,
    int * order ) [inline], [static]
```

Compute the makespan of the given job order.

#### Parameters

<i>one</i>	Representation of the machine.
<i>order</i>	The given job order.

Definition at line 222 of file onemachine.cpp.

Here is the caller graph for this function:



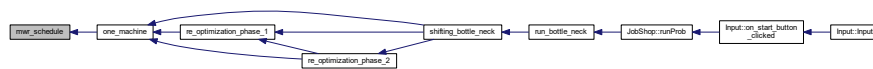
#### 7.4.3.5 mwr\_schedule()

```
static void mwr_schedule (
    onemach_times_t one,
    int * order ) [inline], [static]
```

Algorithm to find the most work remaining schedule by Schrage

Definition at line 170 of file onemachine.cpp.

Here is the caller graph for this function:



#### 7.4.3.6 one\_machine()

```
int one_machine (
    onemach_times_t one,
    int * bestorder )
```

The one-machine sequencing algorithm from "The one-machine sequencing problem" by Jacques Carlier.

##### Parameters

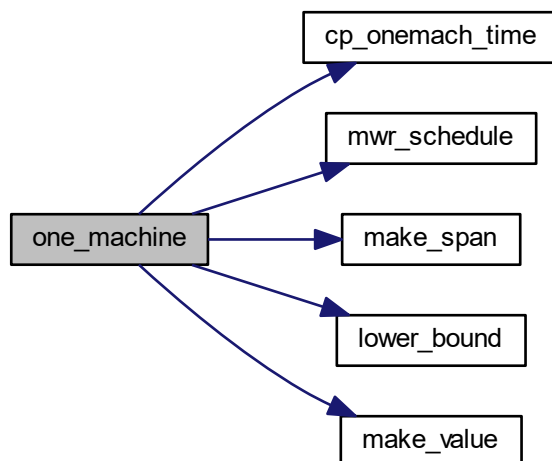
<i>one</i>	Representation of the machine.
<i>bestorder</i>	Best job order

##### Returns

makespan

Definition at line 41 of file onemachine.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



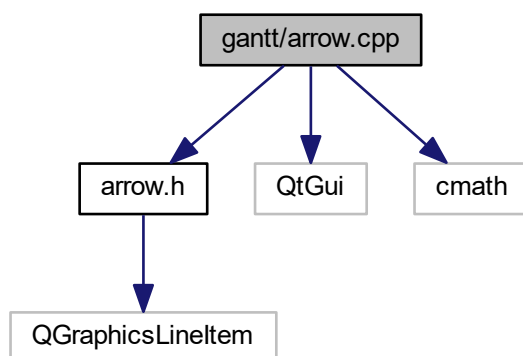
## 7.5 gantt/arrow.cpp File Reference

```

#include <arrow.h>
#include <QtGui>
#include <cmath>

```

Include dependency graph for arrow.cpp:



## Macros

- `#define` [USE\\_MATH\\_DEFINES](#)

## Variables

- `const qreal` [Pi](#) = `M_PI`
- `const qreal` [arrowSize](#) = 10

### 7.5.1 Detailed Description

Draw arrow.

Author

Name1e5s

### 7.5.2 Macro Definition Documentation

#### 7.5.2.1 USE\_MATH\_DEFINES

```
#define USE_MATH_DEFINES
```

Definition at line 11 of file `arrow.cpp`.

### 7.5.3 Variable Documentation

#### 7.5.3.1 arrowSize

```
const qreal arrowSize = 10
```

Definition at line 15 of file arrow.cpp.

#### 7.5.3.2 Pi

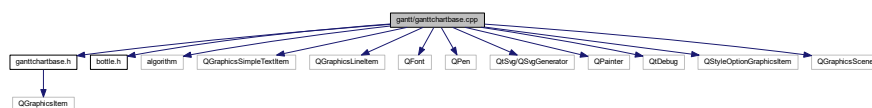
```
const qreal Pi = M_PI
```

Definition at line 14 of file arrow.cpp.

## 7.6 gantt/ganttchartbase.cpp File Reference

```
#include <ganttchartbase.h>
#include <bottle.h>
#include <algorithm>
#include <QGraphicsSimpleTextItem>
#include <QGraphicsLineItem>
#include <QFont>
#include <QPen>
#include <QtSvg/QSvgGenerator>
#include <QPainter>
#include <QtDebug>
#include <QStyleOptionGraphicsItem>
#include <QGraphicsScene>
```

Include dependency graph for ganttchartbase.cpp:



### 7.6.1 Detailed Description

Basic definition of our gantt chart.

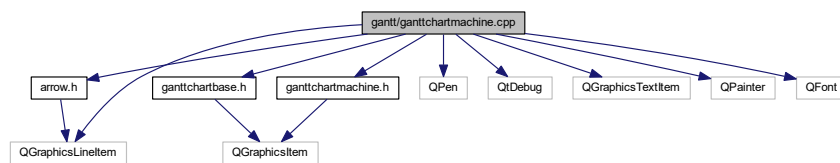
Author

Name1e5s

## 7.7 gantt/ganttchartmachine.cpp File Reference

```
#include <arrow.h>
#include <ganttchartbase.h>
#include <ganttchartmachine.h>
#include <QPen>
#include <QtDebug>
#include <QGraphicsTextItem>
#include <QGraphicsLineItem>
#include <QPainter>
#include <QFont>
```

Include dependency graph for ganttchartmachine.cpp:



### 7.7.1 Detailed Description

Draw machine for our gantt chart.

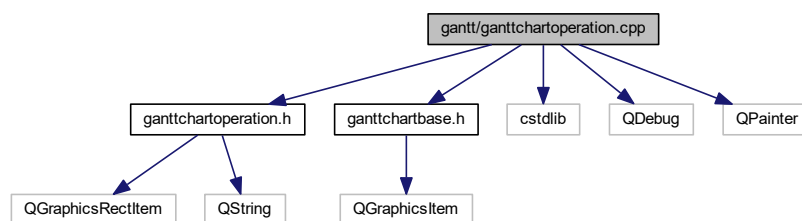
Author

Name1e5s

## 7.8 gantt/ganttchartoperation.cpp File Reference

```
#include <ganttchartoperation.h>
#include <ganttchartbase.h>
#include <cstdlib>
#include <QDebug>
#include <QPainter>
```

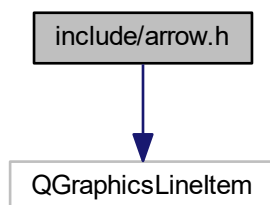
Include dependency graph for ganttchartoperation.cpp:



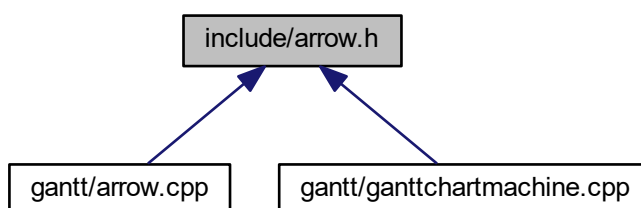
## 7.9 include/arrow.h File Reference

```
#include <QGraphicsLineItem>
```

Include dependency graph for arrow.h:



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



### Classes

- class [Arrow](#)

### 7.9.1 Detailed Description

This file defines how to draw an arrow on the gantt chart.

#### Author

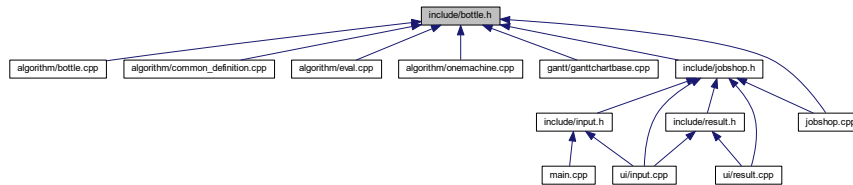
Name1e5s



## 7.10 include/bottle.h File Reference

Header file for the whole project.

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



### Classes

- struct [JOB](#)
- struct [SEQUENCE](#)
- struct [ONEMACHINestime](#)

### Macros

- `#define` [MAXJOB](#) 30
- `#define` [MAXMACHINE](#) 30
- `#define` [INFINITAS](#) 0x7ffffff
- `#define` [MAX](#)(a, b) ((a) > (b) ? (a) : (b))

### Typedefs

- `typedef struct` [JOB](#) [job\\_t](#)
- `typedef struct` [SEQUENCE](#) [sequence\\_t](#)
- `typedef struct` [ONEMACHINestime](#) [onemach\\_times\\_t](#)

### Functions

- void [prestissimo](#) (void)
- void [run\\_bottle\\_neck](#) (void)
- int [one\\_machine](#) ([onemach\\_times\\_t](#) one, int \*bestorder)

### Variables

- [job\\_t](#) [job](#) [[MAXJOB](#)]
- int [job\\_size](#)
- int [machine\\_size](#)
- int [terminate\\_flag](#)

### 7.10.1 Detailed Description

Header file for the whole project.

A Simple Old-fashion Implementation Of The Well-known Shifting Bottleneck Procedure For Job Shop Scheduling Problem(JSSP).The codes are based on "The Shifting Bottleneck Procedure for Job Shop Scheduling" by J. Adams et al.

#### Author

Name1e5s

### 7.10.2 Macro Definition Documentation

#### 7.10.2.1 INFINITAS

```
#define INFINITAS 0x7fffffff
```

A integer that can be seen as infinity – should be bigger than the biggest makespan of all the instances. Hence, 0x7fffffff (a.k.a INT\_MAX) is a good choice

Definition at line 33 of file bottle.h.

#### 7.10.2.2 MAX

```
#define MAX(  
    a,  
    b ) ((a) > (b) ? (a) : (b))
```

A regular macro that returns the bigger value bewteen a and b.

Definition at line 38 of file bottle.h.

#### 7.10.2.3 MAXJOB

```
#define MAXJOB 30
```

The most jobs this program can handle.

Definition at line 21 of file bottle.h.

#### 7.10.2.4 MAXMACHINE

```
#define MAXMACHINE 30
```

The most machines this program can handle.

Definition at line 26 of file bottle.h.

### 7.10.3 Typedef Documentation

#### 7.10.3.1 job\_t

```
typedef struct JOB job_t
```

Data representation for a job.

#### 7.10.3.2 onemach\_times\_t

```
typedef struct ONEMACHINestime onemach_times_t
```

Store the time info for every job runs on the same machine.

#### 7.10.3.3 sequence\_t

```
typedef struct SEQUENCE sequence_t
```

Job sequences on a machine.

### 7.10.4 Function Documentation

#### 7.10.4.1 one\_machine()

```
int one_machine (
    onemach_times_t one,
    int * bestorder )
```

The one-machine sequencing algorithm from "The one-machine sequencing problem" by Jacques Carlier.

##### Parameters

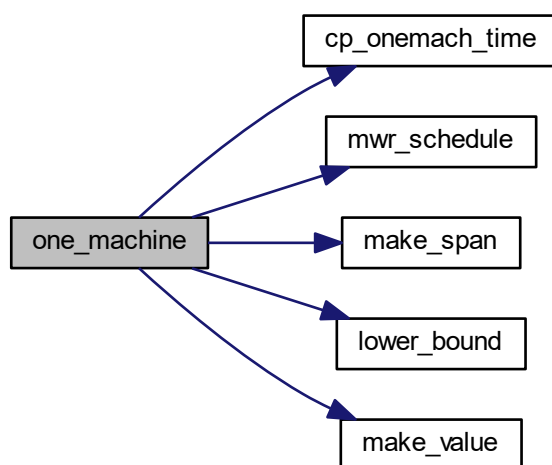
<i>one</i>	Representation of the machine.
<i>bestorder</i>	Best job order

## Returns

makespan

Definition at line 41 of file onemachine.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 7.10.4.2 prestissimo()

```
void prestissimo (
    void )
```

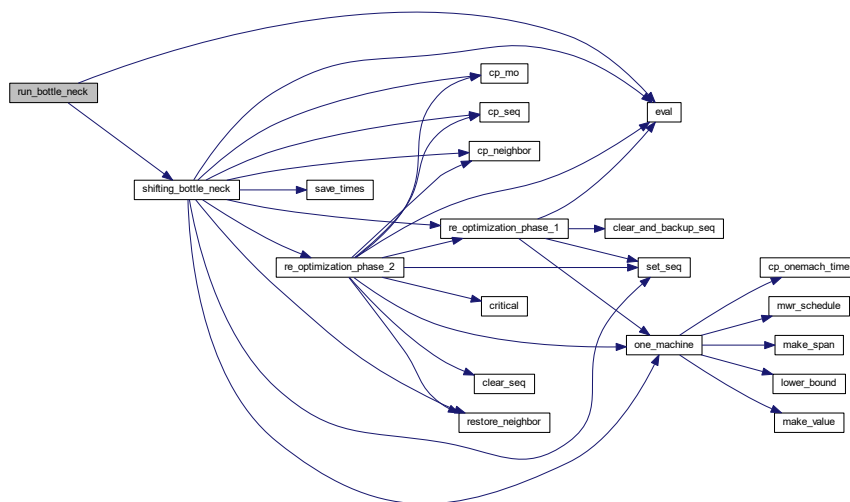
### 7.10.4.3 run\_bottle\_neck()

```
void run_bottle_neck (
    void )
```

Driver of the Shifting Bottleneck Procedure We can change here to have a balance bewteen run time and makespan...

Definition at line 65 of file bottle.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



## 7.10.5 Variable Documentation

### 7.10.5.1 job

`job_t job[MAXJOB]`

Data representation of all the jobs. All operations runs on this variable.

Definition at line 5 of file common\_definition.cpp.

### 7.10.5.2 job\_size

```
int job_size
```

Job number in this instance.

Definition at line 10 of file common\_definition.cpp.

### 7.10.5.3 machine\_size

```
int machine_size
```

Machine number in this instance.

Definition at line 15 of file common\_definition.cpp.

### 7.10.5.4 terminate\_flag

```
int terminate_flag
```

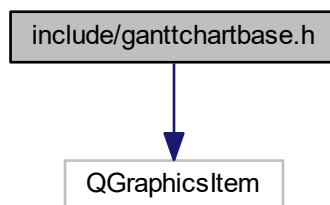
Should we stop???

Definition at line 20 of file common\_definition.cpp.

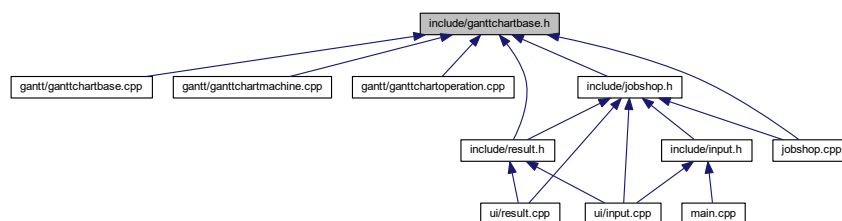
## 7.11 include/ganttchartbase.h File Reference

```
#include <QGraphicsItem>
```

Include dependency graph for ganttchartbase.h:



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



## Classes

- class [GanttChartBase](#)

### 7.11.1 Detailed Description

This file contains the base class of the gantt chart.

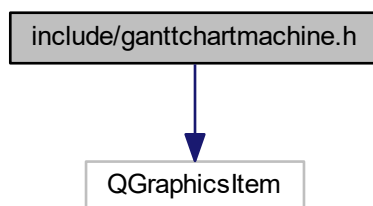
#### Author

Name1e5s

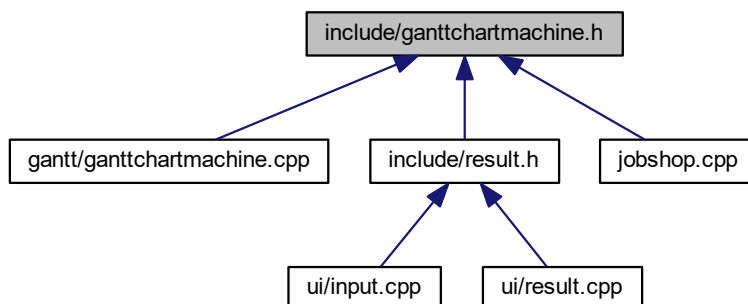
## 7.12 include/ganttchartmachine.h File Reference

```
#include <QGraphicsItem>
```

Include dependency graph for ganttchartmachine.h:



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



## Classes

- class [GanttChartMachine](#)

### 7.12.1 Detailed Description

This file defines how to present the gantt chart per machine.

#### Author

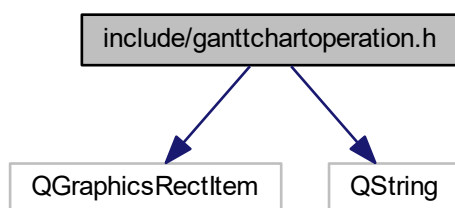
Name1e5s

### 7.13 include/ganttchartoperation.h File Reference

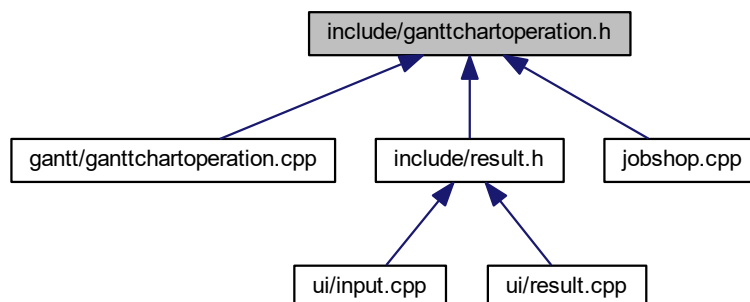
```
#include <QGraphicsRectItem>
```

```
#include <QString>
```

Include dependency graph for ganttchartoperation.h:



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





## Classes

- class [GanttChartOperation](#)

### 7.13.1 Detailed Description

This file defines how to present a operation on the m=gantt chart.

#### Author

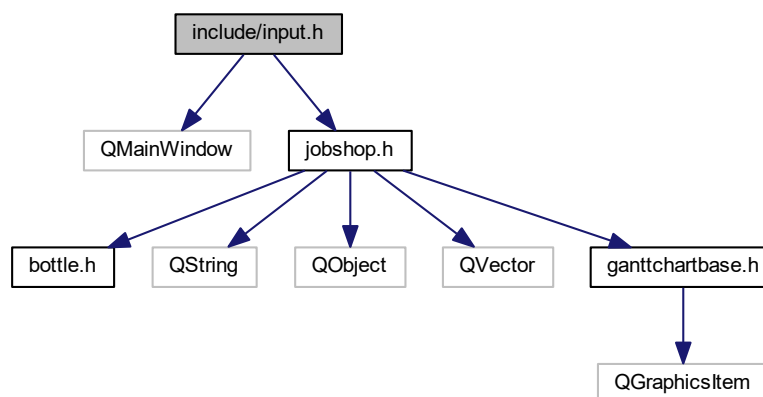
Name1e5s

## 7.14 include/input.h File Reference

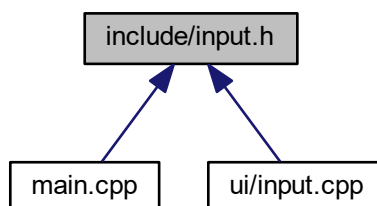
```
#include <QMainWindow>
```

```
#include <jobshop.h>
```

Include dependency graph for input.h:



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



## Classes

- class [Input](#)

## Namespaces

- [Ui](#)

### 7.14.1 Detailed Description

This file defines the input window.

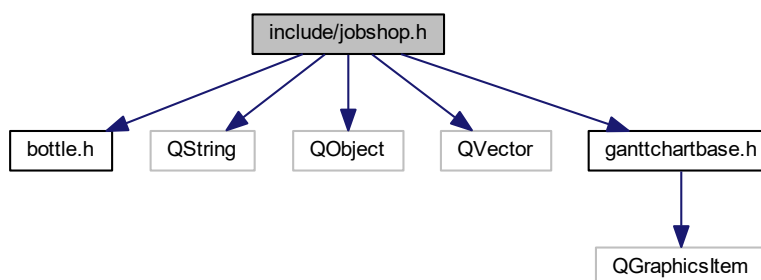
#### Author

Name1e5s

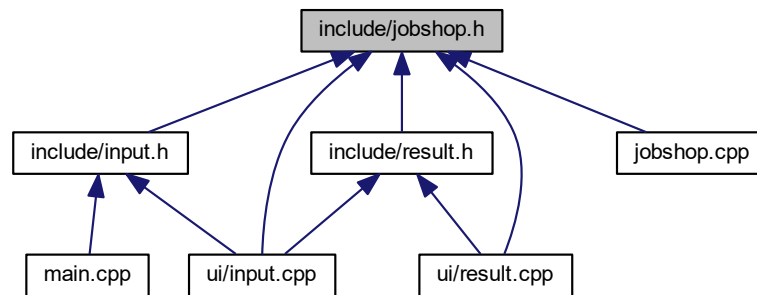
### 7.15 `include/jobshop.h` File Reference

```
#include <bottle.h>
#include <QString>
#include <QObject>
#include <QVector>
#include <gantchartbase.h>
```

Include dependency graph for `jobshop.h`:



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



## Classes

- struct [Fixer](#)
- class [JobShop](#)

### 7.15.1 Detailed Description

This file is a simple wrapper for the previous CLI version of this project to make it compatible with QT.

!!!Note: This implementation is not a good practice.

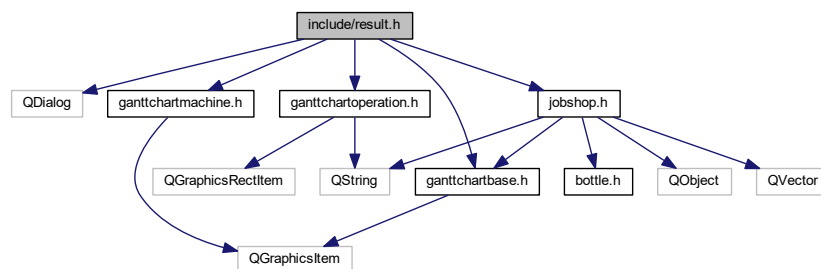
#### Author

Name1e5s

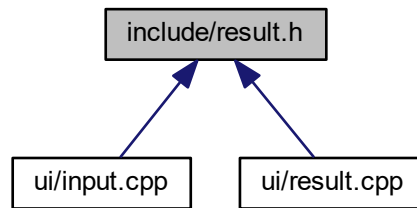
## 7.16 include/result.h File Reference

```
#include <QDialog>
#include <ganttchartbase.h>
#include <ganttchartmachine.h>
#include <ganttchartoperation.h>
#include <jobshop.h>
```

Include dependency graph for result.h:



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



## Classes

- class [Result](#)

## Namespaces

- [Ui](#)

### 7.16.1 Detailed Description

This file defines a simple dialog to show our Gantt chart and it allows user to fix a machins at a given time. The format of the fix command is: `$(time) [machine to be fixed] [duration]`

#### Author

Name1e5s

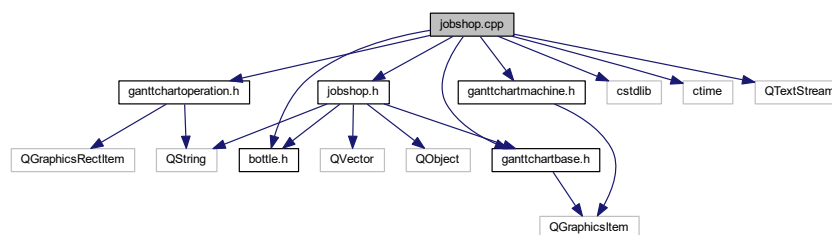
## 7.17 jobshop.cpp File Reference

```

#include <jobshop.h>
#include <bottle.h>
#include <gantchartbase.h>
#include <gantchartmachine.h>
#include <gantchartoperation.h>
#include <cstdlib>
#include <ctime>
#include <QTextStream>

```

Include dependency graph for jobshop.cpp:



## Variables

- int [best\\_makespan](#)  
Store the best makespan value.

### 7.17.1 Detailed Description

Simple wrapper of previous version of JSSP solver.

#### Author

Name1e5s

### 7.17.2 Variable Documentation

#### 7.17.2.1 best\_makespan

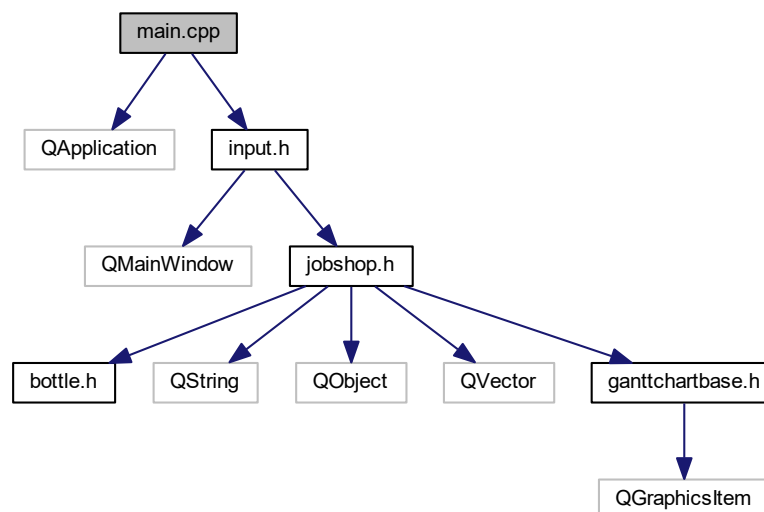
```
int best_makespan
```

Store the best makespan value.

Definition at line 45 of file bottle.cpp.

## 7.18 main.cpp File Reference

```
#include <QApplication>
#include <input.h>
Include dependency graph for main.cpp:
```



## Functions

- int [main](#) (int ac, char \*av[])

### 7.18.1 Detailed Description

Enterpoint of the program.

#### Author

Name1e5s

### 7.18.2 Function Documentation

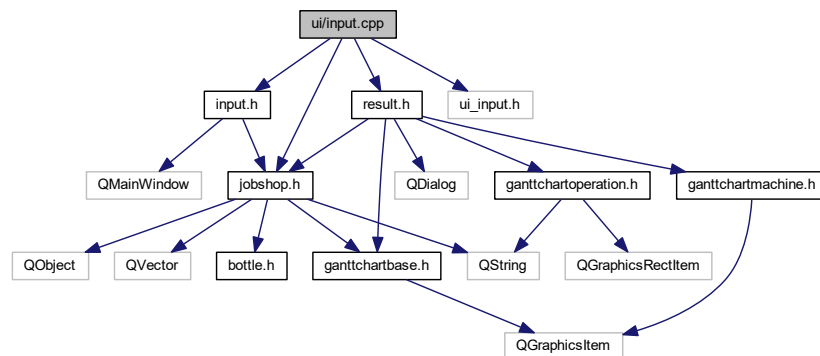
#### 7.18.2.1 main()

```
int main (
    int ac,
    char * av[] )
```

Definition at line 10 of file main.cpp.

## 7.19 ui/input.cpp File Reference

```
#include <input.h>
#include "ui_input.h"
#include <jobshop.h>
#include <result.h>
Include dependency graph for input.cpp:
```



### 7.19.1 Detailed Description

Draw input window.

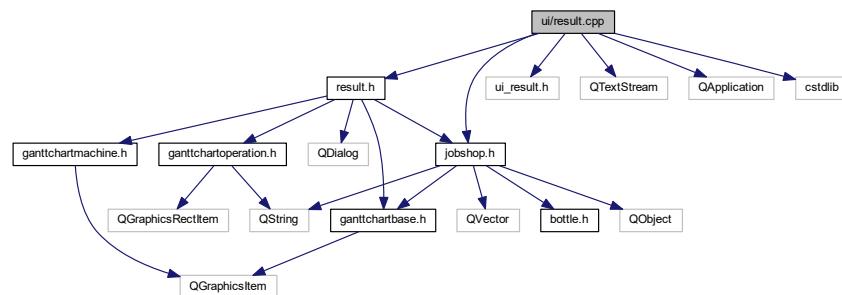
Author

Name1e5s

## 7.20 ui/result.cpp File Reference

```
#include <result.h>
#include "ui_result.h"
#include <jobshop.h>
#include <QTextStream>
#include <QApplication>
#include <cstdlib>
```

Include dependency graph for result.cpp:



### Classes

- struct [pair](#)

*The pair struct for sorting machines.*

### Functions

- int [starttime\\_cmp](#) (const void \*a, const void \*b)

*Function to compare starttime of two pairs for qsort.*

### Variables

- int [best\\_makespan](#)

*Store the best makespan value.*

### 7.20.1 Detailed Description

Draw result dialog.

Author

Name1e5s

### 7.20.2 Function Documentation

#### 7.20.2.1 starttime\_cmp()

```
int starttime_cmp (
    const void * a,
    const void * b )
```

Function to compare starttime of two pairs for qsort.

#### Parameters

<i>a</i>	The first pair.
<i>b</i>	The second pair.

#### Returns

If start time of a is lesser than b, then return a positive value, else return a non-positive value.

Definition at line 33 of file result.cpp.

Here is the caller graph for this function:



### 7.20.3 Variable Documentation

#### 7.20.3.1 best\_makespan

```
int best_makespan
```

Store the best makespan value.

Definition at line 45 of file bottle.cpp.



# Index

- ~GanttChartMachine
  - GanttChartMachine, 24
- ~Input
  - Input, 31
- ~Result
  - Result, 48
- active
  - ONEMACHINE\_BRANCH\_AND\_BOUND\_ASSI←STANT, 44
- algorithm/bottle.cpp, 53
- algorithm/common\_definition.cpp, 64
- algorithm/eval.cpp, 65
- algorithm/onemachine.cpp, 67
- Arrow, 11
  - Arrow, 13
  - boundingRect, 14
  - endPoint, 14
  - paint, 14
  - private\_arrowHead, 16
  - private\_endPoint, 16
  - private\_startPoint, 16
  - setEndPoint, 14
  - setStartPoint, 15
  - shape, 15
  - startPoint, 16
  - Type, 12
  - type, 16
- arrow.cpp
  - arrowSize, 74
  - Pi, 74
  - USE\_MATH\_DEFINES, 73
- arrowSize
  - arrow.cpp, 74
- BLIST, 17
  - machine, 17
  - makespan, 17
  - order, 17
- best\_makespan
  - bottle.cpp, 64
  - jobshop.cpp, 89
  - result.cpp, 92
- blist\_t
  - bottle.cpp, 54
- bottle.cpp
  - best\_makespan, 64
  - blist\_t, 54
  - clear\_and\_backup\_seq, 55
  - clear\_seq, 55
  - cp\_mo, 56
  - cp\_neighbor, 56
  - cp\_seq, 56
  - critical, 57
  - eval, 57
  - mo\_t, 54
  - neighbor\_t, 54
  - re\_optimization\_phase\_1, 59
  - re\_optimization\_phase\_2, 60
  - restore\_neighbor, 61
  - save\_times, 61
  - set\_seq, 62
  - shifting\_bottle\_neck, 62
  - TRY\_COUNT, 54
- bottle.h
  - INFINITAS, 78
  - job, 81
  - job\_size, 81
  - job\_t, 79
  - MAXJOB, 78
  - MAXMACHINE, 78
  - MAX, 78
  - machine\_size, 82
  - one\_machine, 79
  - onemach\_times\_t, 79
  - prestissimo, 80
  - run\_bottle\_neck, 80
  - sequence\_t, 79
  - terminate\_flag, 82
- bound
  - ONEMACHINE\_BRANCH\_AND\_BOUND\_ASSI←STANT, 44
- boundingRect
  - Arrow, 14
  - GanttChartBase, 20
  - GanttChartMachine, 25
- clear\_and\_backup\_seq
  - bottle.cpp, 55
- clear\_seq
  - bottle.cpp, 55
- common\_definition.cpp
  - job, 64
  - job\_size, 64
  - machine\_size, 65
  - terminate\_flag, 65
- cp\_mo
  - bottle.cpp, 56
- cp\_neighbor
  - bottle.cpp, 56

- cp\_onemach\_time
  - onemachine.cpp, 68
- cp\_seq
  - bottle.cpp, 56
- critical
  - bottle.cpp, 57
- duration
  - Fixer, 18
- endPoint
  - Arrow, 14
- endtime
  - pair, 46
- estime
  - JOB, 33
  - ONEMACHINestime, 45
- eval
  - bottle.cpp, 57
  - eval.cpp, 66
- eval.cpp
  - eval, 66
  - job\_machine\_t, 66
  - magicnum, 67
- Fix
  - Result, 49
- Fixer, 18
  - duration, 18
  - machine, 18
  - starttime, 18
- fixer
  - JobShop, 41
- gantt/arrow.cpp, 72
- gantt/ganttchartbase.cpp, 74
- gantt/ganttchartmachine.cpp, 75
- gantt/ganttchartoperation.cpp, 75
- GanttChartBase, 19
  - boundingRect, 20
  - GanttChartBase, 20
  - machineHeight, 22
  - machineHorizontalOffset, 22
  - machineOffset, 21
  - makespan, 22
  - operationHeight, 22
  - operationPosition, 21
  - paint, 21
  - widthUnit, 22
- GanttChartMachine, 23
  - ~GanttChartMachine, 24
  - boundingRect, 25
  - GanttChartMachine, 24
  - machine\_num, 26
  - makespan, 26
  - paint, 25
  - private\_arrow, 27
  - setMakespan, 25
- GanttChartOperation, 27
  - GanttChartOperation, 28
  - m\_color, 29
  - m\_id, 29
  - paint, 28
- generateGantt
  - JobShop, 37
- getJob
  - JobShop, 38
- getJobSize
  - JobShop, 38
- getMachineSize
  - JobShop, 38
- getProb
  - JobShop, 39
- INFINITAS
  - bottle.h, 78
- include/arrow.h, 76
- include/bottle.h, 77
- include/ganttchartbase.h, 82
- include/ganttchartmachine.h, 83
- include/ganttchartoperation.h, 84
- include/input.h, 85
- include/jobshop.h, 86
- include/result.h, 87
- Input, 30
  - ~Input, 31
  - Input, 31
  - jssp, 32
  - on\_start\_button\_clicked, 32
  - ui, 32
- JOBMACHINEPAR, 35
  - job, 35
  - machine, 36
- JOB, 33
  - estime, 33
  - magic, 33
  - mhtime, 34
  - next, 34
  - order, 34
  - prev, 34
  - process\_time, 34
  - start, 35
  - step, 35
- job
  - bottle.h, 81
  - common\_definition.cpp, 64
  - JOBMACHINEPAR, 35
  - SEQUENCE, 51
- job\_machine\_t
  - eval.cpp, 66
- job\_size
  - bottle.h, 81
  - common\_definition.cpp, 64
- job\_t
  - bottle.h, 79
- JobShop, 36
  - fixer, 41

- generateGantt, 37
- getJob, 38
- getJobSize, 38
- getMachineSize, 38
- getProb, 39
- JobShop, 37
- m\_job, 41
- m\_job\_size, 42
- m\_machine\_size, 42
- runProb, 39
- setJobSize, 40
- setMachineSize, 41
- jobshop.cpp, 88
  - best\_makespan, 89
- jssp
  - Input, 32
  - Result, 50
- lower\_bound
  - onemachine.cpp, 69
- m\_color
  - GanttChartOperation, 29
- m\_id
  - GanttChartOperation, 29
- m\_job
  - JobShop, 41
- m\_job\_size
  - JobShop, 42
- m\_machine\_size
  - JobShop, 42
- MACHINEORDER, 42
  - machines, 42
  - size, 43
- MAXJOB
  - bottle.h, 78
- MAXMACHINE
  - bottle.h, 78
- MAX
  - bottle.h, 78
- machine
  - BLIST, 17
  - Fixer, 18
  - JOBMACHINEPAR, 36
- machine\_num
  - GanttChartMachine, 26
- machine\_size
  - bottle.h, 82
  - common\_definition.cpp, 65
- machineHeight
  - GanttChartBase, 22
- machineHorizontalOffset
  - GanttChartBase, 22
- machineOffset
  - GanttChartBase, 21
- machines
  - MACHINEORDER, 42
- magic
  - JOB, 33
- magicnum
  - eval.cpp, 67
- main
  - main.cpp, 90
- main.cpp, 89
  - main, 90
- make\_span
  - onemachine.cpp, 69
- make\_value
  - onemachine.cpp, 70
- makespan
  - BLIST, 17
  - GanttChartBase, 22
  - GanttChartMachine, 26
- mhtime
  - JOB, 34
  - ONEMACHINestime, 45
- mo\_t
  - bottle.cpp, 54
- mwr\_schedule
  - onemachine.cpp, 71
- NEIGHBOR, 43
  - next, 43
  - prev, 43
- neighbor\_t
  - bottle.cpp, 54
- next
  - JOB, 34
  - NEIGHBOR, 43
- ONEMACH\_BBNODES
  - onemachine.cpp, 68
- ONEMACHINE\_BRANCH\_AND\_BOUND\_ASSISTANT, 44
  - active, 44
  - bound, 44
- ONEMACHINestime, 45
  - estime, 45
  - mhtime, 45
  - process\_time, 45
- on\_fixButton\_clicked
  - Result, 50
- on\_start\_button\_clicked
  - Input, 32
- one\_machine
  - bottle.h, 79
  - onemachine.cpp, 71
- onemach\_bb\_ass\_t
  - onemachine.cpp, 68
- onemach\_times\_t
  - bottle.h, 79
- onemachine.cpp
  - cp\_onemach\_time, 68
  - lower\_bound, 69
  - make\_span, 69
  - make\_value, 70
  - mwr\_schedule, 71
  - ONEMACH\_BBNODES, 68

- one\_machine, 71
  - onemach\_bb\_ass\_t, 68
- operationHeight
  - GanttChartBase, 22
- operationPosition
  - GanttChartBase, 21
- order
  - BLIST, 17
  - JOB, 34
- paint
  - Arrow, 14
  - GanttChartBase, 21
  - GanttChartMachine, 25
  - GanttChartOperation, 28
- pair, 46
  - endtime, 46
  - starttime, 46
- Pi
  - arrow.cpp, 74
- prestissimo
  - bottle.h, 80
- prev
  - JOB, 34
  - NEIGHBOR, 43
- private\_arrow
  - GanttChartMachine, 27
- private\_arrowHead
  - Arrow, 16
- private\_chart
  - Result, 50
- private\_endPoint
  - Arrow, 16
- private\_scene
  - Result, 51
- private\_startPoint
  - Arrow, 16
- process\_time
  - JOB, 34
  - ONEMACHINestime, 45
- re\_optimization\_phase\_1
  - bottle.cpp, 59
- re\_optimization\_phase\_2
  - bottle.cpp, 60
- restore\_neighbor
  - bottle.cpp, 61
- Result, 47
  - ~Result, 48
  - Fix, 49
  - jssp, 50
  - on\_fixButton\_clicked, 50
  - private\_chart, 50
  - private\_scene, 51
  - Result, 48
  - ui, 51
- result.cpp
  - best\_makespan, 92
  - starttime\_cmp, 92
- run\_bottle\_neck
  - bottle.h, 80
- runProb
  - JobShop, 39
- SEQUENCE, 51
  - job, 51
- save\_times
  - bottle.cpp, 61
- sequence\_t
  - bottle.h, 79
- set\_seq
  - bottle.cpp, 62
- setEndPoint
  - Arrow, 14
- setJobSize
  - JobShop, 40
- setMachineSize
  - JobShop, 41
- setMakespan
  - GanttChartMachine, 25
- setStartPoint
  - Arrow, 15
- shape
  - Arrow, 15
- shifting\_bottle\_neck
  - bottle.cpp, 62
- size
  - MACHINEORDER, 43
- start
  - JOB, 35
- startPoint
  - Arrow, 16
- starttime
  - Fixer, 18
  - pair, 46
- starttime\_cmp
  - result.cpp, 92
- step
  - JOB, 35
- TRY\_COUNT
  - bottle.cpp, 54
- terminate\_flag
  - bottle.h, 82
  - common\_definition.cpp, 65
- Type
  - Arrow, 12
- type
  - Arrow, 16
- USE\_MATH\_DEFINES
  - arrow.cpp, 73
- Ui, 9
- ui
  - Input, 32
  - Result, 51
- ui/input.cpp, 90
- ui/result.cpp, 91

widthUnit  
    GanttChartBase, [22](#)