My Project

Generated by Doxygen 1.8.17

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

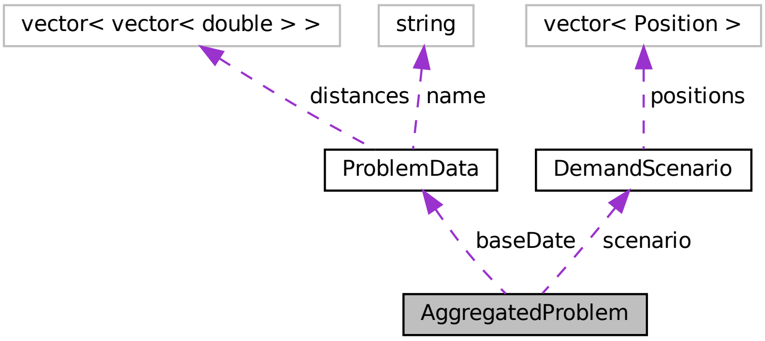
Chapter 4

Class Documentation

4.1 AggregatedProblem Class Reference

`#include <StochasticInfo.h>`

Collaboration diagram for AggregatedProblem:



Private Attributes

- ProblemData baseDate
- DemandScenario scenario

4.1.1 Member Data Documentation

4.1.1.1 baseDate

```
ProblemData AggregatedProblem::baseDate  [private]
```

4.1.1.2 scenario

```
DemandScenario AggregatedProblem::scenario  [private]
```

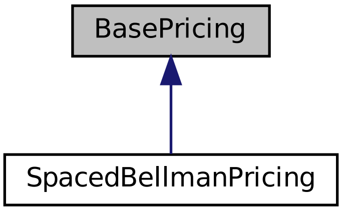
The documentation for this class was generated from the following file:

- include/StochasticInfo.h

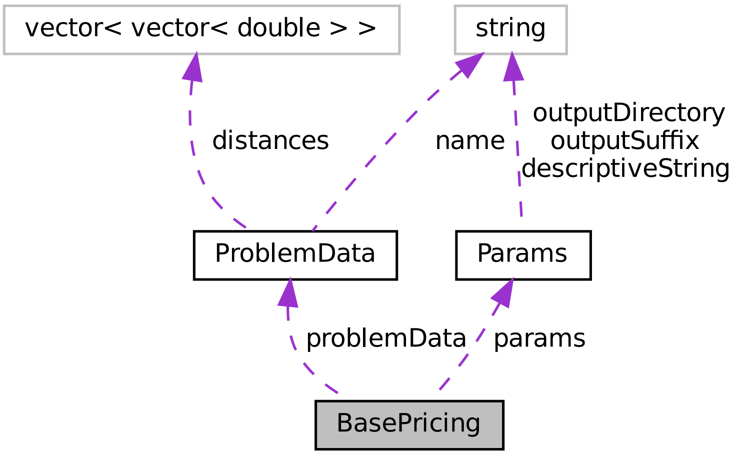
4.2 BasePricing Class Reference

```
#include <BasePricing.h>
```

Inheritance diagram for BasePricing:



Collaboration diagram for BasePricing:



Public Member Functions

- void SetMaxTime (double max_time)
- void SetMaxMemory (int max_memory)
- BasePricing ()
- virtual PricingReturn Price (int vehicle_id, int n_routes, vector< double > &alpha_duals, vector< double > &beta_duals, vector< Route > &outRoutes, vector< int > consideredRequests, set< pair< int, int >> forbiddenEdges, map< pair< int, int >, double > edgeDuals)=0
- PricingReturn Price (int vehicle_id, int n_routes, vector< double > &alpha_duals, vector< double > &beta↩_duals, vector< Route > &outRoutes)
- virtual ∼BasePricing ()

Public Attributes

- ProblemData ∗ problemData

Protected Attributes

- Params ∗ params
- double max_time
- int max_memory

4.2.1 Constructor & Destructor Documentation

4.2.1.1 BasePricing()

```
BasePricing::BasePricing ( )  [inline]
```

4.2.1.2 ∼BasePricing()

```
virtual BasePricing::∼BasePricing ( )  [inline], [virtual]
```

4.2.2 Member Function Documentation

4.2.2.1 Price() [1/2]

```
PricingReturn BasePricing::Price (
              int vehicle_id,
              int n_routes,
              vector< double > & alpha_duals,
              vector< double > & beta_duals,
              vector< Route > & outRoutes )  [inline]
```

4.2.2.2 Price() [2/2]

```
virtual PricingReturn BasePricing::Price (
            int vehicle_id,
            int n_routes,
            vector< double > & alpha_duals,
            vector< double > & beta_duals,
            vector< Route > & outRoutes,
            vector< int > consideredRequests,
            set< pair< int, int >> forbiddenEdges,
            map< pair< int, int >, double > edgeDuals ) [pure virtual]
```

4.2.2.3 SetMaxMemory()

```
void BasePricing::SetMaxMemory (
            int max_memory ) [inline]
```

4.2.2.4 SetMaxTime()

```
void BasePricing::SetMaxTime (
            double max_time ) [inline]
```

4.2.3 Member Data Documentation

4.2.3.1 max_memory

```
int BasePricing::max_memory [protected]
```

4.2.3.2 max_time

```
double BasePricing::max_time [protected]
```

4.2.3.3 params

```
Params* BasePricing::params [protected]
```

4.2.3.4 problemData

```
ProblemData* BasePricing::problemData
```

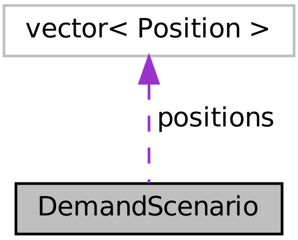
The documentation for this class was generated from the following file:

- include/BasePricing.h

4.3 DemandScenario Class Reference

```
#include <StochasticInfo.h>
```

Collaboration diagram for DemandScenario:



Private Attributes

- vector< Position > positions
- std::vector< Request > requests
- std::vector< Destination > destinations

4.3.1 Member Data Documentation

4.3.1.1 destinations

```
std::vector<Destination> DemandScenario::destinations [private]
```

4.3.1.2 positions

```
vector<Position> DemandScenario::positions [private]
```

4.3.1.3 requests

```
std::vector<Request> DemandScenario::requests [private]
```

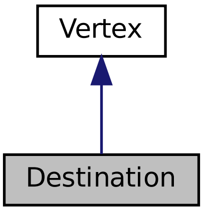
The documentation for this class was generated from the following file:

- include/StochasticInfo.h

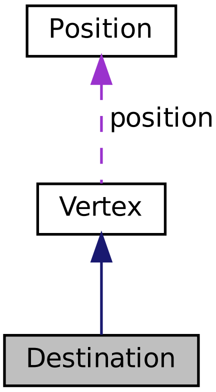
4.4 Destination Struct Reference

```
#include <ProblemData.h>
```

Inheritance diagram for Destination:



Collaboration diagram for Destination:



Public Attributes

- bool projected

Additional Inherited Members

4.4.1 Member Data Documentation

4.4.1.1 projected

```
bool Destination::projected
```

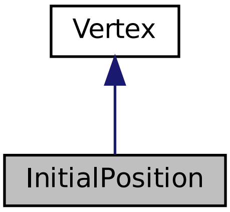
The documentation for this struct was generated from the following file:

- include/ProblemData.h

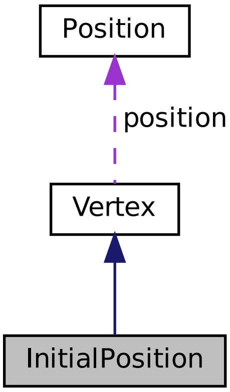
4.5 InitialPosition Struct Reference

```
#include <ProblemData.h>
```

Inheritance diagram for InitialPosition:



Collaboration diagram for InitialPosition:



Additional Inherited Members

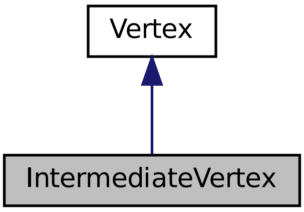
The documentation for this struct was generated from the following file:

- include/ProblemData.h

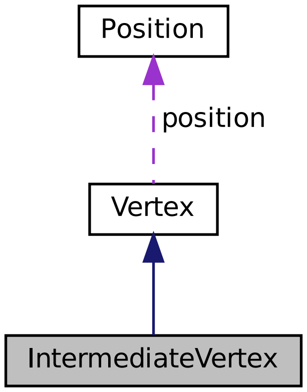
4.6 IntermediateVertex Struct Reference

```
#include <ProblemData.h>
```

Inheritance diagram for IntermediateVertex:



Collaboration diagram for IntermediateVertex:



Public Attributes

- int ws_id

Additional Inherited Members

4.6.1 Member Data Documentation

4.6.1.1 ws_id

```
int IntermediateVertex::ws_id
```

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

- include/ProblemData.h

4.7 OSRMHelper Class Reference

```
#include <OSRMHelper.h>
```

Public Member Functions

- OSRMHelper (std::string osmPath)
- std::vector< std::vector< double > > TableRequest (std::vector< const Vertex * > &vertices)
- std::vector< std::vector< double > > TableRequest (std::vector< double > &longitudes, std::vector< double > &latitudes)
- double GetDistance (double lon1, double lat1, double lon2, double lat2)
- double GetDuration (double lon1, double lat1, double lon2, double lat2)

4.7.1 Detailed Description

Helper class to facilitate interaction with the OSRM-backend API. Requires access to a local OSM database pre-processed with OSRM with the Multi-Level Dijkstra (MLD) configuration (requires extract+partition+customize).See `https://github.com/Project-OSRM/osrm-backend` for more information

4.7.2 Constructor & Destructor Documentation

4.7.2.1 OSRMHelper()

```
OSRMHelper::OSRMHelper (
            std::string osmPath )  [inline]
```

4.7.3 Member Function Documentation

4.7.3.1 GetDistance()

```
double OSRMHelper::GetDistance (
            double lon1,
            double lat1,
            double lon2,
            double lat2 )  [inline]
```

4.7.3.2 GetDuration()

```
double OSRMHelper::GetDuration (
            double lon1,
            double lat1,
            double lon2,
            double lat2 )  [inline]
```

4.7.3.3 TableRequest() [1/2]

```
std::vector<std::vector<double> > OSRMHelper::TableRequest (
            std::vector< const Vertex * > & vertices )  [inline]
```

4.7.3.4 TableRequest() [2/2]

```
std::vector<std::vector<double> > OSRMHelper::TableRequest (
            std::vector< double > & longitudes,
            std::vector< double > & latitudes )  [inline]
```

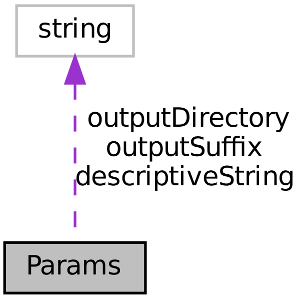
The documentation for this class was generated from the following file:

- include/OSRMHelper.h

4.8 Params Class Reference

```
#include <Params.h>
```

Collaboration diagram for Params:



Public Member Functions

- Params ()
- Params (string pathToInstance, string instanceType="pdptw")
- void StartTime ()
- double GetElapsedTime ()
- bool Timeout ()
- bool AllowsPositiveRCElimination (WaitingStationPolicy wsp)

Public Attributes

- string descriptiveString
- string outputDirectory
- string outputSuffix
- bool solveRelaxedProblem = false
- int heuristic_run = 0
- bool outputDuals
- int newRoutesPerPricing
- double initialDSF
- double DSFDecrement
- PricingAlgorithm pricingAlgorithm
- double max_time
- double max_memory
- double maxTimeSinglePricing
- double maxMemorySinglePricing
- int nbRandomInitialRoutes
- int route_gen_seed
- bool useBranchingOnVehicles
- bool useBranchingOnEdges
- bool alwaysLoopVehicles
- int maxSolverIterations
- int maxNbRoutes
- double RCEpsilon = 1.0
- bool timeout

Private Attributes

- std::default_random_engine generator
- int seed
- std::clock_t alg_start

4.8.1 Constructor & Destructor Documentation

4.8.1.1 Params() [1/2]

```
Params::Params ( )  [inline]
```

4.8.1.2 Params() [2/2]

```
Params::Params (
            string pathToInstance,
            string instanceType = "pdptw" )  [inline]
```

4.8.2 Member Function Documentation

4.8.2.1 AllowsPositiveRCElimination()

```
bool Params::AllowsPositiveRCElimination (
             WaitingStationPolicy wsp )  [inline]
```

4.8.2.2 GetElapsedTime()

```
double Params::GetElapsedTime ( )  [inline]
```

4.8.2.3 StartTime()

```
void Params::StartTime ( )  [inline]
```

4.8.2.4 Timeout()

```
bool Params::Timeout ( )  [inline]
```

4.8.3 Member Data Documentation

4.8.3.1 alg_start

```
std::clock_t Params::alg_start  [private]
```

4.8.3.2 alwaysLoopVehicles

```
bool Params::alwaysLoopVehicles
```

4.8.3.3 descriptiveString

```
string Params::descriptiveString
```

4.8.3.4 DSFDecrement

```
double Params::DSFDecrement
```

4.8.3.5 generator

```
std::default_random_engine Params::generator  [private]
```

4.8.3.6 heuristic_run

```
int Params::heuristic_run = 0
```

4.8.3.7 initialDSF

```
double Params::initialDSF
```

4.8.3.8 max_memory

```
double Params::max_memory
```

4.8.3.9 max_time

```
double Params::max_time
```

4.8.3.10 maxMemorySinglePricing

```
double Params::maxMemorySinglePricing
```

4.8.3.11 maxNbRoutes

```
int Params::maxNbRoutes
```

4.8.3.12 maxSolverIterations

```
int Params::maxSolverIterations
```

4.8.3.13 maxTimeSinglePricing

```
double Params::maxTimeSinglePricing
```

4.8.3.14 nbRandomInitialRoutes

```
int Params::nbRandomInitialRoutes
```

4.8.3.15 newRoutesPerPricing

```
int Params::newRoutesPerPricing
```

4.8.3.16 outputDirectory

```
string Params::outputDirectory
```

4.8.3.17 outputDuals

```
bool Params::outputDuals
```

4.8.3.18 outputSuffix

```
string Params::outputSuffix
```

4.8.3.19 pricingAlgorithm

```
PricingAlgorithm Params::pricingAlgorithm
```

4.8.3.20 RCEpsilon

```
double Params::RCEpsilon = 1.0
```

4.8.3.21 route_gen_seed

```
int Params::route_gen_seed
```

4.8.3.22 seed

```
int Params::seed  [private]
```

4.8.3.23 solveRelaxedProblem

```
bool Params::solveRelaxedProblem = false
```

4.8.3.24 timeout

```
bool Params::timeout
```

4.8.3.25 useBranchingOnEdges

```
bool Params::useBranchingOnEdges
```

4.8.3.26 useBranchingOnVehicles

```
bool Params::useBranchingOnVehicles
```

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

- include/Params.h

4.9 Position Struct Reference

```
#include <ProblemData.h>
```

Public Attributes

- double x
- double y

4.9.1 Member Data Documentation

4.9.1.1 x

```
double Position::x
```

4.9.1.2 y

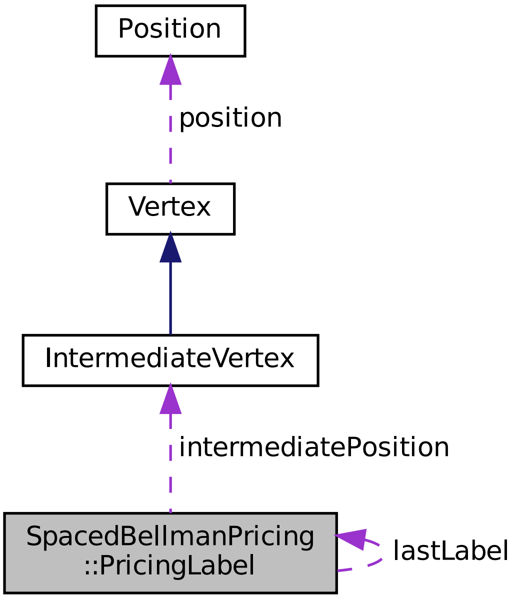
```
double Position::y
```

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

- include/ProblemData.h

4.10 SpacedBellmanPricing::PricingLabel Class Reference

Collaboration diagram for SpacedBellmanPricing::PricingLabel:



Public Attributes

- int reqId
- double reducedCost
- double time
- int lastWaitingStation
- IntermediateVertex ∗ intermediatePosition
- const PricingLabel ∗ lastLabel
- bool referenced
- bool alreadyExpanded

4.10.1 Member Data Documentation

4.10.1.1 alreadyExpanded

```
bool SpacedBellmanPricing::PricingLabel::alreadyExpanded  [mutable]
```

4.10.1.2 intermediatePosition

```
IntermediateVertex* SpacedBellmanPricing::PricingLabel::intermediatePosition
```

4.10.1.3 lastLabel

```
const PricingLabel* SpacedBellmanPricing::PricingLabel::lastLabel
```

4.10.1.4 lastWaitingStation

```
int SpacedBellmanPricing::PricingLabel::lastWaitingStation
```

4.10.1.5 reducedCost

```
double SpacedBellmanPricing::PricingLabel::reducedCost
```

4.10.1.6 referenced

```
bool SpacedBellmanPricing::PricingLabel::referenced  [mutable]
```

4.10.1.7 reqId

```
int SpacedBellmanPricing::PricingLabel::reqId
```

4.10.1.8 time

```
double SpacedBellmanPricing::PricingLabel::time
```

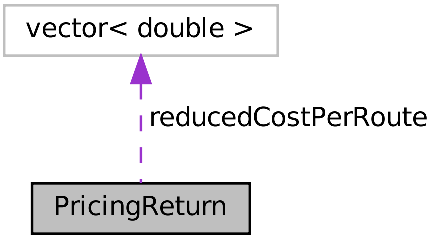
The documentation for this class was generated from the following file:

- include/SpacedBellmanPricing.h

4.11 PricingReturn Struct Reference

```
#include <BasePricing.h>
```

Collaboration diagram for PricingReturn:



Public Attributes

- PricingReturnStatus status
- size_t labelsPriced
- size_t labelsStored
- size_t maxLabelsStoredSimultaneously
- size_t labelsDeleted
- size_t mostLabelsInRequest
- size_t nbConsideredRequests
- bool timeout
- vector< double > reducedCostPerRoute

4.11.1 Member Data Documentation

4.11.1.1 labelsDeleted

```
size_t PricingReturn::labelsDeleted
```

4.11.1.2 labelsPriced

```
size_t PricingReturn::labelsPriced
```

4.11.1.3 labelsStored

```
size_t PricingReturn::labelsStored
```

4.11.1.4 maxLabelsStoredSimultaneously

```
size_t PricingReturn::maxLabelsStoredSimultaneously
```

4.11.1.5 mostLabelsInRequest

```
size_t PricingReturn::mostLabelsInRequest
```

4.11.1.6 nbConsideredRequests

```
size_t PricingReturn::nbConsideredRequests
```

4.11.1.7 reducedCostPerRoute

```
vector<double> PricingReturn::reducedCostPerRoute
```

4.11.1.8 status

```
PricingReturnStatus PricingReturn::status
```

4.11.1.9 timeout

```
bool PricingReturn::timeout
```

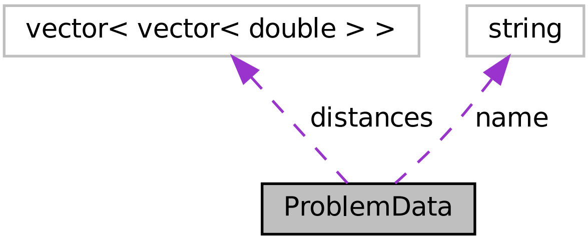
The documentation for this struct was generated from the following file:

- include/BasePricing.h

4.12 ProblemData Class Reference

```
#include <ProblemData.h>
```

Collaboration diagram for ProblemData:



Public Member Functions

- int NbVertices () const
- int NbRequests () const
- int NbVehicles () const
- int NbWaitingStations () const
- int GetClosestDestination (int vertexId) const
- double calculateDistance (const Position &pos1, const Position &pos2)
- void exampleInstance ()
- void readPDPTWInstance (string path)
- void readSDVRPTWInstance (string path)
- const InitialPosition ∗ GetInitialPosition (int id) const
- const InitialPosition ∗ GetInitialPositionByIndex (int index) const
- const Request ∗ GetRequest (int id) const
- const Request ∗ GetRequestByIndex (int index) const
- const Destination ∗ GetDestination (int id) const
- const Destination ∗ GetDestinationByIndex (int index) const
- const WaitingStation ∗ GetWaitingStation (int id) const
- const WaitingStation ∗ GetWaitingStationByIndex (int id) const
- const Vertex ∗ GetVertex (int id) const
- int RequestIdToIndex (int id) const
- int IndexToRequestId (int index) const
- void ResetNumberOfVehicles (int newNbVehicles)
- bool IsRequest (int id) const
- bool IsDestination (int id) const
- bool IsInitialPosition (int id) const
- bool IsWaitingStation (int id) const
- bool IsIntermediateVertex (int id) const
- bool IsCompatible (const Request ∗request, int vehicle_index)
- double OmmitedDistance (int i, int j)
- double Distance (int i, int j)
- const Vehicle ∗ getVehicle (int id)

- double weighted_lateness (int req_id, double time)
- double weighted_lateness (const Request ∗request, double time)
- void SetVehicleAvailability (vector< double > vehicleAvailability)
- void SetVehiclePositions (vector< Position > vehiclePositions)
- ProblemData ()
- ProblemData (string pathToInstance, string instanceType="pdptw")
- ∼ProblemData ()

Static Public Member Functions

- static double calculateDistance (const Position &pos1, const Position &pos2, DistanceType distanceType)
- static double euclidianDistance (const Position &pos1, const Position &pos2)
- static double geodesicDistance (const Position &pos1, const Position &pos2)
- static Position GetIntermediatePosition2 (Position p1, Position p2, double timeFromStart)
- static double VehicleSpeed ()
- static bool readVincentInstance (string requests_path, string hospitals_path, string waiting_stations_path, string cleaning_stations_path, int instance_index, ProblemData &outInstance, bool tenColumns, std::string osmPath, bool useTimeHorizon=false, double timeHorizon=0.0, int overwriteNbVehicles=-1)
- static Position GetIntermediatePosition (Position source, Position destination, double t0, double t)

Public Attributes

- string name
- double timeHorizon
- bool computeTimeHorizon = false
- bool allowRerouting = false
- WaitingStationPolicy waitingStationPolicy = WaitingStationPolicy::optionalStopInClosestWaitingStation
- bool useTargetWaitTimeObjective = false
- std::map< double, double > target_times_per_weight = {{1, 30∗60}, {2, 15∗60}, {4, 10∗60}}
- DistanceType distanceType

Private Member Functions

- int InitialPositionIdToIndex (int id) const
- int InitialPositionIndexToId (int index) const
- int DestinationIdToIndex (int id) const
- int IndexToDestinationId (int index) const
- int WaitingStationIdToIndex (int id) const
- int IndexToWaitingStationId (int index) const
- void PrecomputeClosestWSs ()
- int GetClosestWaitingStation (int vertexId) const
- void Validate ()

Static Private Member Functions

- static void CalculateDistanceMatrix (vector< Vertex ∗ > &vertices, vector< vector< double >> &outMatrix, DistanceType distanceType)

Private Attributes

- std::vector< Vehicle > vehicles
- std::vector< InitialPosition > initialPositions
- std::vector< Request > requests
- std::vector< Destination > destinations
- std::vector< WaitingStation > waitingStations
- vector< vector< double > > distances

4.12.1 Constructor & Destructor Documentation

4.12.1.1 ProblemData() [1/2]

```
ProblemData::ProblemData ( )
```

4.12.1.2 ProblemData() [2/2]

```
ProblemData::ProblemData (
            string pathToInstance,
            string instanceType = "pdptw" )
```

4.12.1.3 ∼ProblemData()

```
ProblemData::∼ProblemData ( )  [inline]
```

4.12.2 Member Function Documentation

4.12.2.1 calculateDistance() [1/2]

```
double ProblemData::calculateDistance (
            const Position & pos1,
            const Position & pos2 )  [inline]
```

4.12.2.2 calculateDistance() [2/2]

```
double ProblemData::calculateDistance (
             const Position & pos1,
             const Position & pos2,
             DistanceType distanceType )  [static]
```

4.12.2.3 CalculateDistanceMatrix()

```
void ProblemData::CalculateDistanceMatrix (
             vector< Vertex * > & vertices,
             vector< vector< double >> & outMatrix,
             DistanceType distanceType )  [static], [private]
```

4.12.2.4 DestinationIdToIndex()

```
int ProblemData::DestinationIdToIndex (
             int id ) const  [private]
```

4.12.2.5 Distance()

```
double ProblemData::Distance (
             int i,
             int j )
```

4.12.2.6 euclidianDistance()

```
double ProblemData::euclidianDistance (
             const Position & pos1,
             const Position & pos2 )  [static]
```

4.12.2.7 exampleInstance()

```
void ProblemData::exampleInstance ( )
```

4.12.2.8 geodesicDistance()

```
double ProblemData::geodesicDistance (
            const Position & pos1,
            const Position & pos2 )  [static]
```

4.12.2.9 GetClosestDestination()

```
int ProblemData::GetClosestDestination (
            int vertexId ) const
```

4.12.2.10 GetClosestWaitingStation()

```
int ProblemData::GetClosestWaitingStation (
            int vertexId ) const  [private]
```

4.12.2.11 GetDestination()

```
const Destination * ProblemData::GetDestination (
            int id ) const
```

4.12.2.12 GetDestinationByIndex()

```
const Destination * ProblemData::GetDestinationByIndex (
            int index ) const
```

4.12.2.13 GetInitialPosition()

```
const InitialPosition * ProblemData::GetInitialPosition (
            int id ) const
```

4.12.2.14 GetInitialPositionByIndex()

```
const InitialPosition * ProblemData::GetInitialPositionByIndex (
            int index ) const
```

4.12.2.15 GetIntermediatePosition()

```
Position ProblemData::GetIntermediatePosition (
            Position source,
            Position destination,
            double t0,
            double t ) [static]
```

4.12.2.16 GetIntermediatePosition2()

```
Position ProblemData::GetIntermediatePosition2 (
            Position p1,
            Position p2,
            double timeFromStart ) [static]
```

4.12.2.17 GetRequest()

```
const Request * ProblemData::GetRequest (
            int id ) const
```

4.12.2.18 GetRequestByIndex()

```
const Request * ProblemData::GetRequestByIndex (
            int index ) const
```

4.12.2.19 getVehicle()

```
const Vehicle* ProblemData::getVehicle (
            int id ) [inline]
```

4.12.2.20 GetVertex()

```
const Vertex * ProblemData::GetVertex (
            int id ) const
```

4.12.2.21 GetWaitingStation()

```
const WaitingStation * ProblemData::GetWaitingStation (
             int id ) const
```

4.12.2.22 GetWaitingStationByIndex()

```
const WaitingStation * ProblemData::GetWaitingStationByIndex (
             int id ) const
```

4.12.2.23 IndexToDestinationId()

```
int ProblemData::IndexToDestinationId (
             int index ) const  [private]
```

4.12.2.24 IndexToRequestId()

```
int ProblemData::IndexToRequestId (
             int index ) const
```

4.12.2.25 IndexToWaitingStationId()

```
int ProblemData::IndexToWaitingStationId (
             int index ) const  [private]
```

4.12.2.26 InitialPositionIdToIndex()

```
int ProblemData::InitialPositionIdToIndex (
             int id ) const  [private]
```

4.12.2.27 InitialPositionIndexToId()

```
int ProblemData::InitialPositionIndexToId (
             int index ) const  [private]
```

4.12.2.28 IsCompatible()

```
bool ProblemData::IsCompatible (
             const Request * request,
             int vehicle_index )
```

4.12.2.29 IsDestination()

```
bool ProblemData::IsDestination (
             int id ) const
```

4.12.2.30 IsInitialPosition()

```
bool ProblemData::IsInitialPosition (
             int id ) const
```

4.12.2.31 IsIntermediateVertex()

```
bool ProblemData::IsIntermediateVertex (
             int id ) const
```

4.12.2.32 IsRequest()

```
bool ProblemData::IsRequest (
             int id ) const
```

4.12.2.33 IsWaitingStation()

```
bool ProblemData::IsWaitingStation (
             int id ) const
```

4.12.2.34 NbRequests()

```
int ProblemData::NbRequests ( ) const  [inline]
```

4.12.2.35 NbVehicles()

```
int ProblemData::NbVehicles ( ) const  [inline]
```

4.12.2.36 NbVertices()

```
int ProblemData::NbVertices ( ) const  [inline]
```

4.12.2.37 NbWaitingStations()

```
int ProblemData::NbWaitingStations ( ) const  [inline]
```

4.12.2.38 OmmitedDistance()

```
double ProblemData::OmmitedDistance (
             int i,
             int j )
```

4.12.2.39 PrecomputeClosestWSs()

```
void ProblemData::PrecomputeClosestWSs ( )  [private]
```

4.12.2.40 readPDPTWInstance()

```
void ProblemData::readPDPTWInstance (
             string path )
```

4.12.2.41 readSDVRPTWInstance()

```
void ProblemData::readSDVRPTWInstance (
             string path )
```

4.12.2.42 readVincentInstance()

```
bool ProblemData::readVincentInstance (
             string requests_path,
             string hospitals_path,
             string waiting_stations_path,
             string cleaning_stations_path,
             int instance_index,
             ProblemData & outInstance,
             bool tenColumns,
             std::string osmPath,
             bool useTimeHorizon = false,
             double timeHorizon = 0.0,
             int overwriteNbVehicles = -1 )  [static]
```

4.12.2.43 RequestIdToIndex()

```
int ProblemData::RequestIdToIndex (
             int id ) const
```

4.12.2.44 ResetNumberOfVehicles()

```
void ProblemData::ResetNumberOfVehicles (
             int newNbVehicles )
```

4.12.2.45 SetVehicleAvailability()

```
void ProblemData::SetVehicleAvailability (
             vector< double > vehicleAvailability )
```

4.12.2.46 SetVehiclePositions()

```
void ProblemData::SetVehiclePositions (
             vector< Position > vehiclePositions )
```

4.12.2.47 Validate()

```
void ProblemData::Validate ( )  [private]
```

4.12.2.48 VehicleSpeed()

```
static double ProblemData::VehicleSpeed ( )  [inline], [static]
```

4.12.2.49 WaitingStationIdToIndex()

```
int ProblemData::WaitingStationIdToIndex (
             int id ) const  [private]
```

4.12.2.50 weighted_lateness() [1/2]

```
double ProblemData::weighted_lateness (
             const Request * request,
             double time )
```

4.12.2.51 weighted_lateness() [2/2]

```
double ProblemData::weighted_lateness (
             int req_id,
             double time )
```

4.12.3 Member Data Documentation

4.12.3.1 allowRerouting

```
bool ProblemData::allowRerouting = false
```

4.12.3.2 computeTimeHorizon

```
bool ProblemData::computeTimeHorizon = false
```

4.12.3.3 destinations

```
std::vector<Destination> ProblemData::destinations  [private]
```

4.12.3.4 distances

```
vector<vector<double> > ProblemData::distances  [private]
```

4.12.3.5 distanceType

```
DistanceType ProblemData::distanceType
```

4.12.3.6 initialPositions

```
std::vector<InitialPosition> ProblemData::initialPositions  [private]
```

4.12.3.7 name

```
string ProblemData::name
```

4.12.3.8 requests

```
std::vector<Request> ProblemData::requests  [private]
```

4.12.3.9 target_times_per_weight

```
std::map<double, double> ProblemData::target_times_per_weight = {{1, 30*60}, {2, 15*60}, {4,
10*60}}
```

4.12.3.10 timeHorizon

```
double ProblemData::timeHorizon
```

4.12.3.11 useTargetWaitTimeObjective

```
bool ProblemData::useTargetWaitTimeObjective = false
```

4.12.3.12 vehicles

```
std::vector<Vehicle> ProblemData::vehicles [private]
```

4.12.3.13 waitingStationPolicy

```
WaitingStationPolicy ProblemData::waitingStationPolicy = WaitingStationPolicy::optionalStopInClosestWaitingSta
```

4.12.3.14 waitingStations

```
std::vector<WaitingStation> ProblemData::waitingStations [private]
```

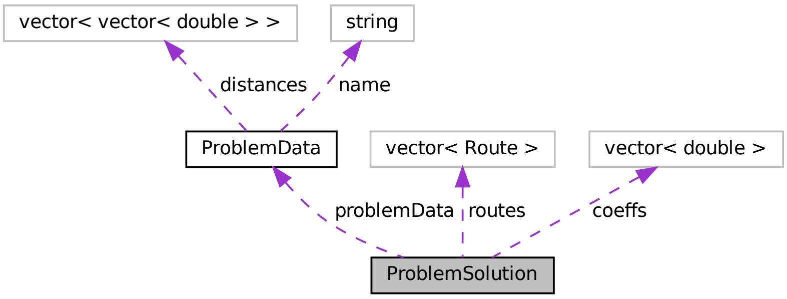
The documentation for this class was generated from the following files:

- include/ProblemData.h
- src/ProblemData.cpp

4.13 ProblemSolution Class Reference

```
#include <ProblemSolution.h>
```

Collaboration diagram for ProblemSolution:



Public Member Functions

- ProblemSolution (ProblemData ∗data)
- void AnySolution ()
- void SetToInitialSolution (ProblemData ∗problemData, Params ∗params, bool outputToFile=false, int heuristic=2)
- ResponseSummary GetResponseSummary ()
- void UpdateCost ()
- void PrintSolution ()
- void WriteSolution (std::string path)
- void WriteRequestsOutput (std::string path)

Static Public Member Functions

- static vector< Route > ClosestAvailableVehicleSolution (ProblemData ∗problemData, Params ∗params)
- static vector< Route > FastestArrivingVehicleSolution (ProblemData ∗problemData, Params ∗params)
- static vector< Route > AllOneSizedRoutesSolution (ProblemData ∗problemData, Params ∗params)
- static void RandomRoutes (ProblemData ∗problemData, Params ∗params, int n_routes, uint32_t seed, vector< Route > &out_routes)

Public Attributes

- ProblemData ∗ problemData
- vector< Route > routes
- vector< double > coeffs
- bool relaxed
- double cost
- double route_cost
- double penalty_cost

4.13.1 Constructor & Destructor Documentation

4.13.1.1 ProblemSolution()

```
ProblemSolution::ProblemSolution (
            ProblemData ∗ data )
```

4.13.2 Member Function Documentation

4.13.2.1 AllOneSizedRoutesSolution()

```
vector< Route > ProblemSolution::AllOneSizedRoutesSolution (
            ProblemData ∗ problemData,
            Params ∗ params )  [static]
```

4.13.2.2 AnySolution()

```
void ProblemSolution::AnySolution ( )
```

4.13.2.3 ClosestAvailableVehicleSolution()

```
vector< Route > ProblemSolution::ClosestAvailableVehicleSolution (
             ProblemData * problemData,
             Params * params )  [static]
```

4.13.2.4 FastestArrivingVehicleSolution()

```
vector< Route > ProblemSolution::FastestArrivingVehicleSolution (
             ProblemData * problemData,
             Params * params )  [static]
```

4.13.2.5 GetResponseSummary()

```
ResponseSummary ProblemSolution::GetResponseSummary ( )
```

4.13.2.6 PrintSolution()

```
void ProblemSolution::PrintSolution ( )
```

4.13.2.7 RandomRoutes()

```
void ProblemSolution::RandomRoutes (
             ProblemData * problemData,
             Params * params,
             int n_routes,
             uint32_t seed,
             vector< Route > & out_routes )  [static]
```

4.13.2.8 SetToInitialSolution()

```
void ProblemSolution::SetToInitialSolution (
            ProblemData * problemData,
            Params * params,
            bool outputToFile = false,
            int heuristic = 2 )
```

4.13.2.9 UpdateCost()

```
void ProblemSolution::UpdateCost ( )
```

4.13.2.10 WriteRequestsOutput()

```
void ProblemSolution::WriteRequestsOutput (
            std::string path )
```

4.13.2.11 WriteSolution()

```
void ProblemSolution::WriteSolution (
            std::string path )
```

4.13.3 Member Data Documentation

4.13.3.1 coeffs

```
vector<double> ProblemSolution::coeffs
```

4.13.3.2 cost

```
double ProblemSolution::cost
```

4.13.3.3 penalty_cost

```
double ProblemSolution::penalty_cost
```

4.13.3.4 problemData

```
ProblemData* ProblemSolution::problemData
```

4.13.3.5 relaxed

```
bool ProblemSolution::relaxed
```

4.13.3.6 route_cost

```
double ProblemSolution::route_cost
```

4.13.3.7 routes

```
vector<Route> ProblemSolution::routes
```

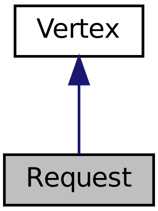
The documentation for this class was generated from the following files:

- include/ProblemSolution.h
- src/ProblemSolution.cpp

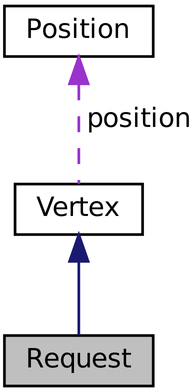
4.14 Request Struct Reference

```
#include <ProblemData.h>
```

Inheritance diagram for Request:



Collaboration diagram for Request:



Public Attributes

- int destination
- int type
- double weight
- double non_service_penalty
- double arrival_time
- double service_time
- bool projected

Additional Inherited Members

4.14.1 Member Data Documentation

4.14.1.1 arrival_time

```
double Request::arrival_time
```

4.14.1.2 destination

```
int Request::destination
```

4.14.1.3 non_service_penalty

```
double Request::non_service_penalty
```

4.14.1.4 projected

```
bool Request::projected
```

4.14.1.5 service_time

```
double Request::service_time
```

4.14.1.6 type

```
int Request::type
```

4.14.1.7 weight

```
double Request::weight
```

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

- include/ProblemData.h

4.15 ResponseSummary Struct Reference

```
#include <ProblemSolution.h>
```

Public Attributes

- double meanResponseTime
- double maxResponseTime
- double meanWeightedResponseTime
- double maxWeightedResponseTime
- int nServiced
- int nNotServiced
- double meanNonServicePenalty
- double maxNonServicePenalty

4.15.1 Member Data Documentation

4.15.1.1 maxNonServicePenalty

```
double ResponseSummary::maxNonServicePenalty
```

4.15.1.2 maxResponseTime

```
double ResponseSummary::maxResponseTime
```

4.15.1.3 maxWeightedResponseTime

```
double ResponseSummary::maxWeightedResponseTime
```

4.15.1.4 meanNonServicePenalty

```
double ResponseSummary::meanNonServicePenalty
```

4.15.1.5 meanResponseTime

```
double ResponseSummary::meanResponseTime
```

4.15.1.6 meanWeightedResponseTime

```
double ResponseSummary::meanWeightedResponseTime
```

4.15.1.7 nNotServiced

```
int ResponseSummary::nNotServiced
```

4.15.1.8 nServiced

```
int ResponseSummary::nServiced
```

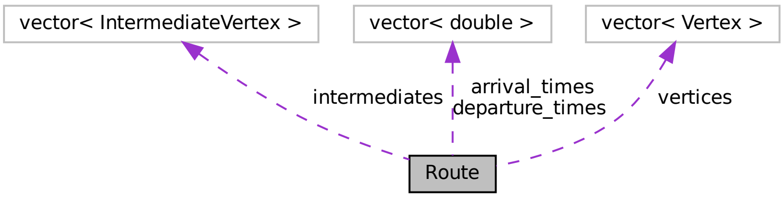
The documentation for this struct was generated from the following file:

- include/ProblemSolution.h

4.16 Route Class Reference

```
#include <ProblemSolution.h>
```

Collaboration diagram for Route:



Public Member Functions

- void UpdateCost (ProblemData ∗problemData)
- void SetArrivalsAndDepartures (ProblemData ∗problemData)
- Route ()=default
- Route (const Route &)=default
- bool operator== (const Route &otherRoute) const
- size_t GetHash (ProblemData ∗problemData)
- int GetRequestCount (ProblemData ∗problemData, int req_id)
- std::map< pair< int, int >, int > GetEdgeUsage (ProblemData ∗problemData)

Public Attributes

- int veh_index
- vector< IntermediateVertex > intermediates
- vector< Vertex > vertices
- vector< double > arrival_times
- vector< double > departure_times
- bool has_cycles
- double total_lateness
- double end_time

4.16.1 Constructor & Destructor Documentation

4.16.1.1 Route() [1/2]

```
Route::Route ( )  [default]
```

4.16.1.2 Route() [2/2]

```
Route::Route (
             const Route &  )  [default]
```

4.16.2 Member Function Documentation

4.16.2.1 GetEdgeUsage()

```
std::map< pair< int, int >, int > Route::GetEdgeUsage (
             ProblemData ∗ problemData )
```

4.16.2.2 GetHash()

```
size_t Route::GetHash (
            ProblemData * problemData )  [inline]
```

4.16.2.3 GetRequestCount()

```
int Route::GetRequestCount (
            ProblemData * problemData,
            int req_id )  [inline]
```

4.16.2.4 operator==()

```
bool Route::operator== (
            const Route & otherRoute ) const  [inline]
```

4.16.2.5 SetArrivalsAndDepartures()

```
void Route::SetArrivalsAndDepartures (
            ProblemData * problemData )
```

4.16.2.6 UpdateCost()

```
void Route::UpdateCost (
            ProblemData * problemData )
```

4.16.3 Member Data Documentation

4.16.3.1 arrival_times

```
vector<double> Route::arrival_times
```

4.16.3.2 departure_times

```
vector<double> Route::departure_times
```

4.16.3.3 end_time

```
double Route::end_time
```

4.16.3.4 has_cycles

```
bool Route::has_cycles
```

4.16.3.5 intermediates

```
vector<IntermediateVertex> Route::intermediates
```

4.16.3.6 total_lateness

```
double Route::total_lateness
```

4.16.3.7 veh_index

```
int Route::veh_index
```

4.16.3.8 vertices

```
vector<Vertex> Route::vertices
```

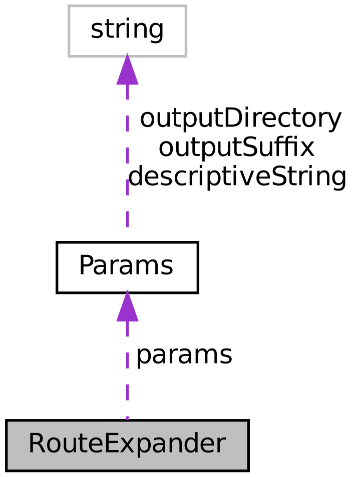
The documentation for this class was generated from the following files:

- include/ProblemSolution.h
- src/ProblemSolution.cpp

4.17 RouteExpander Class Reference

```
#include <RouteExpander.h>
```

Collaboration diagram for RouteExpander:



Public Member Functions

- RouteExpander (Params ∗params)
- bool checkRouteExpansion (ProblemData ∗problemData, int vehicle_id, const Request ∗nextRequest, const Vertex ∗lastVertex, double lastVertexArrivalTime, double &outTime, int &outWaitingStationId, bool &outUseIntermediateIntermediateVertex, IntermediateVertex &outIntermediateVertex)
- bool checkRouteExpansion (ProblemData ∗problemData, int vehicle_id, const Request ∗nextRequest, const Vertex ∗lastVertex, double lastVertexArrivalTime, double &outTime, int &outWaitingStationId)

Private Attributes

- Params ∗ params

4.17.1 Constructor & Destructor Documentation

4.17.1.1 RouteExpander()

```
RouteExpander::RouteExpander (
            Params ∗ params )  [inline], [explicit]
```

4.17.2 Member Function Documentation

4.17.2.1 checkRouteExpansion() [1/2]

```
bool RouteExpander::checkRouteExpansion (
             ProblemData * problemData,
             int vehicle_id,
             const Request * nextRequest,
             const Vertex * lastVertex,
             double lastVertexArrivalTime,
             double & outTime,
             int & outWaitingStationId )  [inline]
```

4.17.2.2 checkRouteExpansion() [2/2]

```
bool RouteExpander::checkRouteExpansion (
             ProblemData * problemData,
             int vehicle_id,
             const Request * nextRequest,
             const Vertex * lastVertex,
             double lastVertexArrivalTime,
             double & outTime,
             int & outWaitingStationId,
             bool & outUseIntermediateIntermediateVertex,
             IntermediateVertex & outIntermediateVertex )
```

4.17.3 Member Data Documentation

4.17.3.1 params

```
Params* RouteExpander::params  [private]
```

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

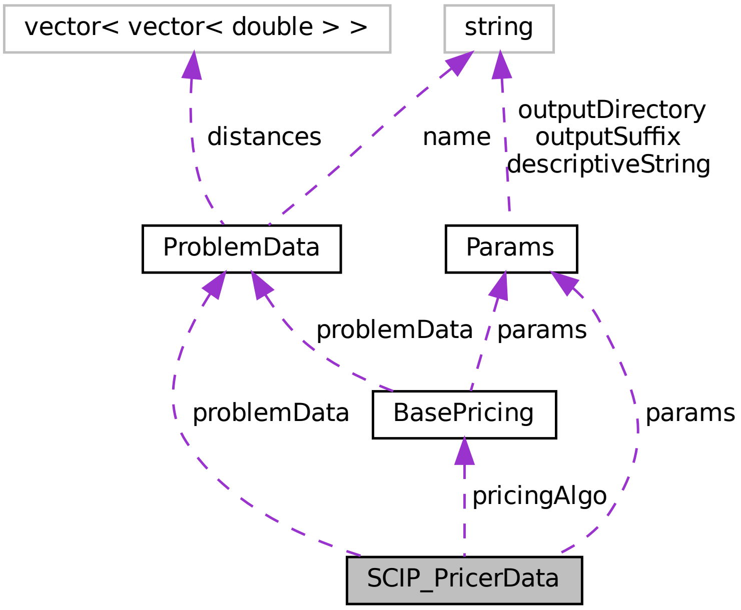
- include/RouteExpander.h
- src/RouteExpander.cpp

4.18 SCIP_PricerData Struct Reference

Variable pricer data used in the pricer.

```
#include <pricer_SPwCG.h>
```

Collaboration diagram for SCIP_PricerData:



Public Attributes

- SCIP_CONSHDLR ∗ conshdlr
- SCIP_CONS ∗∗ conss
- Params ∗ params
- ProblemData ∗ problemData
- BasePricing ∗ pricingAlgo
- bool heuristicPricing
- int lastSuccessfullVehicle
- double total_pricing_time
- int total_pricing_calls
- int total_pricing_fails
- int total_pricing_timeouts
- size_t labelsPriced
- size_t labelsStored
- size_t labelsDeleted
- size_t sumOfMaxLabelsStoredSimultaneously
- size_t sumOfMostLabelsInRequest
- size_t sumOfNbConsideredRequests

4.18.1 Detailed Description

Variable pricer data used in the pricer.

4.18.2 Member Data Documentation

4.18.2.1 conshdlr

```
SCIP_CONSHDLR* SCIP_PricerData::conshdlr
```

comstraint handler for "same" and "diff" constraints

4.18.2.2 conss

```
SCIP_CONS** SCIP_PricerData::conss
```

set covering constraints for the items

4.18.2.3 heuristicPricing

```
bool SCIP_PricerData::heuristicPricing
```

$<$ implementation of pricing algorithm

4.18.2.4 labelsDeleted

```
size_t SCIP_PricerData::labelsDeleted
```

4.18.2.5 labelsPriced

```
size_t SCIP_PricerData::labelsPriced
```

4.18.2.6 labelsStored

```
size_t SCIP_PricerData::labelsStored
```

4.18.2.7 lastSuccessfullVehicle

```
int SCIP_PricerData::lastSuccessfullVehicle
```

4.18.2.8 params

```
Params* SCIP_PricerData::params
```

4.18.2.9 pricingAlgo

```
BasePricing* SCIP_PricerData::pricingAlgo
```

4.18.2.10 problemData

```
ProblemData* SCIP_PricerData::problemData
```

general problem info

4.18.2.11 sumOfMaxLabelsStoredSimultaneously

```
size_t SCIP_PricerData::sumOfMaxLabelsStoredSimultaneously
```

4.18.2.12 sumOfMostLabelsInRequest

```
size_t SCIP_PricerData::sumOfMostLabelsInRequest
```

4.18.2.13 sumOfNbConsideredRequests

```
size_t SCIP_PricerData::sumOfNbConsideredRequests
```

4.18.2.14 total_pricing_calls

```
int SCIP_PricerData::total_pricing_calls
```

4.18.2.15 total_pricing_fails

```
int SCIP_PricerData::total_pricing_fails
```

4.18.2.16 total_pricing_time

```
double SCIP_PricerData::total_pricing_time
```

4.18.2.17 total_pricing_timeouts

```
int SCIP_PricerData::total_pricing_timeouts
```

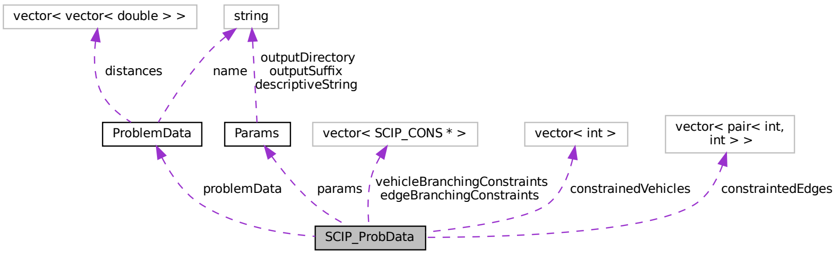
The documentation for this struct was generated from the following file:

- include/pricer_SPwCG.h

4.19 SCIP_ProbData Struct Reference

Problem data which is accessible in all places.

Collaboration diagram for SCIP_ProbData:



Public Attributes

- SCIP_VAR ∗∗ vars
- SCIP_CONS ∗∗ conss
- int nvars
- int varssize
- Params ∗ params
- ProblemData ∗ problemData
- vector< SCIP_CONS ∗ > ∗ vehicleBranchingConstraints
- vector< int > ∗ constrainedVehicles
- vector< SCIP_CONS ∗ > ∗ edgeBranchingConstraints
- vector< pair< int, int > > ∗ constraintedEdges
- std::multimap< size_t, SCIP_VAR ∗ > ∗ RouteToVarMap
- size_t timesBranchedWithRule [3]
- size_t timesRepeatedRouteWasPriced
- double repeatedRoutesTotalReducedCost

4.19.1 Detailed Description

Problem data which is accessible in all places.

This problem data is used to store the input of the binpacking, all variables which are created, and all constraints.

4.19.2 Member Data Documentation

4.19.2.1 conss

```
SCIP_CONS** SCIP_ProbData::conss
```

set partitioning constraints for each item exactly one

4.19.2.2 constrainedVehicles

```
vector<int>* SCIP_ProbData::constrainedVehicles
```

4.19.2.3 constraintedEdges

```
vector<pair<int, int> >* SCIP_ProbData::constraintedEdges
```

4.19.2.4 edgeBranchingConstraints

```
vector<SCIP_CONS*>* SCIP_ProbData::edgeBranchingConstraints
```

4.19.2.5 nvars

```
int SCIP_ProbData::nvars
```

number of generated variables

4.19.2.6 params

```
Params* SCIP_ProbData::params
```

4.19.2.7 problemData

```
ProblemData* SCIP_ProbData::problemData
```

4.19.2.8 repeatedRoutesTotalReducedCost

```
double SCIP_ProbData::repeatedRoutesTotalReducedCost
```

4.19.2.9 RouteToVarMap

```
std::multimap<size_t, SCIP_VAR*>* SCIP_ProbData::RouteToVarMap
```

4.19.2.10 timesBranchedWithRule

```
size_t SCIP_ProbData::timesBranchedWithRule[3]
```

4.19.2.11 timesRepeatedRouteWasPriced

```
size_t SCIP_ProbData::timesRepeatedRouteWasPriced
```

4.19.2.12 vars

```
SCIP_VAR** SCIP_ProbData::vars
```

all exiting variables in the problem

4.19.2.13 varssize

```
int SCIP_ProbData::varssize
```

size of the variable array

4.19.2.14 vehicleBranchingConstraints

```
vector<SCIP_CONS*>* SCIP_ProbData::vehicleBranchingConstraints
```

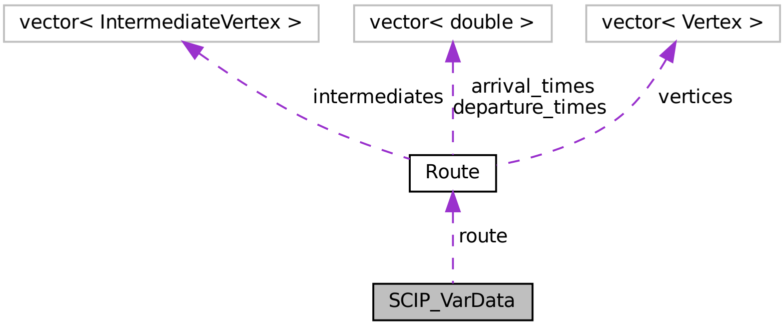
all data for this problem instance is acessible via this C++ Class

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

- src/probdata_SPwCG.cpp

4.20 SCIP_VarData Struct Reference

Collaboration diagram for SCIP_VarData:



Public Attributes

- int ∗ consids
- int ∗ conscoeffs
- int nconsids
- Route ∗ route

4.20.1 Detailed Description

Variable data which is attached to all variables.

This variable data is used to store in which constraints this variable appears. Therefore, the variable data contains the ids of constraints in which the variable is part of. Hence, that data give us a column view.

4.20.2 Member Data Documentation

4.20.2.1 conscoeffs

```
int* SCIP_VarData::conscoeffs
```

4.20.2.2 consids

```
int* SCIP_VarData::consids
```

4.20.2.3 nconsids

```
int SCIP_VarData::nconsids
```

4.20.2.4 route

```
Route* SCIP_VarData::route
```

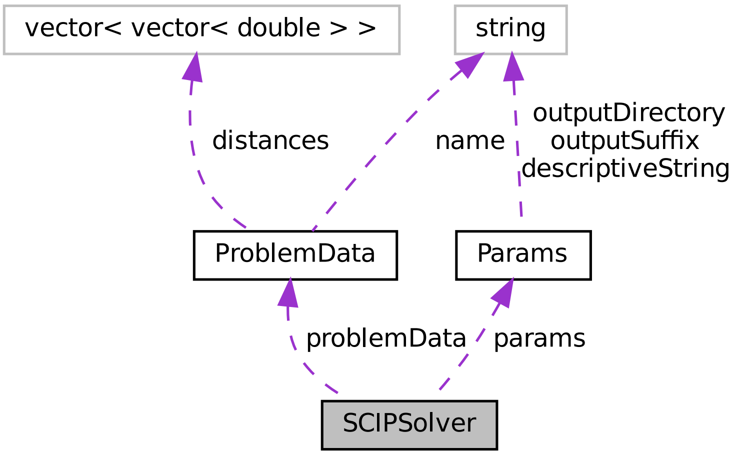
The documentation for this struct was generated from the following file:

- src/vardata_SPwCG.cpp

4.21 SCIPSolver Class Reference

```
#include <SCIPSolver.h>
```

Collaboration diagram for SCIPSolver:



Public Member Functions

- SCIPSolver (Params ∗params, ProblemData ∗problemData)
- void solve (ProblemSolution &solution)

Private Attributes

- Params ∗ params
- ProblemData ∗ problemData

4.21.1 Constructor & Destructor Documentation

4.21.1.1 SCIPSolver()

```
SCIPSolver::SCIPSolver (
            Params * params,
            ProblemData * problemData )  [inline]
```

4.21.2 Member Function Documentation

4.21.2.1 solve()

```
void SCIPSolver::solve (
            ProblemSolution & solution )
```

4.21.3 Member Data Documentation

4.21.3.1 params

```
Params* SCIPSolver::params  [private]
```

4.21.3.2 problemData

```
ProblemData* SCIPSolver::problemData  [private]
```

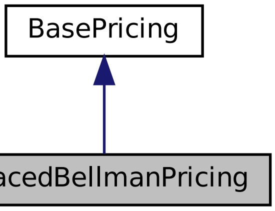
The documentation for this class was generated from the following files:

- include/SCIPSolver.h
- src/SCIPSolver.cpp

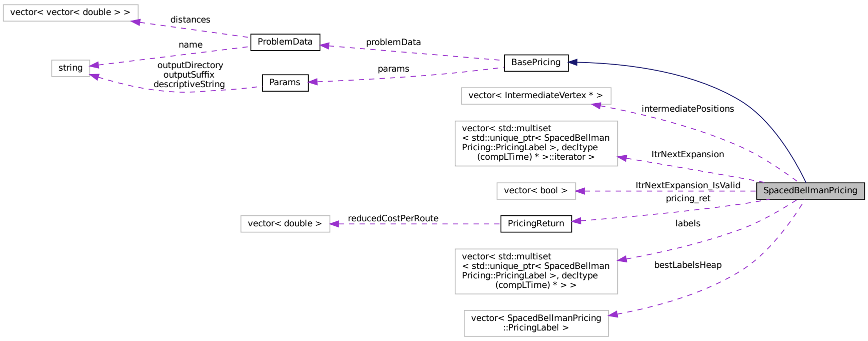
4.22 SpacedBellmanPricing Class Reference

```
#include <SpacedBellmanPricing.h>
```

Inheritance diagram for SpacedBellmanPricing:



Collaboration diagram for SpacedBellmanPricing:



Classes

- class PricingLabel

Public Member Functions

- void SetHeuristicPricing (bool value)
- SpacedBellmanPricing (Params ∗params, ProblemData ∗problemData)
- ∼SpacedBellmanPricing ()
- PricingReturn Price (int vehicle_id, int n_routes, vector< double > &alpha_duals, vector< double > &beta↩ _duals, vector< Route > &outRoutes, vector< int > consideredRequests, std::set< pair< int, int >> forbiddenEdges, map< pair< int, int >, double > edgeDuals)

Public Attributes

- bool limitNbLabels
- int maxLabels
- bool useRepeatedSetVerification

Private Member Functions

- bool TryAddLabel (PricingLabel &newLabel, int j, bool initial)
- bool comp3 (const PricingLabel &lhs, const PricingLabel &rhs)
- void FilterLabels (int index, PricingLabel &just_added, std::multiset< unique_ptr< PricingLabel >, decltype(compLTime) ∗ >::iterator just_added_itr)
- void Cleanup ()
- bool TryAddToBestLabelsHeap (PricingLabel &label)

Static Private Member Functions

- static bool compGTime (const PricingLabel &l1, const PricingLabel &l2)
- static bool compLRC (const unique_ptr< PricingLabel > &l1, const unique_ptr< PricingLabel > &l2)
- static bool compLTime (const unique_ptr< PricingLabel > &l1, const unique_ptr< PricingLabel > &l2)
- static bool compLRCRef (const PricingLabel &l1, const PricingLabel &l2)
- static bool compLTimeRef (const PricingLabel &l1, const PricingLabel &l2)
- static bool compGRC (const PricingLabel &l1, const PricingLabel &l2)
- static bool comp (const PricingLabel &lhs, const PricingLabel &rhs)
- static bool comp2 (const PricingLabel ∗lhs, const PricingLabel ∗rhs)

Private Attributes

- PricingReturn pricing_ret
- vector< std::multiset< std::unique_ptr< PricingLabel >, decltype(compLTime) ∗ > > labels
- bool heuristicPricing
- vector< std::multiset< std::unique_ptr< PricingLabel >, decltype(compLTime) ∗ >::iterator > ItrNextExpansion
- vector< bool > ItrNextExpansion_IsValid
- vector< IntermediateVertex ∗ > intermediatePositions
- size_t total_labels
- size_t n_desired_routes
- vector< PricingLabel > bestLabelsHeap

Additional Inherited Members

4.22.1 Constructor & Destructor Documentation

4.22.1.1 SpacedBellmanPricing()

```
SpacedBellmanPricing::SpacedBellmanPricing (
            Params ∗ params,
            ProblemData ∗ problemData )
```

4.22.1.2 ∼SpacedBellmanPricing()

```
SpacedBellmanPricing::∼SpacedBellmanPricing ( )  [inline]
```

4.22.2 Member Function Documentation

4.22.2.1 Cleanup()

```
void SpacedBellmanPricing::Cleanup ( )  [private]
```

4.22.2.2 comp()

```
bool SpacedBellmanPricing::comp (
             const PricingLabel & lhs,
             const PricingLabel & rhs )  [static], [private]
```

4.22.2.3 comp2()

```
bool SpacedBellmanPricing::comp2 (
             const PricingLabel * lhs,
             const PricingLabel * rhs )  [static], [private]
```

4.22.2.4 comp3()

```
bool SpacedBellmanPricing::comp3 (
             const PricingLabel & lhs,
             const PricingLabel & rhs )  [private]
```

4.22.2.5 compGRC()

```
bool SpacedBellmanPricing::compGRC (
             const PricingLabel & l1,
             const PricingLabel & l2 )  [static], [private]
```

4.22.2.6 compGTime()

```
bool SpacedBellmanPricing::compGTime (
            const PricingLabel & l1,
            const PricingLabel & l2 )  [static], [private]
```

4.22.2.7 compLRC()

```
bool SpacedBellmanPricing::compLRC (
            const unique_ptr< PricingLabel > & l1,
            const unique_ptr< PricingLabel > & l2 )  [static], [private]
```

4.22.2.8 compLRCRef()

```
bool SpacedBellmanPricing::compLRCRef (
            const PricingLabel & l1,
            const PricingLabel & l2 )  [static], [private]
```

4.22.2.9 compLTime()

```
bool SpacedBellmanPricing::compLTime (
            const unique_ptr< PricingLabel > & l1,
            const unique_ptr< PricingLabel > & l2 )  [static], [private]
```

4.22.2.10 compLTimeRef()

```
bool SpacedBellmanPricing::compLTimeRef (
            const PricingLabel & l1,
            const PricingLabel & l2 )  [static], [private]
```

4.22.2.11 FilterLabels()

```
void SpacedBellmanPricing::FilterLabels (
            int index,
            PricingLabel & just_added,
            std::multiset< unique_ptr< PricingLabel >, decltype(compLTime) * >::iterator
just_added_itr )  [private]
```

4.22.2.12 Price()

```
PricingReturn SpacedBellmanPricing::Price (
             int vehicle_id,
             int n_routes,
             vector< double > & alpha_duals,
             vector< double > & beta_duals,
             vector< Route > & outRoutes,
             vector< int > consideredRequests,
             std::set< pair< int, int >> forbiddenEdges,
             map< pair< int, int >, double > edgeDuals )
```

4.22.2.13 SetHeuristicPricing()

```
void SpacedBellmanPricing::SetHeuristicPricing (
             bool value )  [inline]
```

4.22.2.14 TryAddLabel()

```
bool SpacedBellmanPricing::TryAddLabel (
             PricingLabel & newLabel,
             int j,
             bool initial = false )  [private]
```

4.22.2.15 TryAddToBestLabelsHeap()

```
bool SpacedBellmanPricing::TryAddToBestLabelsHeap (
             PricingLabel & label )  [private]
```

4.22.3 Member Data Documentation

4.22.3.1 bestLabelsHeap

```
vector<PricingLabel> SpacedBellmanPricing::bestLabelsHeap  [private]
```

4.22.3.2 heuristicPricing

```
bool SpacedBellmanPricing::heuristicPricing [private]
```

4.22.3.3 intermediatePositions

```
vector<IntermediateVertex*> SpacedBellmanPricing::intermediatePositions [private]
```

4.22.3.4 ItrNextExpansion

```
vector<std::multiset <std::unique_ptr<PricingLabel>, decltype(compLTime)* >::iterator> Spaced↩
BellmanPricing::ItrNextExpansion [private]
```

4.22.3.5 ItrNextExpansion_IsValid

```
vector<bool> SpacedBellmanPricing::ItrNextExpansion_IsValid [private]
```

4.22.3.6 labels

```
vector< std::multiset < std::unique_ptr<PricingLabel> , decltype(compLTime)* > > Spaced↩
BellmanPricing::labels [private]
```

4.22.3.7 limitNbLabels

```
bool SpacedBellmanPricing::limitNbLabels
```

4.22.3.8 maxLabels

```
int SpacedBellmanPricing::maxLabels
```

4.22.3.9 n_desired_routes

```
size_t SpacedBellmanPricing::n_desired_routes [private]
```

4.22.3.10 pricing_ret

```
PricingReturn SpacedBellmanPricing::pricing_ret [private]
```

4.22.3.11 total_labels

```
size_t SpacedBellmanPricing::total_labels [private]
```

4.22.3.12 useRepeatedSetVerification

```
bool SpacedBellmanPricing::useRepeatedSetVerification
```

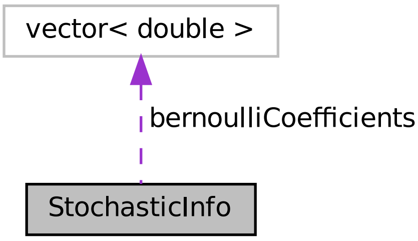
The documentation for this class was generated from the following files:

- include/SpacedBellmanPricing.h
- src/SpacedBellmanPricing.cpp

4.23 StochasticInfo Class Reference

```
#include <StochasticInfo.h>
```

Collaboration diagram for StochasticInfo:



Public Member Functions

- void SetupGrid (double lx, double ux, double ly, double uy, int nbVertical, int nbHorizontal, double bernoulli↩ Coefficient)
- StochasticInfo ()
- void GenerateScenarios (const ProblemData &base, int n_scenarios, vector< ProblemData > &out↩ Scenarios)

Private Attributes

- vector< double > bernoulliCoefficients
- double lx
- double ux
- double ly
- double uy
- int nbVertical
- int nbHorizontal

4.23.1 Constructor & Destructor Documentation

4.23.1.1 StochasticInfo()

```
StochasticInfo::StochasticInfo ( )
```

4.23.2 Member Function Documentation

4.23.2.1 GenerateScenarios()

```
void StochasticInfo::GenerateScenarios (
            const ProblemData & base,
            int n_scenarios,
            vector< ProblemData > & outScenarios )
```

4.23.2.2 SetupGrid()

```
void StochasticInfo::SetupGrid (
            double lx,
            double ux,
            double ly,
            double uy,
            int nbVertical,
            int nbHorizontal,
            double bernoulliCoefficient )
```

4.23.3 Member Data Documentation

4.23.3.1 bernoulliCoefficients

```
vector<double> StochasticInfo::bernoulliCoefficients  [private]
```

4.23.3.2 lx

```
double StochasticInfo::lx  [private]
```

4.23.3.3 ly

```
double StochasticInfo::ly  [private]
```

4.23.3.4 nbHorizontal

```
int StochasticInfo::nbHorizontal  [private]
```

4.23.3.5 nbVertical

```
int StochasticInfo::nbVertical  [private]
```

4.23.3.6 ux

```
double StochasticInfo::ux  [private]
```

4.23.3.7 uy

```
double StochasticInfo::uy  [private]
```

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

- include/StochasticInfo.h
- src/StochasticInfo.cpp

4.24 Vehicle Struct Reference

```
#include <ProblemData.h>
```

Public Attributes

- int type
- double timeAvailable
- int preferredWaitingStation

4.24.1 Member Data Documentation

4.24.1.1 preferredWaitingStation

```
int Vehicle::preferredWaitingStation
```

4.24.1.2 timeAvailable

```
double Vehicle::timeAvailable
```

4.24.1.3 type

```
int Vehicle::type
```

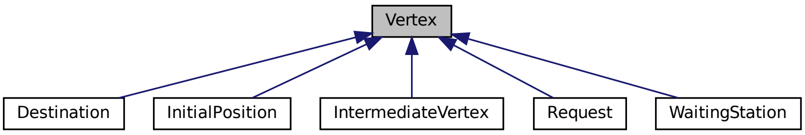
The documentation for this struct was generated from the following file:

- include/ProblemData.h

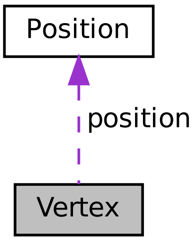
4.25 Vertex Struct Reference

```
#include <ProblemData.h>
```

Inheritance diagram for Vertex:



Collaboration diagram for Vertex:



Public Member Functions

- bool operator== (const Vertex &otherVertex) const

Public Attributes

- int identifier
- int id
- Position position
- int closestWaitingStation

4.25.1 Member Function Documentation

4.25.1.1 operator==()

```
bool Vertex::operator== (
             const Vertex & otherVertex ) const  [inline]
```

4.25.2 Member Data Documentation

4.25.2.1 closestWaitingStation

```
int Vertex::closestWaitingStation
```

4.25.2.2 id

```
int Vertex::id
```

4.25.2.3 identifier

```
int Vertex::identifier
```

4.25.2.4 position

```
Position Vertex::position
```

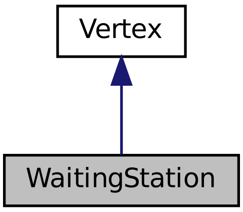
The documentation for this struct was generated from the following file:

- include/ProblemData.h

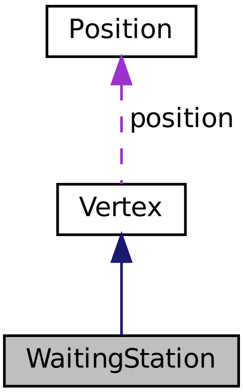
4.26 WaitingStation Struct Reference

```
#include <ProblemData.h>
```

Inheritance diagram for WaitingStation:



Collaboration diagram for WaitingStation:



Public Attributes

- int capacity

Additional Inherited Members

4.26.1 Member Data Documentation

4.26.1.1 capacity

```
int WaitingStation::capacity
```

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

- include/ProblemData.h

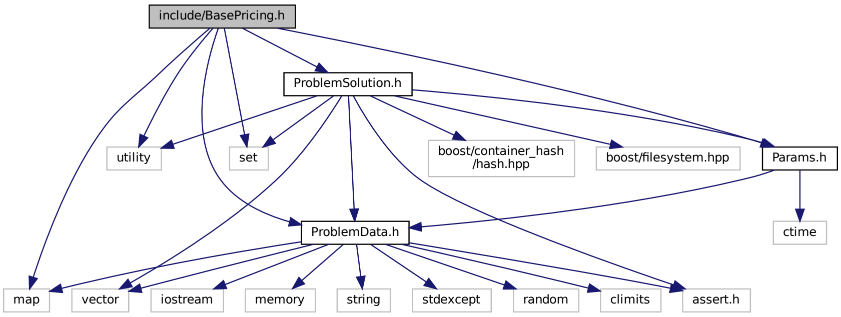
Chapter 5

File Documentation

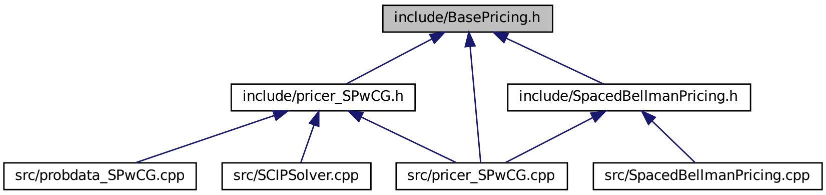
5.1 include/BasePricing.h File Reference

```
#include <utility>
#include <set>
#include <map>
#include "ProblemData.h"
#include "ProblemSolution.h"
#include "Params.h"
```

Include dependency graph for BasePricing.h:



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



Classes

- struct PricingReturn
- class BasePricing

Enumerations

- enum PricingReturnStatus { OK, FAIL }

5.1.1 Enumeration Type Documentation

5.1.1.1 PricingReturnStatus

```
enum PricingReturnStatus
```

Enumerator

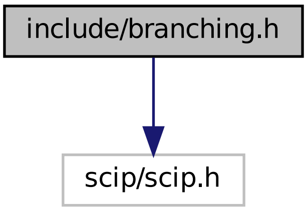
OK	
FAIL	

5.2 include/branching.h File Reference

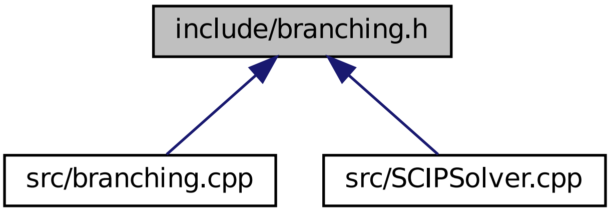
Implementation of Custom Branching Rules.

```
#include "scip/scip.h"
```

Include dependency graph for branching.h:



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



Functions

- SCIP_RETCODE SCIPincludeCustomBranchingRule (SCIP ∗scip)

5.2.1 Detailed Description

Implementation of Custom Branching Rules.

Author

André Mazal Krauss

This file implements branching rules for the problem.

5.2.2 Function Documentation

5.2.2.1 SCIPincludeCustomBranchingRule()

```
SCIP_RETCODE SCIPincludeCustomBranchingRule (
             SCIP * scip )
```

creates the branching rule and includes it in SCIP

Parameters

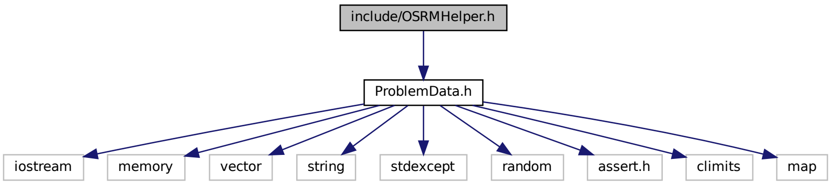
scip	SCIP data structure

5.3 include/OSRMHelper.h File Reference

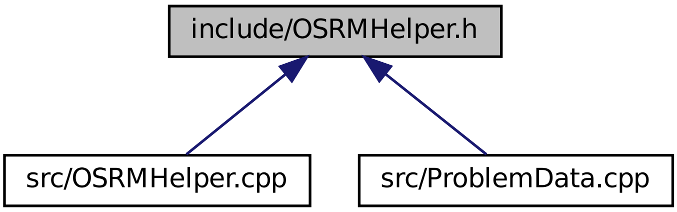
Definition of helper interface to use OSM data via OSRM.

```
#include "ProblemData.h"
```

Include dependency graph for OSRMHelper.h:



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



Classes

- class OSRMHelper

5.3.1 Detailed Description

Definition of helper interface to use OSM data via OSRM.

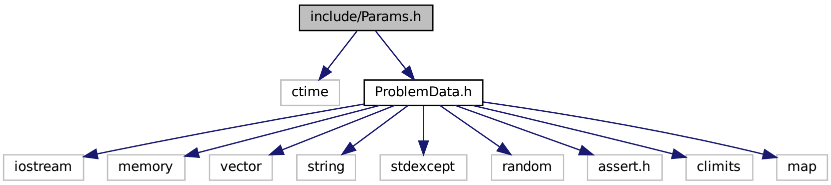
Author

André Mazal Krauss

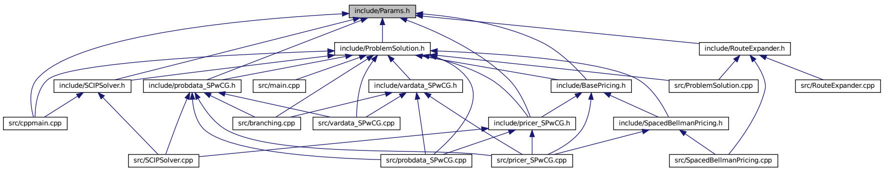
5.4 include/Params.h File Reference

```
#include <ctime>
#include "ProblemData.h"
```

Include dependency graph for Params.h:



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



Classes

- class Params

Enumerations

- enum PricingAlgorithm {
  PricingAlgorithm::DAG, PricingAlgorithm::bellman, PricingAlgorithm::spacedBellman, PricingAlgorithm::spacedBellman2, PricingAlgorithm::bellmanWSets, PricingAlgorithm::spacedBellmanWSets, PricingAlgorithm::PricerTester, PricingAlgorithm::hybrid }

5.4.1 Enumeration Type Documentation

Enumerator

5.4.1.1 PricingAlgorithm

`enum PricingAlgorithm [strong]`

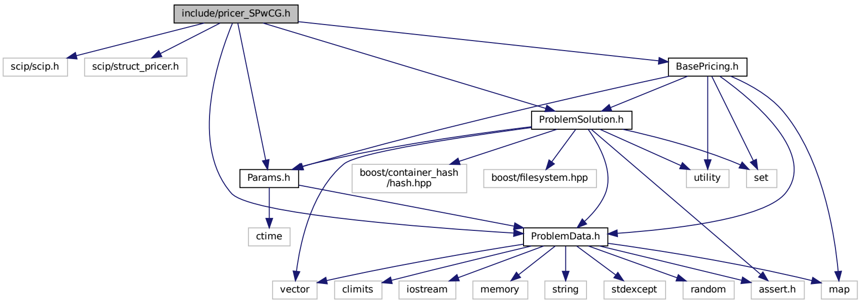
Enumerator

DAG	
bellman	
spacedBellman	
spacedBellman2	
bellmanWSets	
spacedBellmanWSets	
PricerTester	
hybrid	

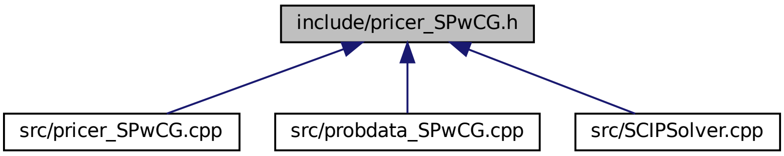
5.5 include/pricer_SPwCG.h File Reference

```
#include "scip/scip.h"
#include "scip/struct_pricer.h"
#include "Params.h"
#include "BasePricing.h"
#include "ProblemData.h"
#include "ProblemSolution.h"
```

Include dependency graph for pricer_SPwCG.h:



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



Classes

- struct SCIP_PricerData

  Variable pricer data used in the pricer.

Functions

- SCIP_RETCODE SCIPincludePricerSPwCG (SCIP ∗scip)
- SCIP_RETCODE SCIPpricerSPwCGActivate (SCIP ∗scip, SCIP_CONS ∗∗conss, Params ∗params, ProblemData ∗problemData)
- bool createRouteVariable (SCIP ∗scip, Params ∗params, SCIP_CONS ∗∗all_cons, ProblemData ∗problem←
  Data, Route ∗route, SCIP_VAR ∗∗retVar, bool initial_var, double reducedCost=0.0)

5.5.1 Function Documentation

5.5.1.1 createRouteVariable()

```
bool createRouteVariable (
            SCIP * scip,
            Params * params,
            SCIP_CONS ** all_cons,
            ProblemData * problemData,
            Route * route,
            SCIP_VAR ** retVar,
            bool initial_var,
            double reducedCost = 0.0 )
```

5.5.1.2 SCIPincludePricerSPwCG()

```
SCIP_RETCODE SCIPincludePricerSPwCG (
            SCIP * scip )
```

creates the binpacking variable pricer and includes it in SCIP

5.5.1.3 SCIPpricerSPwCGActivate()

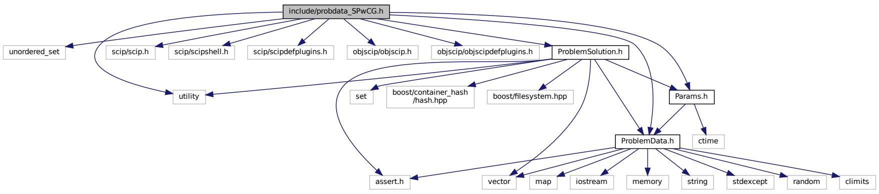
```
SCIP_RETCODE SCIPpricerSPwCGActivate (
            SCIP * scip,
            SCIP_CONS ** conss,
            Params * params,
            ProblemData * problemData )
```

added problem specific data to pricer and activates pricer

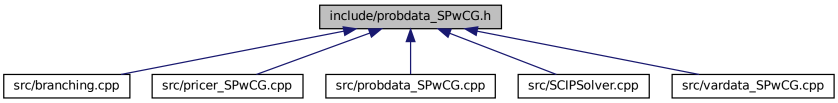
5.6 include/probdata_SPwCG.h File Reference

```
#include <unordered_set>
#include <utility>
#include "scip/scip.h"
#include "scip/scipshell.h"
#include "scip/scipdefplugins.h"
#include "objscip/objscip.h"
#include "objscip/objscipdefplugins.h"
#include "ProblemData.h"
#include "ProblemSolution.h"
#include "Params.h"
```

Include dependency graph for probdata_SPwCG.h:



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

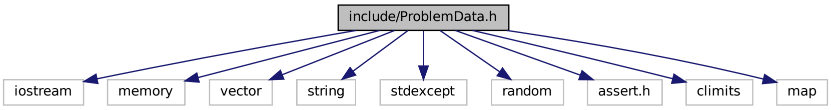


5.7 include/ProblemData.h File Reference

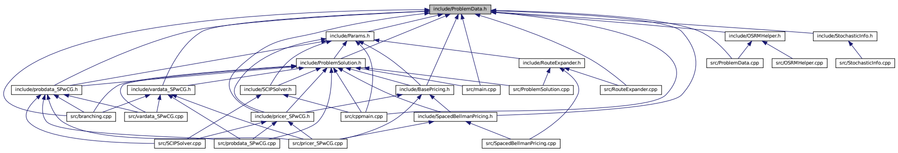
```
#include <iostream>
#include <memory>
#include <vector>
#include <string>
#include <stdexcept>
#include <random>
```

```
#include <assert.h>
#include <climits>
#include <map>
```

Include dependency graph for ProblemData.h:



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



Classes

- struct Vehicle
- struct Position
- struct Vertex
- struct IntermediateVertex
- struct Destination
- struct InitialPosition
- struct Request
- struct WaitingStation
- class ProblemData

Enumerations

- enum DistanceType { DistanceType::euclidian, DistanceType::geodesic, DistanceType::osrm }
- enum WaitingStationPolicy { WaitingStationPolicy::mandatoryStopInFixedStation, WaitingStationPolicy::optionalStopInFixedSta WaitingStationPolicy::optionalStopInClosestWaitingStation, WaitingStationPolicy::bestOptionalStop }

5.7.1 Enumeration Type Documentation

5.7.1.1 DistanceType

```
enum DistanceType  [strong]
```

Enumerator

euclidian	
geodesic	
osrm	

5.7.1.2 WaitingStationPolicy

```
enum WaitingStationPolicy [strong]
```

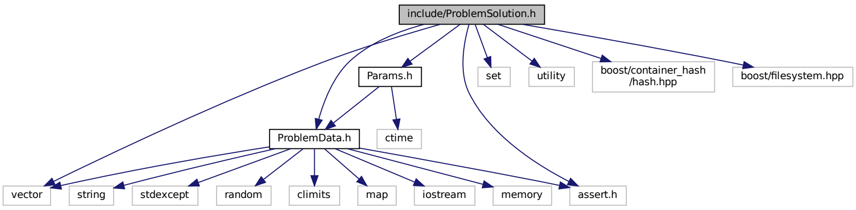
Enumerator

mandatoryStopInFixedStation	
optionalStopInFixedStation	
optionalStopInClosestWaitingStation	
bestOptionalStop	

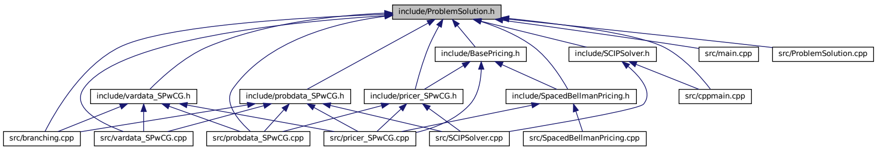
5.8 include/ProblemSolution.h File Reference

```
#include <vector>
#include "ProblemData.h"
#include "Params.h"
#include <assert.h>
#include <set>
#include <utility>
#include <boost/container_hash/hash.hpp>
#include <boost/filesystem.hpp>
```

Include dependency graph for ProblemSolution.h:



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



Classes

- class Route
- struct ResponseSummary
- class ProblemSolution

Macros

- #define GIT_COMMIT_HASH "?"

5.8.1 Macro Definition Documentation

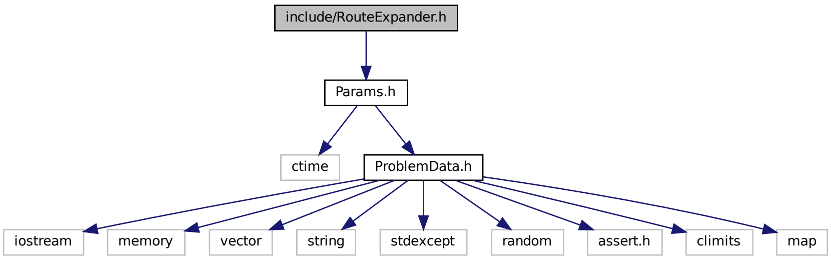
5.8.1.1 GIT_COMMIT_HASH

```
#define GIT_COMMIT_HASH "?"
```

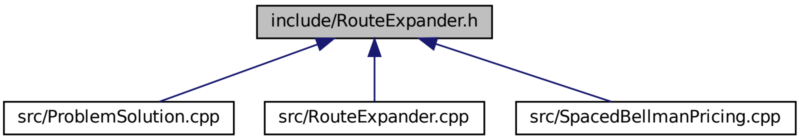
5.9 include/RouteExpander.h File Reference

```
#include "Params.h"
```

Include dependency graph for RouteExpander.h:



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



Classes

- class RouteExpander

Enumerations

- enum WhichStation { closest, best, preferred }

5.9.1 Enumeration Type Documentation

5.9.1.1 WhichStation

```
enum WhichStation
```

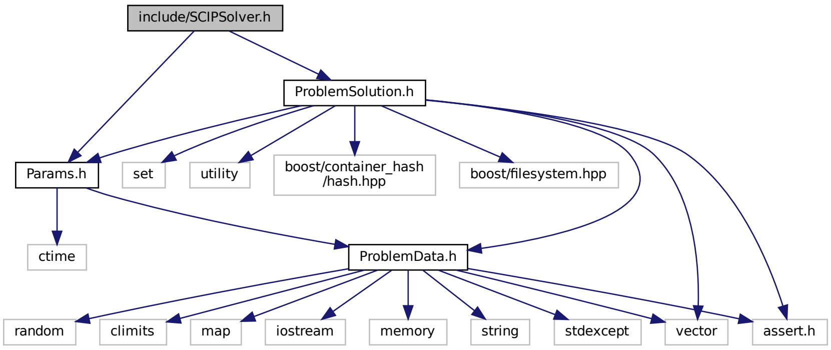
Enumerator

closest	
best	
preferred	

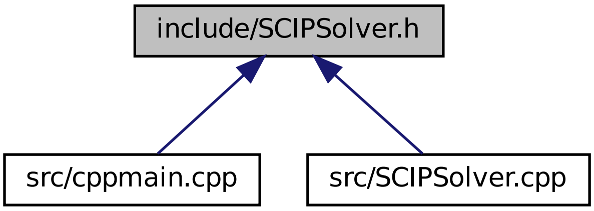
5.10 include/SCIPSolver.h File Reference

```
#include "Params.h"
#include "ProblemSolution.h"
```

Include dependency graph for SCIPSolver.h:



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



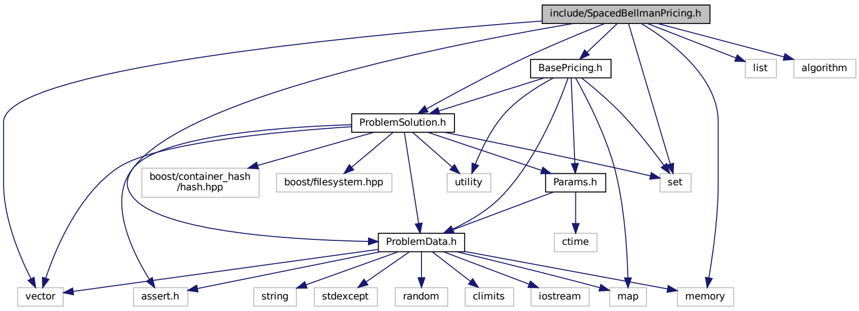
Classes

- class SCIPSolver

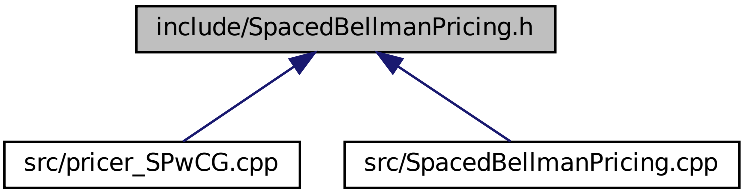
5.11 include/SpacedBellmanPricing.h File Reference

```
#include <vector>
#include <memory>
#include <set>
#include <list>
#include <algorithm>
#include "ProblemData.h"
#include "ProblemSolution.h"
#include "BasePricing.h"
```

Include dependency graph for SpacedBellmanPricing.h:



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



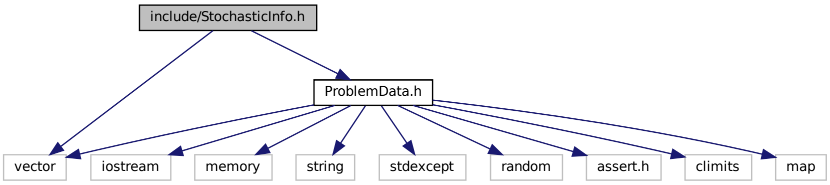
Classes

- class SpacedBellmanPricing
- class SpacedBellmanPricing::PricingLabel

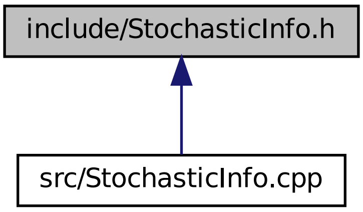
5.12 include/StochasticInfo.h File Reference

```
#include <vector>
#include "ProblemData.h"
```

Include dependency graph for StochasticInfo.h:



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



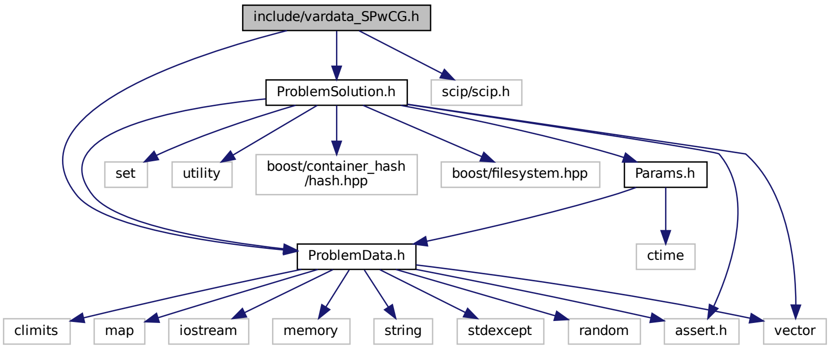
Classes

- class DemandScenario
- class AggregatedProblem
- class StochasticInfo

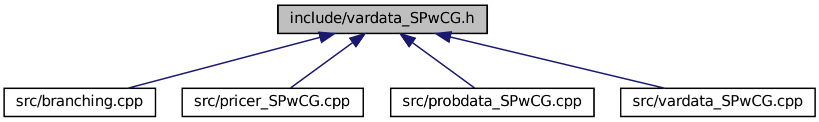
5.13 include/vardata_SPwCG.h File Reference

```
#include "ProblemData.h"
#include "ProblemSolution.h"
#include "scip/scip.h"
```

Include dependency graph for vardata_SPwCG.h:



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



Functions

- SCIP_RETCODE SCIPvardataCreateBinpacking (SCIP ∗scip, SCIP_VARDATA ∗∗vardata, int ∗consids, int ∗conscoeffs, int nconss, Route ∗route)
- int SCIPvardataGetNConsids (SCIP_VARDATA ∗vardata)
- int ∗ SCIPvardataGetConsids (SCIP_VARDATA ∗vardata)
- Route ∗ SCIPvardataGetRoute (SCIP_VARDATA ∗vardata)
- SCIP_RETCODE SCIPcreateVarBinpacking (SCIP ∗scip, SCIP_VAR ∗∗var, const char ∗name, SCIP_Real obj, SCIP_Bool initial, SCIP_Bool removable, bool relaxed, SCIP_VARDATA ∗vardata)
- void SCIPvardataPrint (SCIP ∗scip, SCIP_VARDATA ∗vardata, FILE ∗file)

5.13.1 Function Documentation

5.13.1.1 SCIPcreateVarBinpacking()

```
SCIP_RETCODE SCIPcreateVarBinpacking (
             SCIP * scip,
             SCIP_VAR ** var,
             const char * name,
             SCIP_Real obj,
             SCIP_Bool initial,
             SCIP_Bool removable,
             bool relaxed,
             SCIP_VARDATA * vardata )
```

creates variable

5.13.1.2 SCIPvardataCreateBinpacking()

```
SCIP_RETCODE SCIPvardataCreateBinpacking (
             SCIP * scip,
             SCIP_VARDATA ** vardata,
             int * consids,
             int * conscoeffs,
             int nconsids,
             Route * route )
```

create variable data

5.13.1.3 SCIPvardataGetConsids()

```
int* SCIPvardataGetConsids (
             SCIP_VARDATA * vardata )
```

returns sorted constraint id array

5.13.1.4 SCIPvardataGetNConsids()

```
int SCIPvardataGetNConsids (
             SCIP_VARDATA * vardata )
```

get number of constraints

5.13.1.5 SCIPvardataGetRoute()

```
Route* SCIPvardataGetRoute (
             SCIP_VARDATA * vardata )
```

returns route pointer

5.13.1.6 SCIPvardataPrint()

```
void SCIPvardataPrint (
            SCIP * scip,
            SCIP_VARDATA * vardata,
            FILE * file )
```

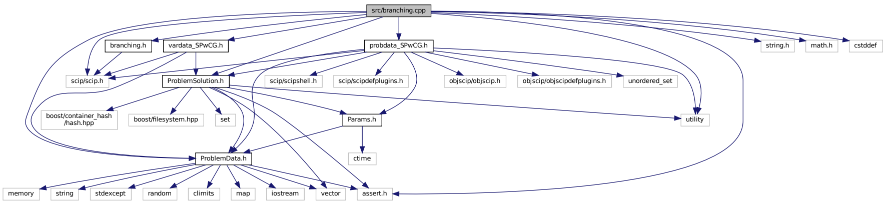
prints vardata to file stream

5.14 src/branching.cpp File Reference

Implementation of custom branching rules.

```
#include <assert.h>
#include <string.h>
#include <math.h>
#include <utility>
#include <cstddef>
#include "branching.h"
#include "scip/scip.h"
#include "probdata_SPwCG.h"
#include "vardata_SPwCG.h"
#include "ProblemData.h"
#include "ProblemSolution.h"
```

Include dependency graph for branching.cpp:



Macros

Branching rule properties

- #define BRANCHRULE_NAME "CustomRules"
- #define BRANCHRULE_DESC "Custom made branching rule"
- #define BRANCHRULE_PRIORITY 50000
- #define BRANCHRULE_MAXDEPTH -1
- #define BRANCHRULE_MAXBOUNDDIST 1.0

Functions

Helper methods

- SCIP_RETCODE SCIPcreateConsSumVehicle (SCIP ∗scip, SCIP_CONS ∗∗cons, const char ∗name, int vehicle_id, double desired_sum, SCIP_NODE ∗node, SCIP_Bool local)

  Creates a new branching constraint constraining the usage of a vehicle to a certain value (0/1)
- SCIP_RETCODE SCIPcreateConsSumEdge (SCIP ∗scip, SCIP_CONS ∗∗cons, const char ∗name, pair< int, int > edge, double desired_sum, SCIP_NODE ∗node, SCIP_Bool local)

  Creates a new branching constraint constraining the usage of an edge to a certain value (0/1)
- static bool IsSimpleBranchingRepeated (SCIP ∗scip, ProblemData ∗problemData, SCIP_VAR ∗var)

  Checks if variable has already been branched at for the current node.
- static bool IsVehicleBranchingRepeated (SCIP ∗scip, ProblemData ∗problemData, int chosenVehicle)

  Checks if vehicle branching rule has already been applied for a given vehicle at the current node.
- static bool IsEdgeBranchingRepeated (SCIP ∗scip, ProblemData ∗problemData, pair< int, int > chosen↩ Edge)

  Checks if edge branching rule has already been applied for a given edge at the current node.
- static SCIP_RETCODE PrintConstraintValues (SCIP ∗scip, std::map< SCIP_VAR ∗, double > &var_↩ solutionValue)

  Debug function for printing out branching constraints for a given value.

Callback methods

- static SCIP_RETCODE branchExeclpBranchingRules (SCIP ∗scip, SCIP_BRANCHRULE ∗branchrule, SCIP_Bool allowaddcons, SCIP_RESULT ∗result)

Interface methods

- SCIP_RETCODE SCIPincludeCustomBranchingRule (SCIP ∗scip)

5.14.1 Detailed Description

Implementation of custom branching rules.

Author

André Mazal Krauss

This file implements branching rules for the problem.

5.14.2 Macro Definition Documentation

5.14.2.1 BRANCHRULE_DESC

```
#define BRANCHRULE_DESC "Custom made branching rule"
```

5.14.2.2 BRANCHRULE_MAXBOUNDDIST

```
#define BRANCHRULE_MAXBOUNDDIST 1.0
```

5.14.2.3 BRANCHRULE_MAXDEPTH

```
#define BRANCHRULE_MAXDEPTH -1
```

5.14.2.4 BRANCHRULE_NAME

```
#define BRANCHRULE_NAME "CustomRules"
```

5.14.2.5 BRANCHRULE_PRIORITY

```
#define BRANCHRULE_PRIORITY 50000
```

5.14.3 Function Documentation

5.14.3.1 branchExeclpBranchingRules()

```
static SCIP_RETCODE branchExeclpBranchingRules (
            SCIP * scip,
            SCIP_BRANCHRULE * branchrule,
            SCIP_Bool allowaddcons,
            SCIP_RESULT * result )  [static]
```

Branching execution method for fractional LP solutions. This method is called by SCIP when it is needs to perform branching In our implementation, this single function inspects the variables at the current node, decides on the best of the 3 branching rules to apply, and applies it. If branching options have been exhausted, it reports this to scip

Its definition must be exactly so to dialogue correctly with SCIP. We refer to SCIP's documentation for SCIP_DE↩ CL_BRANCHEXECLP

5.14.3.2 IsEdgeBranchingRepeated()

```
static bool IsEdgeBranchingRepeated (
            SCIP * scip,
            ProblemData * problemData,
            pair< int, int > chosenEdge )  [static]
```

Checks if edge branching rule has already been applied for a given edge at the current node.

5.14.3.3 IsSimpleBranchingRepeated()

```
static bool IsSimpleBranchingRepeated (
            SCIP * scip,
            ProblemData * problemData,
            SCIP_VAR * var ) [static]
```

Checks if variable has already been branched at for the current node.

5.14.3.4 IsVehicleBranchingRepeated()

```
static bool IsVehicleBranchingRepeated (
            SCIP * scip,
            ProblemData * problemData,
            int chosenVehicle ) [static]
```

Checks if vehicle branching rule has already been applied for a given vehicle at the current node.

5.14.3.5 PrintConstraintValues()

```
static SCIP_RETCODE PrintConstraintValues (
            SCIP * scip,
            std::map< SCIP_VAR *, double > & var_solutionValue ) [static]
```

Debug function for printing out branching constraints for a given value.

5.14.3.6 SCIPcreateConsSumEdge()

```
SCIP_RETCODE SCIPcreateConsSumEdge (
            SCIP * scip,
            SCIP_CONS ** cons,
            const char * name,
            pair< int, int > edge,
            double desired_sum,
            SCIP_NODE * node,
            SCIP_Bool local )
```

Creates a new branching constraint constraining the usage of an edge to a certain value (0/1)

Parameters

scip	SCIP data structure
cons	pointer to hold the created constraint
name	name of constraint
node	the node in the B&B-tree at which the cons is sticking
local	is constraint only valid locally?

5.14.3.7 SCIPcreateConsSumVehicle()

```
SCIP_RETCODE SCIPcreateConsSumVehicle (
             SCIP * scip,
             SCIP_CONS ** cons,
             const char * name,
             int vehicle_id,
             double desired_sum,
             SCIP_NODE * node,
             SCIP_Bool local )
```

Creates a new branching constraint constraining the usage of a vehicle to a certain value (0/1)

Parameters

scip	SCIP data structure
cons	pointer to hold the created constraint
name	name of constraint
node	the node in the B&B-tree at which the cons is sticking
local	is constraint only valid locally?

5.14.3.8 SCIPincludeCustomBranchingRule()

```
SCIP_RETCODE SCIPincludeCustomBranchingRule (
             SCIP * scip )
```

creates the branching rule and includes it in SCIP

Parameters

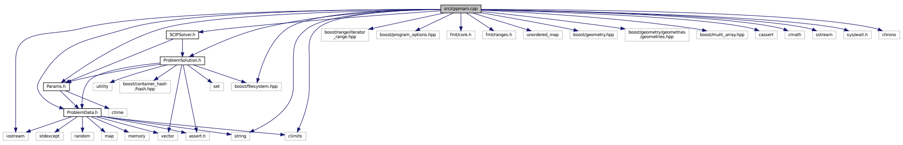
scip	SCIP data structure

5.15 src/cppmain.cpp File Reference

```
#include <iostream>
#include <boost/filesystem.hpp>
#include <boost/range/iterator_range.hpp>
#include <boost/program_options.hpp>
#include <fmt/core.h>
#include <fmt/ranges.h>
#include <string>
#include <unordered_map>
#include <boost/geometry.hpp>
#include <boost/geometry/geometries/geometries.hpp>
```

```
#include <boost/multi_array.hpp>
#include <cassert>
#include <cmath>
#include <sstream>
#include <climits>
#include <sys/wait.h>
#include "ProblemData.h"
#include "ProblemSolution.h"
#include "Params.h"
#include "SCIPSolver.h"
#include <chrono>
```

Include dependency graph for cppmain.cpp:



Functions

- bool FindVInstanceByIndex (string dirPath, int instance_index, string osmPath, ProblemData &problemData, string timeHorizonUsage, double timeHorizon, int setNbVehicles)
- bool ParseCommandLine (int argc, char ∗∗argv, Params &params, ProblemData &problemData)
- int ComputingCanadaMain (int argc, char ∗∗argv)
- int main (int argc, char ∗∗argv)

5.15.1 Function Documentation

5.15.1.1 ComputingCanadaMain()

```
int ComputingCanadaMain (
             int argc,
             char ** argv )
```

5.15.1.2 FindVInstanceByIndex()

```
bool FindVInstanceByIndex (
             string dirPath,
             int instance_index,
             string osmPath,
             ProblemData & problemData,
             string timeHorizonUsage,
             double timeHorizon,
             int setNbVehicles )
```

5.15.1.3 main()

```
int main (
            int argc,
            char ** argv )
```

Parameters

argc	number of arguments from the shell
argv	array of shell arguments

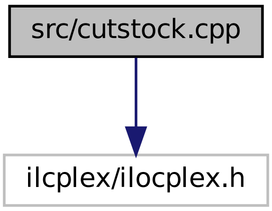
5.15.1.4 ParseCommandLine()

```
bool ParseCommandLine (
            int argc,
            char ** argv,
            Params & params,
            ProblemData & problemData )
```

5.16 src/cutstock.cpp File Reference

`#include <ilcplex/ilocplex.h>`

Include dependency graph for cutstock.cpp:



Macros

- #define IL_STD
- #define RC_EPS 1.0e-6

Functions

- static void readData (const char ∗filename, IloNum &rollWidth, IloNumArray &size, IloNumArray &amount)
- static void report1 (IloCplex &cutSolver, IloNumVarArray Cut, IloRangeArray Fill)
- static void report2 (IloAlgorithm &patSolver, IloNumVarArray Use, IloObjective obj)
- static void report3 (IloCplex &cutSolver, IloNumVarArray Cut)
- int cutstock_main (int argc, char ∗∗argv)

  MAIN PROGRAM ///.

5.16.1 Macro Definition Documentation

5.16.1.1 IL_STD

```
#define IL_STD
```

5.16.1.2 RC_EPS

```
#define RC_EPS 1.0e-6
```

5.16.2 Function Documentation

5.16.2.1 cutstock_main()

```
int cutstock_main (
             int argc,
             char ** argv )
```

MAIN PROGRAM ///.

CUTTING-OPTIMIZATION PROBLEM ///

PATTERN-GENERATION PROBLEM ///

COLUMN-GENERATION PROCEDURE ///

COLUMN-GENERATION PROCEDURE ///

OPTIMIZE OVER CURRENT PATTERNS ///

FIND AND ADD A NEW PATTERN ///

5.16.2.2 readData()

```
static void readData (
             const char * filename,
             IloNum & rollWidth,
             IloNumArray & size,
             IloNumArray & amount )  [static]
```

5.16.2.3 report1()

```
static void report1 (
             IloCplex & cutSolver,
             IloNumVarArray Cut,
             IloRangeArray Fill )  [static]
```

5.16.2.4 report2()

```
static void report2 (
             IloAlgorithm & patSolver,
             IloNumVarArray Use,
             IloObjective obj )  [static]
```

5.16.2.5 report3()

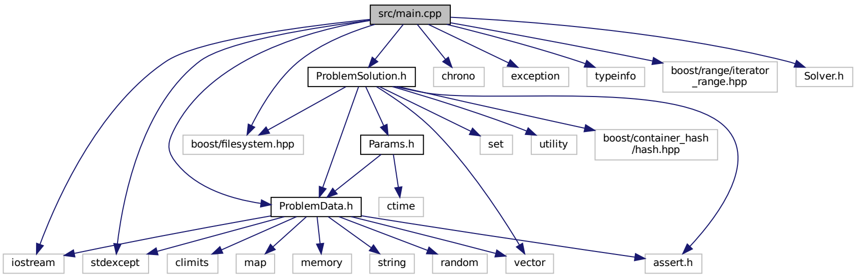
```
static void report3 (
             IloCplex & cutSolver,
             IloNumVarArray Cut )  [static]
```

5.17 src/main.cpp File Reference

```
#include <iostream>
#include <chrono>
#include <exception>
#include <typeinfo>
#include <stdexcept>
#include <boost/filesystem.hpp>
#include <boost/range/iterator_range.hpp>
#include "ProblemData.h"
#include "ProblemSolution.h"
```

```
#include "Solver.h"
```

Include dependency graph for main.cpp:



Macros

- #define RC_EPS 1.0e-6

Functions

- int run_tests ()
- void simulateOperation (Params ∗params)
- int main (int argc, char ∗∗argv)

5.17.1 Macro Definition Documentation

5.17.1.1 RC_EPS

```
#define RC_EPS 1.0e-6
```

5.17.2 Function Documentation

5.17.2.1 main()

```
int main (
            int argc,
            char ** argv )
```

5.17.2.2 run_tests()

```
int run_tests ( )
```

5.17.2.3 simulateOperation()

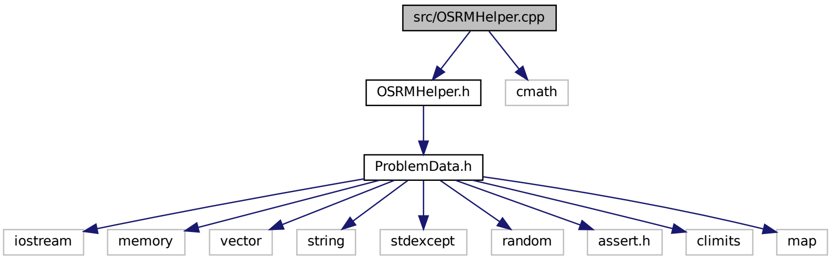
```
void simulateOperation (
            Params * params )
```

5.18 src/OSRMHelper.cpp File Reference

Implementation of a helper interface for OSRM.

```
#include "OSRMHelper.h"
#include <cmath>
```

Include dependency graph for OSRMHelper.cpp:



5.18.1 Detailed Description

Implementation of a helper interface for OSRM.

Author

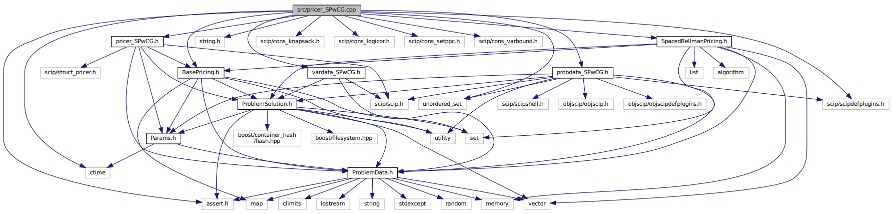
André Mazal Krauss

This file implements the usage of OSRM to query distance/time information from an OSM database

5.19 src/pricer_SPwCG.cpp File Reference

```
#include "pricer_SPwCG.h"
#include <assert.h>
#include <string.h>
#include <ctime>
#include <unordered_set>
#include "scip/cons_knapsack.h"
#include "scip/cons_logicor.h"
#include "scip/cons_setppc.h"
#include "scip/cons_varbound.h"
#include "scip/scipdefplugins.h"
#include "vardata_SPwCG.h"
#include "probdata_SPwCG.h"
#include "BasePricing.h"
#include "SpacedBellmanPricing.h"
```

Include dependency graph for pricer_SPwCG.cpp:



Functions

- SCIP_RETCODE SCIPincludePricerSPwCG (SCIP ∗scip)
- SCIP_RETCODE SCIPpricerSPwCGActivate (SCIP ∗scip, SCIP_CONS ∗∗conss, Params ∗params, ProblemData ∗problemData)

- static int ncons (ProblemData ∗problemData)
- static SCIP_DECL_PRICERFREE (pricerFreeSPwCG)
- static SCIP_DECL_PRICERINIT (pricerInitSPwCG)
- static SCIP_DECL_PRICEREXITSOL (pricerExitsolSPwCG)
- static bool compareArrays (int ∗arr1, int size1, int ∗arr2, int size2)
- static bool routeContainsRequest (ProblemData ∗problemData, Route ∗route, int reqId)
- static bool DoesRouteViolateBranching (SCIP ∗scip, ProblemData ∗problemData, Route ∗route)
- static void buildConsideredRequestsVector (SCIP ∗scip, ProblemData ∗problemData, std::vector< int > &outVec)
- static void buildForbiddenEdges (SCIP ∗scip, ProblemData ∗problemData, std::set< pair< int, int >> &outSet)
- bool createRouteVariable (SCIP ∗scip, Params ∗params, SCIP_CONS ∗∗all_cons, ProblemData ∗problemData, Route ∗route, SCIP_VAR ∗∗retVar, bool initial_var, double reducedCost)
- static SCIP_RETCODE DoPricing (SCIP ∗scip, SCIP_PRICER ∗pricer, SCIP_Bool farkas, SCIP_RESULT ∗result)
- static SCIP_DECL_PRICERREDCOST (pricerRedcostSPwCG)
- static SCIP_DECL_PRICERFARKAS (pricerFarkasBinpacking)

Pricer properties

- #define PRICER_NAME "SPwCG"
- #define PRICER_DESC "pricer for pricing new routes"
- #define PRICER_PRIORITY 0
- #define PRICER_DELAY TRUE /∗ only call pricer if all problem variables have non-negative reduced costs ∗/
- static void checkForVarRedCosts (SCIP ∗scip)

5.19.1 Macro Definition Documentation

5.19.1.1 PRICER_DELAY

```
#define PRICER_DELAY TRUE /* only call pricer if all problem variables have non-negative reduced
costs */
```

5.19.1.2 PRICER_DESC

```
#define PRICER_DESC "pricer for pricing new routes"
```

5.19.1.3 PRICER_NAME

```
#define PRICER_NAME "SPwCG"
```

5.19.1.4 PRICER_PRIORITY

```
#define PRICER_PRIORITY 0
```

5.19.2 Function Documentation

5.19.2.1 buildConsideredRequestsVector()

```
static void buildConsideredRequestsVector (
            SCIP * scip,
            ProblemData * problemData,
            std::vector< int > & outVec )  [static]
```

5.19.2.2 buildForbiddenEdges()

```
static void buildForbiddenEdges (
             SCIP * scip,
             ProblemData * problemData,
             std::set< pair< int, int >> & outSet )  [static]
```

5.19.2.3 checkForVarRedCosts()

```
static void checkForVarRedCosts (
             SCIP * scip )  [static]
```

5.19.2.4 compareArrays()

```
static bool compareArrays (
             int * arr1,
             int size1,
             int * arr2,
             int size2 )  [static]
```

5.19.2.5 createRouteVariable()

```
bool createRouteVariable (
             SCIP * scip,
             Params * params,
             SCIP_CONS ** all_cons,
             ProblemData * problemData,
             Route * route,
             SCIP_VAR ** retVar,
             bool initial_var,
             double reducedCost )
```

5.19.2.6 DoesRouteViolateBranching()

```
static bool DoesRouteViolateBranching (
             SCIP * scip,
             ProblemData * problemData,
             Route * route )  [static]
```

5.19.2.7 DoPricing()

```
static SCIP_RETCODE DoPricing (
             SCIP * scip,
             SCIP_PRICER * pricer,
             SCIP_Bool farkas,
             SCIP_RESULT * result )  [static]
```

Parameters

scip	SCIP data structure
pricer	pricer
farkas	TRUE: Farkas pricing; FALSE: Redcost pricing

5.19.2.8 ncons()

```
static int ncons (
             ProblemData * problemData )  [static]
```

name Callback methods

5.19.2.9 routeContainsRequest()

```
static bool routeContainsRequest (
             ProblemData * problemData,
             Route * route,
             int reqId )  [static]
```

5.19.2.10 SCIP_DECL_PRICEREXITSOL()

```
static SCIP_DECL_PRICEREXITSOL (
             pricerExitsolSPwCG  )  [static]
```

solving process deinitialization method of variable pricer (called before branch and bound process data is freed)

5.19.2.11 SCIP_DECL_PRICERFARKAS()

```
static SCIP_DECL_PRICERFARKAS (
             pricerFarkasBinpacking  )  [static]
```

farkas pricing method of variable pricer for infeasible LPs

5.19.2.12 SCIP_DECL_PRICERFREE()

```
static SCIP_DECL_PRICERFREE (
             pricerFreeSPwCG  )  [static]
```

destructor of variable pricer to free user data (called when SCIP is exiting)

5.19.2.13 SCIP_DECL_PRICERINIT()

```
static SCIP_DECL_PRICERINIT (
             pricerInitSPwCG ) [static]
```

initialization method of variable pricer (called after problem was transformed)

5.19.2.14 SCIP_DECL_PRICERREDCOST()

```
static SCIP_DECL_PRICERREDCOST (
             pricerRedcostSPwCG ) [static]
```

reduced cost pricing method of variable pricer for feasible LPs

5.19.2.15 SCIPincludePricerSPwCG()

```
SCIP_RETCODE SCIPincludePricerSPwCG (
             SCIP * scip )
```

creates the binpacking variable pricer and includes it in SCIP

Parameters

scip	SCIP data structure

5.19.2.16 SCIPpricerSPwCGActivate()

```
SCIP_RETCODE SCIPpricerSPwCGActivate (
             SCIP * scip,
             SCIP_CONS ** conss,
             Params * params,
             ProblemData * problemData )
```

added problem specific data to pricer and activates pricer

Parameters

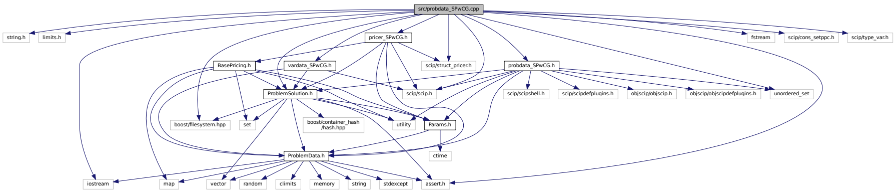
scip	SCIP data structure
conss	set covering constraints for the items

5.20 src/probdata_SPwCG.cpp File Reference

```
#include <string.h>
#include <limits.h>
```

```
#include <iostream>
#include <fstream>
#include <boost/filesystem.hpp>
#include "probdata_SPwCG.h"
#include "pricer_SPwCG.h"
#include "vardata_SPwCG.h"
#include "scip/cons_setppc.h"
#include "scip/scip.h"
#include "scip/type_var.h"
#include <assert.h>
#include <unordered_set>
#include "ProblemSolution.h"
#include "scip/struct_pricer.h"
```

Include dependency graph for probdata_SPwCG.cpp:



Classes

- struct SCIP_ProbData

  Problem data which is accessible in all places.

Macros

Event handler properties

- #define EVENTHDLR_NAME "addedvar"
- #define EVENTHDLR_DESC "event handler for catching added variables"

Functions

- ProblemData ∗ GetProblemData (SCIP_ProbData ∗probdata)
- Params ∗ GetParams (SCIP_ProbData ∗probdata)
- bool IsVarRepeated (SCIP_ProbData ∗probdata, Route ∗route, SCIP ∗scip)
- void AddVehicleBranchingCons (SCIP ∗scip, SCIP_PROBDATA ∗probdata, SCIP_CONS ∗cons, int vehicle←_id)
- void AddEdgeBranchingCons (SCIP ∗scip, SCIP_PROBDATA ∗probdata, SCIP_CONS ∗cons, pair< int, int > edge)
- const vector< SCIP_CONS ∗ > ∗ GetVehicleBranchingConstraints (SCIP_PROBDATA ∗probdata)
- const vector< int > ∗ GetConstrainedVehicles (SCIP_PROBDATA ∗probdata)
- const vector< SCIP_CONS ∗ > ∗ GetEdgeBranchingConstraints (SCIP_PROBDATA ∗probdata)
- const vector< pair< int, int > > ∗ GetConstrainedEdges (SCIP_PROBDATA ∗probdata)
- void IncrementUsedBranchingRule (SCIP_PROBDATA ∗probdata, int ruleIndex)
- void LogRepeatedRoute (SCIP_PROBDATA ∗probdata, double reducedCost)
- SCIP_VAR ∗∗ SCIPprobdataGetVars (SCIP_PROBDATA ∗probdata)
- int SCIPprobdataGetNVars (SCIP_PROBDATA ∗probdata)

- SCIP_CONS ∗∗ SCIPprobdataGetCons (SCIP_PROBDATA ∗probdata)
- int SCIPprobdataGetNCons (SCIP_PROBDATA ∗probdata)

Local methods

- static int ncons (ProblemData ∗problemData)
- static SCIP_RETCODE probdataCreate (SCIP ∗scip, SCIP_PROBDATA ∗∗probdata, SCIP_VAR ∗∗vars, SCIP_CONS ∗∗conss, int nvars, Params ∗params, ProblemData ∗problemData)
- static SCIP_RETCODE probdataFree (SCIP ∗scip, SCIP_PROBDATA ∗∗probdata)

Callback methods of event handler

- SCIP_RETCODE SCIPprobdataAddVar (SCIP ∗scip, SCIP_PROBDATA ∗probdata, SCIP_VAR ∗var)
- static SCIP_DECL_EVENTEXEC (eventExecAddedVar)

Callback methods of problem data

- static SCIP_DECL_PROBDELORIG (probdelorigSPwCG)
- static SCIP_DECL_PROBDELTRANS (probdeltransSPwCG)
- static SCIP_DECL_PROBTRANS (probtransSPwCG)
- static SCIP_DECL_PROBINITSOL (probinitsolSPwCG)
- static SCIP_DECL_PROBEXITSOL (probexitsolSPwCG)

Interface methods

- SCIP_RETCODE loadProblem (SCIP ∗scip, Params ∗params, ProblemData ∗problemData)
- void QuerySolution (SCIP ∗scip, ProblemSolution &solution)
- ExecutionSummary GetExecutionSummary (SCIP ∗scip)
- void OutputDuals (SCIP ∗scip, Params ∗params)

5.20.1 Macro Definition Documentation

5.20.1.1 EVENTHDLR_DESC

```
#define EVENTHDLR_DESC "event handler for catching added variables"
```

5.20.1.2 EVENTHDLR_NAME

```
#define EVENTHDLR_NAME "addedvar"
```

5.20.2 Function Documentation

5.20.2.1 AddEdgeBranchingCons()

```
void AddEdgeBranchingCons (
            SCIP * scip,
            SCIP_PROBDATA * probdata,
            SCIP_CONS * cons,
            pair< int, int > edge )
```

5.20.2.2 AddVehicleBranchingCons()

```
void AddVehicleBranchingCons (
            SCIP * scip,
            SCIP_PROBDATA * probdata,
            SCIP_CONS * cons,
            int vehicle_id )
```

5.20.2.3 GetConstrainedEdges()

```
const vector<pair<int, int> >* GetConstrainedEdges (
            SCIP_PROBDATA * probdata )
```

5.20.2.4 GetConstrainedVehicles()

```
const vector<int>* GetConstrainedVehicles (
            SCIP_PROBDATA * probdata )
```

5.20.2.5 GetEdgeBranchingConstraints()

```
const vector<SCIP_CONS*>* GetEdgeBranchingConstraints (
            SCIP_PROBDATA * probdata )
```

5.20.2.6 GetExecutionSummary()

```
ExecutionSummary GetExecutionSummary (
            SCIP * scip )
```

5.20.2.7 GetParams()

```
Params* GetParams (
            SCIP_ProbData * probdata )
```

5.20.2.8 GetProblemData()

```
ProblemData* GetProblemData (
            SCIP_ProbData * probdata )
```

5.20.2.9 GetVehicleBranchingConstraints()

```
const vector<SCIP_CONS*>* GetVehicleBranchingConstraints (
            SCIP_PROBDATA * probdata )
```

5.20.2.10 IncrementUsedBranchingRule()

```
void IncrementUsedBranchingRule (
            SCIP_PROBDATA * probdata,
            int ruleIndex )
```

5.20.2.11 IsVarRepeated()

```
bool IsVarRepeated (
            SCIP_ProbData * probdata,
            Route * route,
            SCIP * scip )
```

5.20.2.12 loadProblem()

```
SCIP_RETCODE loadProblem (
            SCIP * scip,
            Params * params,
            ProblemData * problemData )
```

sets up SCIP execution for a given (my) ProblemData structure

Parameters

scip	SCIP data structure
params	global params
problemData	my problem's data structure

5.20.2.13 LogRepeatedRoute()

```
void LogRepeatedRoute (
            SCIP_PROBDATA * probdata,
            double reducedCost )
```

5.20.2.14 ncons()

```
static int ncons (
            ProblemData * problemData )  [static]
```

5.20.2.15 OutputDuals()

```
void OutputDuals (
            SCIP * scip,
            Params * params )
```

5.20.2.16 probdataCreate()

```
static SCIP_RETCODE probdataCreate (
            SCIP * scip,
            SCIP_PROBDATA ** probdata,
            SCIP_VAR ** vars,
            SCIP_CONS ** conss,
            int nvars,
            Params * params,
            ProblemData * problemData )  [static]
```

creates problem data

Parameters

scip	SCIP data structure
probdata	pointer to problem data
vars	array of ALL vars
conss	array of all constraints
nvars	number of route variables
params	global params
problemData	general data of this problem instance

5.20.2.17 probdataFree()

```
static SCIP_RETCODE probdataFree (
            SCIP * scip,
            SCIP_PROBDATA ** probdata )  [static]
```

frees the memory of the given problem data

Parameters

scip	SCIP data structure
probdata	pointer to problem data

5.20.2.18 QuerySolution()

```
void QuerySolution (
            SCIP * scip,
            ProblemSolution & solution )
```

5.20.2.19 SCIP_DECL_EVENTEXEC()

```
static SCIP_DECL_EVENTEXEC (
            eventExecAddedVar  )  [static]
```

execution method of event handler

5.20.2.20 SCIP_DECL_PROBDELORIG()

```
static SCIP_DECL_PROBDELORIG (
            probdelorigSPwCG  )  [static]
```

frees user data of original problem (called when the original problem is freed)

5.20.2.21 SCIP_DECL_PROBDELTRANS()

```
static SCIP_DECL_PROBDELTRANS (
            probdeltransSPwCG  )  [static]
```

frees user data of transformed problem (called when the transformed problem is freed)

5.20.2.22 SCIP_DECL_PROBEXITSOL()

```
static SCIP_DECL_PROBEXITSOL (
            probexitsolSPwCG  )  [static]
```

solving process deinitialization method of transformed data (called before the branch and bound data is freed)

5.20.2.23 SCIP_DECL_PROBINITSOL()

```
static SCIP_DECL_PROBINITSOL (
            probinitsolSPwCG  )  [static]
```

solving process initialization method of transformed data (called before the branch and bound process begins)

5.20.2.24 SCIP_DECL_PROBTRANS()

```
static SCIP_DECL_PROBTRANS (
            probtransSPwCG  )  [static]
```

creates user data of transformed problem by transforming the original user problem data (called after problem was transformed)

5.20.2.25 SCIPprobdataAddVar()

```
SCIP_RETCODE SCIPprobdataAddVar (
            SCIP * scip,
            SCIP_PROBDATA * probdata,
            SCIP_VAR * var )
```

adds given variable to the problem data

Parameters

scip	SCIP data structure
probdata	problem data
var	variables to add

5.20.2.26 SCIPprobdataGetCons()

```
SCIP_CONS** SCIPprobdataGetCons (
            SCIP_PROBDATA * probdata )
```

returns array of all variables itemed in the way they got generated

Parameters

probdata	problem data

5.20.2.27 SCIPprobdataGetNCons()

```
int SCIPprobdataGetNCons (
            SCIP_PROBDATA * probdata )
```

returns number of variables

Parameters

probdata	problem data

5.20.2.28 SCIPprobdataGetNVars()

```
int SCIPprobdataGetNVars (
            SCIP_PROBDATA * probdata )
```

returns number of variables

Parameters

probdata	problem data

5.20.2.29 SCIPprobdataGetVars()

```
SCIP_VAR** SCIPprobdataGetVars (
            SCIP_PROBDATA * probdata )
```

returns array of all variables itemed in the way they got generated

Parameters

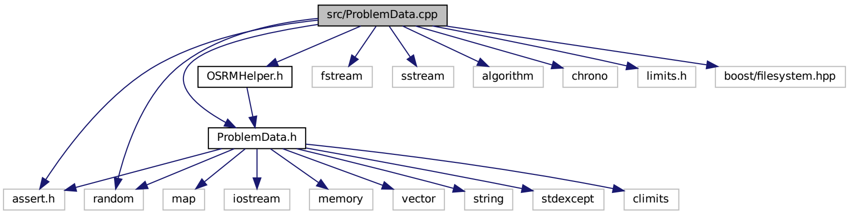
probdata	problem data

5.21 src/ProblemData.cpp File Reference

```
#include <assert.h>
```

```
#include <fstream>
#include <sstream>
#include <algorithm>
#include <chrono>
#include <random>
#include <limits.h>
#include "ProblemData.h"
#include "OSRMHelper.h"
#include <boost/filesystem.hpp>
```

Include dependency graph for ProblemData.cpp:



Macros

- #define RC_EPS 0.1

Functions

- static double norm (Position &a, Position &b)

5.21.1 Macro Definition Documentation

5.21.1.1 RC_EPS

```
#define RC_EPS 0.1
```

5.21.2 Function Documentation

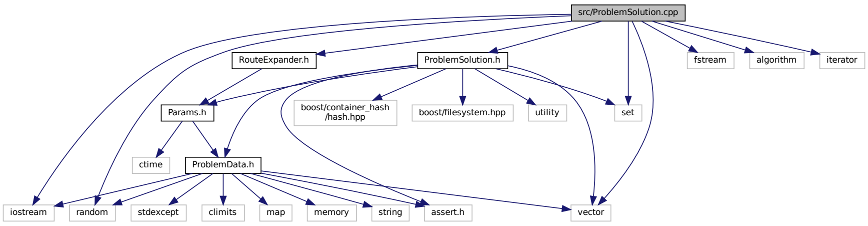
5.21.2.1 norm()

```
static double norm (
            Position & a,
            Position & b )  [static]
```

5.22 src/ProblemSolution.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <random>
#include <algorithm>
#include <iterator>
#include <vector>
#include "ProblemSolution.h"
#include "RouteExpander.h"
#include <set>
```

Include dependency graph for ProblemSolution.cpp:



Macros

- #define RC_EPS 0.1

5.22.1 Macro Definition Documentation

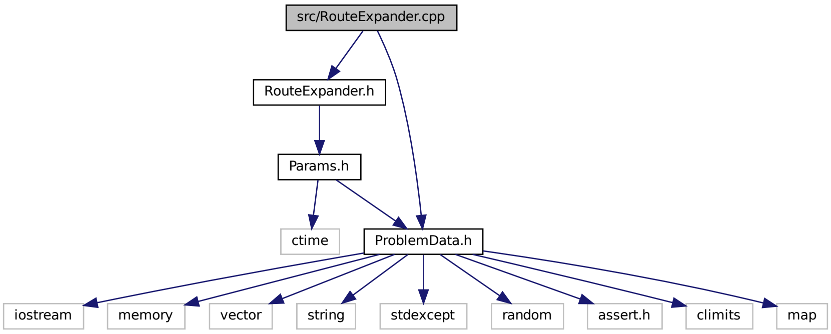
5.22.1.1 RC_EPS

```
#define RC_EPS 0.1
```

5.23 src/RouteExpander.cpp File Reference

```
#include "RouteExpander.h"
#include "ProblemData.h"
```

Include dependency graph for RouteExpander.cpp:



Macros

- #define RC_EPS 0.1

5.23.1 Macro Definition Documentation

5.23.1.1 RC_EPS

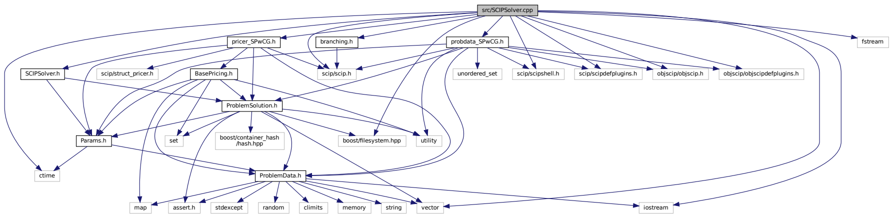
```
#define RC_EPS 0.1
```

5.24 src/SCIPSolver.cpp File Reference

```
#include <ctime>
#include <vector>
#include <iostream>
#include <fstream>
#include <boost/filesystem.hpp>
#include "SCIPSolver.h"
#include "scip/scip.h"
#include "scip/scipshell.h"
#include "scip/scipdefplugins.h"
#include "objscip/objscip.h"
#include "objscip/objscipdefplugins.h"
#include "branching.h"
#include "pricer_SPwCG.h"
```

```
#include "probdata_SPwCG.h"
```

Include dependency graph for SCIPSolver.cpp:



Macros

- #define GIT_COMMIT_HASH "?"

Functions

- static SCIP_RETCODE runSCIP (Params ∗params, ProblemData ∗problemData, ProblemSolution &solution)

5.24.1 Macro Definition Documentation

5.24.1.1 GIT_COMMIT_HASH

```
#define GIT_COMMIT_HASH "?"
```

5.24.2 Function Documentation

5.24.2.1 runSCIP()

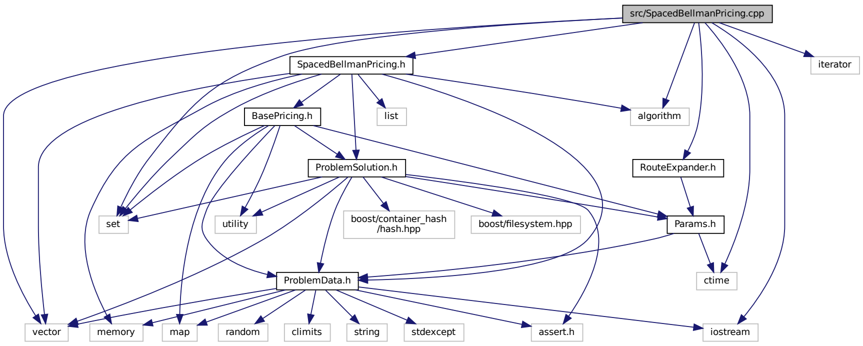
```
static SCIP_RETCODE runSCIP (
            Params * params,
            ProblemData * problemData,
            ProblemSolution & solution )  [static]
```

scip execution: creates a SCIP instance with default plugins, loads rules, callbacks etc.

5.25 src/SpacedBellmanPricing.cpp File Reference

```
#include "SpacedBellmanPricing.h"
#include "RouteExpander.h"
#include <iterator>
#include <ctime>
#include <vector>
#include <set>
#include <iostream>
#include <algorithm>
```

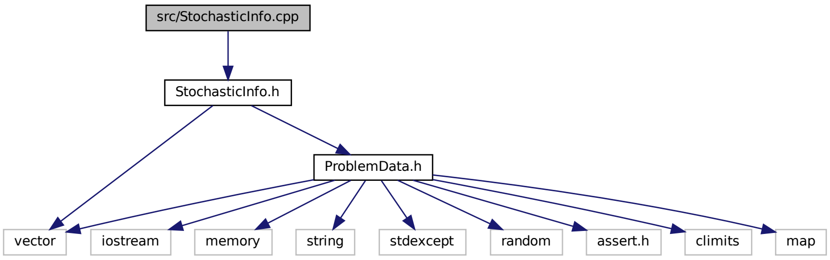
Include dependency graph for SpacedBellmanPricing.cpp:



5.26 src/StochasticInfo.cpp File Reference

```
#include "StochasticInfo.h"
```

Include dependency graph for StochasticInfo.cpp:

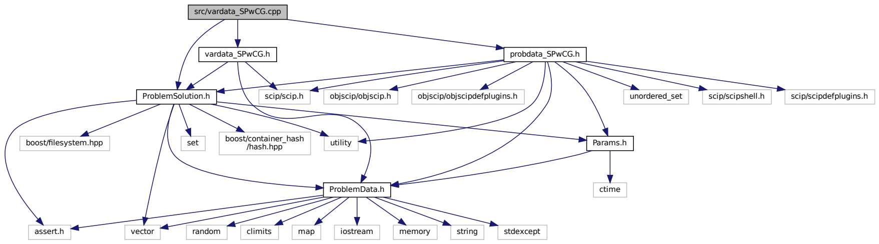


5.27 src/vardata_SPwCG.cpp File Reference

```
#include "ProblemSolution.h"
#include "probdata_SPwCG.h"
```

```
#include "vardata_SPwCG.h"
```

Include dependency graph for vardata_SPwCG.cpp:



Classes

- struct SCIP_VarData

Functions

Local methods

- static SCIP_RETCODE vardataCreate (SCIP ∗scip, SCIP_VARDATA ∗∗vardata, int ∗consids, int ∗conscoeffs, int nconsids, Route ∗route)
- static SCIP_RETCODE vardataDelete (SCIP ∗scip, SCIP_VARDATA ∗∗vardata)

Callback methods

- static SCIP_DECL_VARDELTRANS (vardataDelTrans)

Interface methods

- SCIP_RETCODE SCIPvardataCreateBinpacking (SCIP ∗scip, SCIP_VARDATA ∗∗vardata, int ∗consids, int ∗conscoeffs, int nconsids, Route ∗route)
- int SCIPvardataGetNConsids (SCIP_VARDATA ∗vardata)
- Route ∗ SCIPvardataGetRoute (SCIP_VARDATA ∗vardata)
- int ∗ SCIPvardataGetConsids (SCIP_VARDATA ∗vardata)
- int ∗ SCIPvardataGetConsCoeffs (SCIP_VARDATA ∗vardata)
- SCIP_RETCODE SCIPcreateVarBinpacking (SCIP ∗scip, SCIP_VAR ∗∗var, const char ∗name, SCIP_↩ Real obj, SCIP_Bool initial, SCIP_Bool removable, bool relaxed, SCIP_VARDATA ∗vardata)
- void SCIPvardataPrint (SCIP ∗scip, SCIP_VARDATA ∗vardata, FILE ∗file)

5.27.1 Function Documentation

5.27.1.1 SCIP_DECL_VARDELTRANS()

```
static SCIP_DECL_VARDELTRANS (
            vardataDelTrans )  [static]
```

frees user data of transformed variable (called when the transformed variable is freed)

5.27.1.2 SCIPcreateVarBinpacking()

```
SCIP_RETCODE SCIPcreateVarBinpacking (
             SCIP * scip,
             SCIP_VAR ** var,
             const char * name,
             SCIP_Real obj,
             SCIP_Bool initial,
             SCIP_Bool removable,
             bool relaxed,
             SCIP_VARDATA * vardata )
```

creates variable

Parameters

scip	SCIP data structure
var	pointer to variable object
name	name of variable, or NULL for automatic name creation
obj	objective function value
initial	should var's column be present in the initial root LP?
removable	is var's column removable from the LP (due to aging or cleanup)?
vardata	should the var be binary (false) or continuous (true) ? user data for this specific variable

5.27.1.3 SCIPvardataCreateBinpacking()

```
SCIP_RETCODE SCIPvardataCreateBinpacking (
             SCIP * scip,
             SCIP_VARDATA ** vardata,
             int * consids,
             int * conscoeffs,
             int nconsids,
             Route * route )
```

create variable data

Parameters

scip	SCIP data structure
vardata	pointer to vardata
consids	array of constraints ids
conscoeffs	array of contraint coefficients
nconsids	number of constraints

5.27.1.4 SCIPvardataGetConsCoeffs()

```
int* SCIPvardataGetConsCoeffs (
             SCIP_VARDATA * vardata )
```

returns constraint coeffs

Parameters

vardata	variable data

5.27.1.5 SCIPvardataGetConsids()

```
int* SCIPvardataGetConsids (
             SCIP_VARDATA * vardata )
```

returns sorted constraint id array

Parameters

vardata	variable data

5.27.1.6 SCIPvardataGetNConsids()

```
int SCIPvardataGetNConsids (
             SCIP_VARDATA * vardata )
```

get number of constraints

Parameters

vardata	variable data

5.27.1.7 SCIPvardataGetRoute()

```
Route* SCIPvardataGetRoute (
             SCIP_VARDATA * vardata )
```

returns route pointer

Parameters

vardata	variable data

5.27.1.8 SCIPvardataPrint()

```
void SCIPvardataPrint (
            SCIP * scip,
            SCIP_VARDATA * vardata,
            FILE * file )
```

prints vardata to file stream

Parameters

scip	SCIP data structure
vardata	variable data
file	the text file to store the information into

5.27.1.9 vardataCreate()

```
static SCIP_RETCODE vardataCreate (
            SCIP * scip,
            SCIP_VARDATA ** vardata,
            int * consids,
            int * conscoeffs,
            int nconsids,
            Route * route )  [static]
```

create a vardata

Parameters

scip	SCIP data structure
vardata	pointer to vardata
consids	array of constraints ids
conscoeffs	array of constraints coefficients
nconsids	number of constraints

5.27.1.10 vardataDelete()

```
static SCIP_RETCODE vardataDelete (
            SCIP * scip,
            SCIP_VARDATA ** vardata )  [static]
```

frees user data of variable

Parameters

scip	SCIP data structure
vardata	vardata to delete

Index