

BiCIAM

1.0

Generado el Viernes, 5 de diciembre de 2025 20:47:18 para BiCIAM por Doxygen 1.14.0

Viernes, 5 de diciembre de 2025 20:47:18

Capítulo 1

Índice de espacios de nombres

1.1. Lista de paquetes

Estos son los paquetes con breves descripciones (si están disponibles):

config.tspDynamic	??
evolutionary_algorithms.complement	??
factory_interface	
@(#) IFFactoryAcceptCandidate.java	??
factory_method	
@(#) FactoryAcceptCandidate.java	??
local_search	??
local_search.acceptation_type	
@(#) AcceptAnyone.java	??
local_search.candidate_type	
@(#) TypeCandidate.java	??
local_search.complement	
@(#) Strategy.java	??
metaheuristics.generators	??
metaheuristics.strategy	??
problem.definition	??
problem.extension	??
problem_operators	??

Capítulo 2

Índice jerárquico

2.1. Jerarquía de clases

Este listado de herencia está ordenado de forma general pero no está en orden alfabético estricto:

local_search.acception_type.AcceptableCandidate	??
local_search.acception_type.AcceptAnyone	??
local_search.acception_type.AcceptBest	??
local_search.acception_type.AcceptMulticase	??
local_search.acception_type.AcceptNotBad	??
local_search.acception_type.AcceptNotBadT	??
local_search.acception_type.AcceptNotBadU	??
local_search.acception_type.AcceptNotDominated	??
local_search.acception_type.AcceptNotDominatedTabu	??
local_search.acception_type.AcceptType	??
local_search.candidate_type.CandidateType	??
local_search.candidate_type.CandidateValue	??
Cloneable	
problem.definition.State	??
problem.definition.Codification	??
evolutionary_algorithms.complement.Crossover	??
evolutionary_algorithms.complement.OnePointCrossover	??
evolutionary_algorithms.complement.UniformCrossover	??
evolutionary_algorithms.complement.CrossoverType	??
evolutionary_algorithms.complement.Distribution	??
evolutionary_algorithms.complement.Univariate	??
evolutionary_algorithms.complement.DistributionType	??
local_search.acception_type.Dominance	??
factory_method.FactoryLoader	??
evolutionary_algorithms.complement.FatherSelection	??
evolutionary_algorithms.complement.RouletteSelection	??
evolutionary_algorithms.complement.TruncationSelection	??
metaheuristics.generators.Generator	??
local_search.MultiCaseSimulatedAnnealing	??
local_search.complement.MultiCaseSimulatedAnnealing	??
metaheuristics.generators.DistributionEstimationAlgorithm	??
metaheuristics.generators.EvolutionStrategies	??
metaheuristics.generators.GeneticAlgorithm	??
metaheuristics.generators.HillClimbing	??

metaheuristics.generators.HillClimbingRestart	??
metaheuristics.generators.LimitThreshold	??
metaheuristics.generators.MultiCaseSimulatedAnnealing	??
metaheuristics.generators.MultiGenerator	??
metaheuristics.generators.MutiobjectiveHillClimbingDistance	??
metaheuristics.generators.MutiobjectiveHillClimbingRestart	??
metaheuristics.generators.MutiobjectiveStochasticHillClimbing	??
metaheuristics.generators.MutiobjectiveTabuSearch	??
metaheuristics.generators.Particle	??
metaheuristics.generators.ParticleSwarmOptimization	??
metaheuristics.generators.RandomSearch	??
metaheuristics.generators.SimulatedAnnealing	??
metaheuristics.generators.TabuSearch	??
metaheuristics.generators.GeneratorType	??
factory_interface.IFFactoryAcceptCandidate	??
factory_method.FactoryAcceptCandidate	??
factory_interface.IFFactoryCandidate	??
factory_method.FactoryCandidate	??
factory_interface.IFFactoryCrossover	??
factory_method.FactoryCrossover	??
factory_interface.IFFactoryDistribution	??
factory_method.FactoryDistribution	??
factory_interface.IFFactoryFatherSelection	??
factory_method.FactoryFatherSelection	??
factory_interface.IFFactoryGenerator	??
factory_method.FactoryGenerator	??
factory_interface.IFFactoryMutation	??
factory_method.FactoryMutation	??
factory_interface.IFFactoryReplace	??
factory_method.FactoryReplace	??
factory_interface.IFFactorySolutionMethod	??
factory_method.FactorySolutionMethod	??
factory_interface.IFFSampling	??
factory_method.FactorySampling	??
metaheuristics.generators.LimitRoulette	??
problem.extension.MetricasMultiobjetivo	??
evolutionary_algorithms.complement.Mutation	??
evolutionary_algorithms.complement.AIOMutation	??
evolutionary_algorithms.complement.OnePointMutation	??
evolutionary_algorithms.complement.TowPointsMutation	??
evolutionary_algorithms.complement.MutationType	??
metaheuristics.generators.MyTestClass	??
problem.definition.ObjetiveFunction	??
problem.definition.Operator	??
problem_operators.MutationOperator	??
evolutionary_algorithms.complement.Probability	??
problem.definition.Problem	??
problem.definition.Problem.ProblemType	??
evolutionary_algorithms.complement.Range	??
evolutionary_algorithms.complement.Replace	??
evolutionary_algorithms.complement.GenerationalReplace	??
evolutionary_algorithms.complement.SteadyStateReplace	??
evolutionary_algorithms.complement.ReplaceType	??
Runnable	

metaheuristics.generators.InstanceDE	??
metaheuristics.generators.InstanceEE	??
metaheuristics.generators.InstanceGA	??
evolutionary_algorithms.complement.Sampling	??
evolutionary_algorithms.complement.ProbabilisticSampling	??
evolutionary_algorithms.complement.SamplingType	??
local_search.candidate_type.SearchCandidate	??
local_search.candidate_type.GreaterCandidate	??
local_search.candidate_type.NotDominatedCandidate	??
local_search.candidate_type.RandomCandidate	??
local_search.candidate_type.SmallerCandidate	??
evolutionary_algorithms.complement.SelectionType	??
metaheuristics.generators.SimpleTestClass	??
problem.extension.SolutionMethod	??
problem.extension.FactoresPonderados	??
problem.extension.MultiObjetivoPuro	??
local_search.complement.StopExecute	??
metaheuristics.strategy.Strategy	??
local_search.complement.StrATEGYType	??
local_search.complement.TabuSolutions	??
local_search.TestDummy	??
config.tspDynamic.TSPState	??
problem.extension.TypeSolutionMethod	??
local_search.complement.UpdateParameter	??

Capítulo 3

Índice de clases

3.1. Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

local_search.acception_type.AcceptableCandidate	??
local_search.acception_type.AcceptAnyone	??
local_search.acception_type.AcceptBest	??
local_search.acception_type.AcceptMulticase	??
Implementa la lógica de aceptación de candidatos para el algoritmo de Recocido Simulado Multicase	??
local_search.acception_type.AcceptNotBad	??
local_search.acception_type.AcceptNotBadT	??
local_search.acception_type.AcceptNotBadU	??
local_search.acception_type.AcceptNotDominated	??
local_search.acception_type.AcceptNotDominatedTabu	??
local_search.acception_type.AcceptType	??
evolutionary_algorithms.complement.AIOMutation	??
local_search.candidate_type.CandidateType	??
local_search.candidate_type.CandidateValue	??
problem.definition.Codification	??
evolutionary_algorithms.complement.Crossover	??
evolutionary_algorithms.complement.CrossoverType	??
evolutionary_algorithms.complement.Distribution	??
metaheuristics.generators.DistributionEstimationAlgorithm	??
evolutionary_algorithms.complement.DistributionType	??
local_search.acception_type.Dominance	??
metaheuristics.generators.EvolutionStrategies	??
problem.extension.FactoresPonderados	??
factory_method.FactoryAcceptCandidate	??
factory_method.FactoryCandidate	??
factory_method.FactoryCrossover	??
factory_method.FactoryDistribution	??
factory_method.FactoryFatherSelection	??
factory_method.FactoryGenerator	??
factory_method.FactoryLoader	??
factory_method.FactoryMutation	??
factory_method.FactoryReplace	??
factory_method.FactorySampling	??
factory_method.FactorySolutionMethod	??

evolutionary_algorithms.complement.FatherSelection	??
evolutionary_algorithms.complement.GenerationalReplace	??
metaheuristics.generators.Generator	??
metaheuristics.generators.GeneratorType	??
metaheuristics.generators.GeneticAlgorithm	??
local_search.candidate_type.GreaterCandidate	??
metaheuristics.generators.HillClimbing	??
metaheuristics.generators.HillClimbingRestart	??
factory_interface.IFFactoryAcceptCandidate	??
factory_interface.IFFactoryCandidate	??
factory_interface.IFFactoryCrossover	??
factory_interface.IFFactoryDistribution	??
factory_interface.IFFactoryFatherSelection	??
factory_interface.IFFactoryGenerator	??
factory_interface.IFFactoryMutation	??
factory_interface.IFFactoryReplace	??
factory_interface.IFFactorySolutionMethod	??
factory_interface.IFFSampling	??
metaheuristics.generators.InstanceDE	??
metaheuristics.generators.InstanceEE	??
metaheuristics.generators.InstanceGA	??
metaheuristics.generators.LimitRoulette	??
metaheuristics.generators.LimitThreshold	??
problem.extension.MetricasMultiobjetivo	??
local_search.complement.MultiCaseSimulatedAnnealing	??
local_search.MultiCaseSimulatedAnnealing	??
metaheuristics.generators.MultiCaseSimulatedAnnealing	??
metaheuristics.generators.MultiGenerator	??
metaheuristics.generators.MultiobjectiveHillClimbingDistance	??
metaheuristics.generators.MultiobjectiveHillClimbingRestart	??
metaheuristics.generators.MultiobjectiveStochasticHillClimbing	??
metaheuristics.generators.MultiobjectiveTabuSearch	??
problem.extension.MultiObjetivoPuro	??
evolutionary_algorithms.complement.Mutation	??
problem_operators.MutationOperator	??
evolutionary_algorithms.complement.MutationType	??
metaheuristics.generators.MyTestClass	??
local_search.candidate_type.NotDominatedCandidate	??
problem.definition.ObjetiveFunction	??
evolutionary_algorithms.complement.OnePointCrossover	??
evolutionary_algorithms.complement.OnePointMutation	??
problem.definition.Operator	??
metaheuristics.generators.Particle	
Representa una partícula en el algoritmo de Optimización por Enjambre de Partículas (PSO)	??
metaheuristics.generators.ParticleSwarmOptimization	
Implementación del algoritmo de Optimización por Enjambre de Partículas (PSO)	??
evolutionary_algorithms.complement.ProbabilisticSampling	??
evolutionary_algorithms.complement.Probability	??
problem.definition.Problem	??
problem.definition.ProblemType	??
local_search.candidate_type.RandomCandidate	??
metaheuristics.generators.RandomSearch	??
evolutionary_algorithms.complement.Range	??
evolutionary_algorithms.complement.Replace	??
evolutionary_algorithms.complement.ReplaceType	??
evolutionary_algorithms.complement.RouletteSelection	??
evolutionary_algorithms.complement.Sampling	??
evolutionary_algorithms.complement.SamplingType	??

local_search.candidate_type.SearchCandidate	??
evolutionary_algorithms.complement.SelectionType	??
metaheuristics.generators.SimpleTestClass	??
metaheuristics.generators.SimulatedAnnealing	??
local_search.candidate_type.SmallerCandidate	??
problem.extension.SolutionMethod	??
problem.definition.State	??
evolutionary_algorithms.complement.SteadyStateReplace	??
local_search.complement.StopExecute	??
metaheuristics.strategy.Strategy	??
local_search.complement.StrATEGYType	??
metaheuristics.generators.TabuSearch	??
local_search.complement.TabuSolutions	??
local_search.TestDummy	??
evolutionary_algorithms.complement.TowPointsMutation	??
evolutionary_algorithms.complement.TruncationSelection	??
config.tspDynamic.TSPState	??
problem.extension.TypeSolutionMethod	??
evolutionary_algorithms.complement.UniformCrossover	??
evolutionary_algorithms.complement.Univariate	??
local_search.complement.UpdateParameter	??

Capítulo 4

Índice de archivos

4.1. Lista de archivos

Lista de todos los archivos con breves descripciones:

src/main/java/config/tspDynamic/ TSPState.java	??
src/main/java/evolutionary_algorithms/complement/ AIOMutation.java	??
src/main/java/evolutionary_algorithms/complement/ Crossover.java	??
src/main/java/evolutionary_algorithms/complement/ CrossoverType.java	??
src/main/java/evolutionary_algorithms/complement/ Distribution.java	??
src/main/java/evolutionary_algorithms/complement/ DistributionType.java	??
src/main/java/evolutionary_algorithms/complement/ FatherSelection.java	??
src/main/java/evolutionary_algorithms/complement/ GenerationalReplace.java	??
src/main/java/evolutionary_algorithms/complement/ Mutation.java	??
src/main/java/evolutionary_algorithms/complement/ MutationType.java	??
src/main/java/evolutionary_algorithms/complement/ OnePointCrossover.java	??
src/main/java/evolutionary_algorithms/complement/ OnePointMutation.java	??
src/main/java/evolutionary_algorithms/complement/ ProbabilisticSampling.java	??
src/main/java/evolutionary_algorithms/complement/ Probability.java	??
src/main/java/evolutionary_algorithms/complement/ Range.java	??
src/main/java/evolutionary_algorithms/complement/ Replace.java	??
src/main/java/evolutionary_algorithms/complement/ ReplaceType.java	??
src/main/java/evolutionary_algorithms/complement/ RouletteSelection.java	??
src/main/java/evolutionary_algorithms/complement/ Sampling.java	??
src/main/java/evolutionary_algorithms/complement/ SamplingType.java	??
src/main/java/evolutionary_algorithms/complement/ SelectionType.java	??
src/main/java/evolutionary_algorithms/complement/ SteadyStateReplace.java	??
src/main/java/evolutionary_algorithms/complement/ TowPointsMutation.java	??
src/main/java/evolutionary_algorithms/complement/ TruncationSelection.java	??
src/main/java/evolutionary_algorithms/complement/ UniformCrossover.java	??
src/main/java/evolutionary_algorithms/complement/ Univariate.java	??
src/main/java/factory_interface/ IFFFactoryAcceptCandidate.java	??
src/main/java/factory_interface/ IFFFactoryCandidate.java	??
src/main/java/factory_interface/ IFFFactoryCrossover.java	??
src/main/java/factory_interface/ IFFFactoryDistribution.java	??
src/main/java/factory_interface/ IFFFactoryFatherSelection.java	??
src/main/java/factory_interface/ IFFFactoryGenerator.java	??
src/main/java/factory_interface/ IFFFactoryMutation.java	??
src/main/java/factory_interface/ IFFFactoryReplace.java	??
src/main/java/factory_interface/ IFFFactorySolutionMethod.java	??

src/main/java/factory_interface/IFFSampling.java	??
src/main/java/factory_method/FactoryAcceptCandidate.java	??
src/main/java/factory_method/FactoryCandidate.java	??
src/main/java/factory_method/FactoryCrossover.java	??
src/main/java/factory_method/FactoryDistribution.java	??
src/main/java/factory_method/FactoryFatherSelection.java	??
src/main/java/factory_method/FactoryGenerator.java	??
src/main/java/factory_method/FactoryLoader.java	??
src/main/java/factory_method/FactoryMutation.java	??
src/main/java/factory_method/FactoryReplace.java	??
src/main/java/factory_method/FactorySampling.java	??
src/main/java/factory_method/FactorySolutionMethod.java	??
src/main/java/local_search/MultiCaseSimulatedAnnealing.java	??
src/main/java/local_search/TestDummy.java	??
src/main/java/local_search/acceptation_type/AcceptableCandidate.java	??
src/main/java/local_search/acceptation_type/AcceptAnyone.java	??
src/main/java/local_search/acceptation_type/AcceptBest.java	??
src/main/java/local_search/acceptation_type/AcceptMulticase.java	??
src/main/java/local_search/acceptation_type/AcceptNotBad.java	??
src/main/java/local_search/acceptation_type/AcceptNotBadT.java	??
src/main/java/local_search/acceptation_type/AcceptNotBadU.java	??
src/main/java/local_search/acceptation_type/AcceptNotDominated.java	??
src/main/java/local_search/acceptation_type/AcceptNotDominatedTabu.java	??
src/main/java/local_search/acceptation_type/AcceptType.java	??
src/main/java/local_search/acceptation_type/Dominance.java	??
src/main/java/local_search/candidate_type/CandidateType.java	??
src/main/java/local_search/candidate_type/CandidateValue.java	??
src/main/java/local_search/candidate_type/GreaterCandidate.java	??
src/main/java/local_search/candidate_type/NotDominatedCandidate.java	??
src/main/java/local_search/candidate_type/RandomCandidate.java	??
src/main/java/local_search/candidate_type/SearchCandidate.java	??
src/main/java/local_search/candidate_type/SmallerCandidate.java	??
src/main/java/local_search/complement/MultiCaseSimulatedAnnealing.java	??
src/main/java/local_search/complement/StopExecute.java	??
src/main/java/local_search/complement/StrategyType.java	??
src/main/java/local_search/complement/TabuSolutions.java	??
src/main/java/local_search/complement/UpdateParameter.java	??
src/main/java/metaheuristics/generators/DistributionEstimationAlgorithm.java	??
src/main/java/metaheuristics/generators/EvolutionStrategies.java	??
src/main/java/metaheuristics/generators/Generator.java	??
src/main/java/metaheuristics/generators/GeneratorType.java	??
src/main/java/metaheuristics/generators/GeneticAlgorithm.java	??
src/main/java/metaheuristics/generators/HillClimbing.java	??
src/main/java/metaheuristics/generators/HillClimbingRestart.java	??
src/main/java/metaheuristics/generators/InstanceDE.java	??
src/main/java/metaheuristics/generators/InstanceEE.java	??
src/main/java/metaheuristics/generators/InstanceGA.java	??
src/main/java/metaheuristics/generators/LimitRoulette.java	??
src/main/java/metaheuristics/generators/LimitThreshold.java	??
src/main/java/metaheuristics/generators/MultiCaseSimulatedAnnealing.java	??
src/main/java/metaheuristics/generators/MultiGenerator.java	??
src/main/java/metaheuristics/generators/MultiobjectiveHillClimbingDistance.java	??
src/main/java/metaheuristics/generators/MultiobjectiveHillClimbingRestart.java	??
src/main/java/metaheuristics/generators/MultiobjectiveStochasticHillClimbing.java	??
src/main/java/metaheuristics/generators/MultiobjectiveTabuSearch.java	??
src/main/java/metaheuristics/generators/MyTestClass.java	??
src/main/java/metaheuristics/generators/Particle.java	??
src/main/java/metaheuristics/generators/ParticleSwarmOptimization.java	??

src/main/java/metaheuristics/generators/ RandomSearch.java	??
src/main/java/metaheuristics/generators/ SimpleTestClass.java	??
src/main/java/metaheuristics/generators/ SimulatedAnnealing.java	??
src/main/java/metaheuristics/generators/ TabuSearch.java	??
src/main/java/metaheuristics/strategy/ Strategy.java	??
src/main/java/problem/definition/ Codification.java	??
src/main/java/problem/definition/ ObjetiveFunction.java	??
src/main/java/problem/definition/ Operator.java	??
src/main/java/problem/definition/ Problem.java	??
src/main/java/problem/definition/ State.java	??
src/main/java/problem/extension/ FactoresPonderados.java	??
src/main/java/problem/extension/ MetricasMultiobjetivo.java	??
src/main/java/problem/extension/ MultiObjetivoPuro.java	??
src/main/java/problem/extension/ SolutionMethod.java	??
src/main/java/problem/extension/ TypeSolutionMethod.java	??
src/main/java/problem_operators/ MutationOperator.java	??

Capítulo 5

Documentación de espacios de nombres

5.1. Paquete config.tspDynamic

Clases

- class [TSPState](#)

5.2. Paquete evolutionary_algorithms.complement

Clases

- class [AIOMutation](#)
- class [Crossover](#)
- enum [CrossoverType](#)
- class [Distribution](#)
- enum [DistributionType](#)
- class [FatherSelection](#)
- class [GenerationalReplace](#)
- class [Mutation](#)
- enum [MutationType](#)
- class [OnePointCrossover](#)
- class [OnePointMutation](#)
- class [ProbabilisticSampling](#)
- class [Probability](#)
- class [Range](#)
- class [Replace](#)
- enum [ReplaceType](#)
- class [RouletteSelection](#)
- class [Sampling](#)
- enum [SamplingType](#)
- enum [SelectionType](#)
- class [SteadyStateReplace](#)
- class [TowPointsMutation](#)
- class [TruncationSelection](#)
- class [UniformCrossover](#)
- class [Univariate](#)

5.3. Paquete factory_interface

@(#)[IFFactoryAcceptCandidate.java](#)

Clases

- interface [IFFactoryAcceptCandidate](#)
- interface [IFFactoryCandidate](#)
- interface [IFFactoryCrossover](#)
- interface [IFFactoryDistribution](#)
- interface [IFFactoryFatherSelection](#)
- interface [IFFactoryGenerator](#)
- interface [IFFactoryMutation](#)
- interface [IFFactoryReplace](#)
- interface [IFFactorySolutionMethod](#)
- interface [IFFSampling](#)

5.3.1. Descripción detallada

@(#)[IFFactoryAcceptCandidate.java](#)

@(#)[IFFactoryCandidate.java](#)

5.4. Paquete factory_method

@(#)[FactoryAcceptCandidate.java](#)

Clases

- class [FactoryAcceptCandidate](#)
- class [FactoryCandidate](#)
- class [FactoryCrossover](#)
- class [FactoryDistribution](#)
- class [FactoryFatherSelection](#)
- class [FactoryGenerator](#)
- class [FactoryLoader](#)
- class [FactoryMutation](#)
- class [FactoryReplace](#)
- class [FactorySampling](#)
- class [FactorySolutionMethod](#)

5.4.1. Descripción detallada

@(#)[FactoryAcceptCandidate.java](#)

@(#)[FactoryCandidate.java](#)

5.5. Paquete local_search

Paquetes

- package [acceptation_type](#)
 @(#) [AcceptAnyone.java](#)
- package [candidate_type](#)
 @(#) [TypeCandidate.java](#)
- package [complement](#)
 @(#) [Strategy.java](#)

Clases

- class [MultiCaseSimulatedAnnealing](#)
- class [TestDummy](#)

5.6. Paquete local_search.acceptation_type

@(#) [AcceptAnyone.java](#)

Clases

- class [AcceptableCandidate](#)
- class [AcceptAnyone](#)
- class [AcceptBest](#)
- class [AcceptMulticase](#)
Implementa la lógica de aceptación de candidatos para el algoritmo de Recocido Simulado Multicaso.
- class [AcceptNotBad](#)
- class [AcceptNotBadT](#)
- class [AcceptNotBadU](#)
- class [AcceptNotDominated](#)
- class [AcceptNotDominatedTabu](#)
- enum [AcceptType](#)
- class [Dominance](#)

5.6.1. Descripción detallada

@(#) [AcceptAnyone.java](#)

@(#) [TypeAcceptation.java](#)

@(#) [AcceptNoBadU.java](#)

@(#) [AcceptNoBadT.java](#)

@(#) [AcceptNoBad.java](#)

@(#) [AcceptBest.java](#)

5.7. Paquete local_search.candidate_type

@(#) [TypeCandidate.java](#)

Clases

- enum [CandidateType](#)
- class [CandidateValue](#)
- class [GreaterCandidate](#)
- class [NotDominatedCandidate](#)
- class [RandomCandidate](#)
- class [SearchCandidate](#)
- class [SmallerCandidate](#)

5.7.1. Descripción detallada

@(#) [TypeCandidate.java](#)

@(#) [SmallerCandidate.java](#)

@(#) [SearchCandidate.java](#)

@(#) [AleatoryCandidate.java](#)

@(#) [MajorCandidate.java](#)

@(#) [CandidateValue.java](#)

5.8. Paquete local_search.complement

@(#) [Strategy.java](#)

Clases

- class [MultiCaseSimulatedAnnealing](#)
- class [StopExecute](#)
- enum [StrategyType](#)
- class [TabuSolutions](#)
- class [UpdateParameter](#)

5.8.1. Descripción detallada

@(#) [Strategy.java](#)

5.9. Paquete metaheuristics.generators

Clases

- class [DistributionEstimationAlgorithm](#)
- class [EvolutionStrategies](#)
- class [Generator](#)
- enum [GeneratorType](#)
- class [GeneticAlgorithm](#)
- class [HillClimbing](#)
- class [HillClimbingRestart](#)
- class [InstanceDE](#)
- class [InstanceEE](#)
- class [InstanceGA](#)
- class [LimitRoulette](#)
- class [LimitThreshold](#)
- class [MultiCaseSimulatedAnnealing](#)
- class [MultiGenerator](#)
- class [MultiobjectiveHillClimbingDistance](#)
- class [MultiobjectiveHillClimbingRestart](#)
- class [MultiobjectiveStochasticHillClimbing](#)
- class [MultiobjectiveTabuSearch](#)
- class [MyTestClass](#)
- class [Particle](#)

Representa una partícula en el algoritmo de Optimización por Enjambre de Partículas (PSO).

- class [ParticleSwarmOptimization](#)

Implementación del algoritmo de Optimización por Enjambre de Partículas (PSO).

- class [RandomSearch](#)
- class [SimpleTestClass](#)
- class [SimulatedAnnealing](#)
- class [TabuSearch](#)

5.10. Paquete metaheuristics.strategy

Clases

- class [Strategy](#)

5.11. Paquete problem.definition

Clases

- class [Codification](#)
- class [ObjetiveFunction](#)
- class [Operator](#)
- class [Problem](#)
- class [State](#)

5.12. Paquete problem.extension

Clases

- class [FactoresPonderados](#)
- class [MetricasMultiobjetivo](#)
- class [MultiObjetivoPuro](#)
- class [SolutionMethod](#)
- enum [TypeSolutionMethod](#)

5.13. Paquete problem_operators

Clases

- class [MutationOperator](#)

Capítulo 6

Documentación de clases

6.1. Referencia de la clase

`local_search.acceptation_type.AcceptableCandidate`

Diagrama de herencia de `local_search.acceptation_type.AcceptableCandidate`

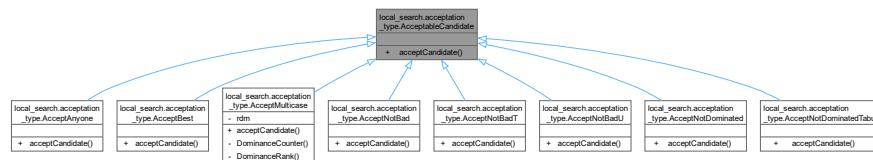
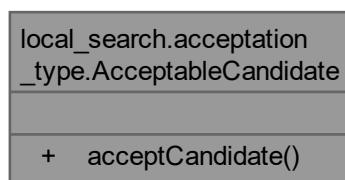


Diagrama de colaboración de `local_search.acceptation_type.AcceptableCandidate`:



Métodos públicos

- abstract Boolean `acceptCandidate (State stateCurrent, State stateCandidate)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

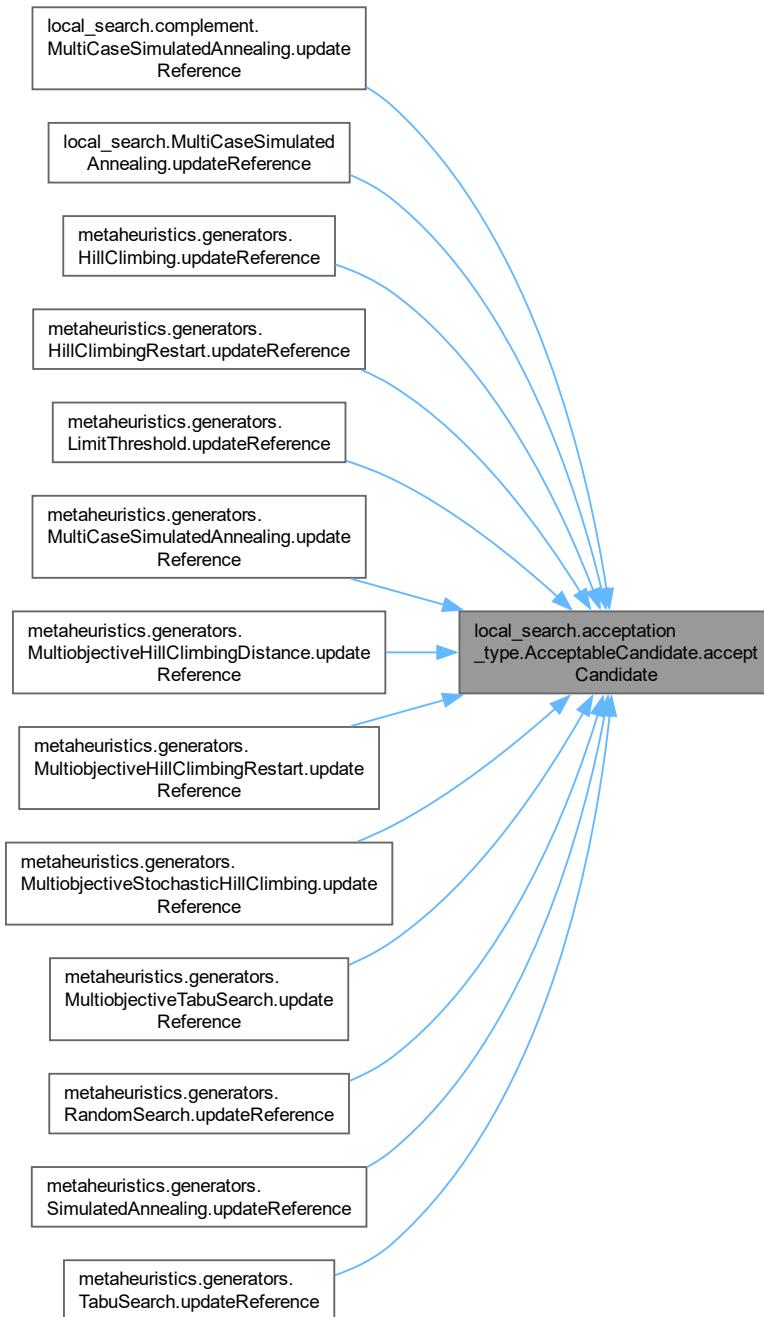
6.1.1. Documentación de funciones miembro

6.1.1.1. acceptCandidate()

```
abstract Boolean local_search.acceptation_type.AcceptableCandidate.acceptCandidate (
    State stateCurrent,
    State stateCandidate) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException [abstract]
```

Reimplementado en [local_search.acceptation_type.AcceptAnyone](#), [local_search.acceptation_type.AcceptBest](#),
[local_search.acceptation_type.AcceptMulticase](#), [local_search.acceptation_type.AcceptNotBad](#), [local_search.acceptation_type.AcceptNotBadU](#), [local_search.acceptation_type.AcceptNotDominated](#) y [local_search.acceptation_type.AcceptNotWorst](#).

Gráfico de llamadas a esta función:



La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/local_search/acceptation_type/AcceptableCandidate.java`

6.2. Referencia de la clase local_search.acceptation_type.AcceptAnyone

Diagrama de herencia de local_search.acceptation_type.AcceptAnyone

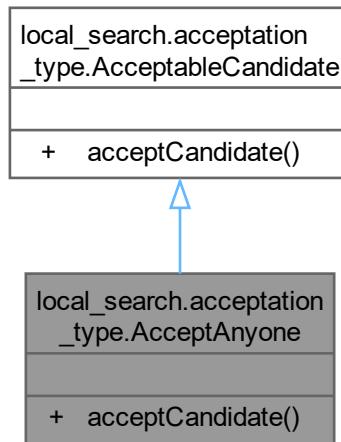
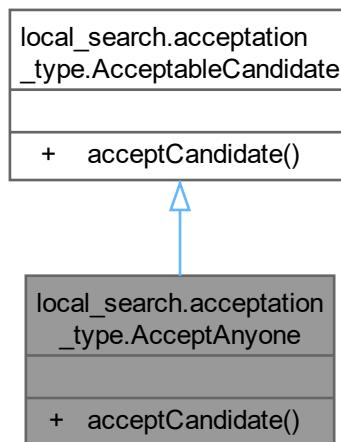


Diagrama de colaboración de local_search.acceptation_type.AcceptAnyone:



Métodos públicos

- Boolean [acceptCandidate](#) ([State](#) stateCurrent, [State](#) stateCandidate)

6.2.1. Documentación de funciones miembro

6.2.1.1. acceptCandidate()

```
Boolean local_search.acceptation_type.AcceptAnyone.acceptCandidate (
    State stateCurrent,
    State stateCandidate)
```

Reimplementado de [local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/acceptation_type/AcceptAnyone.java](#)

6.3. Referencia de la clase local_search.acceptation_type.AcceptBest

Diagrama de herencia de local_search.acceptation_type.AcceptBest

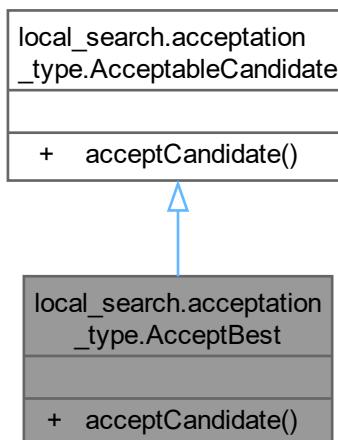
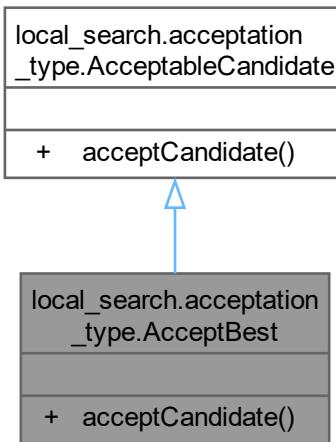


Diagrama de colaboración de local_search.acceptation_type.AcceptBest:



Métodos públicos

- Boolean `acceptCandidate (State stateCurrent, State stateCandidate)` throws `IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.3.1. Documentación de funciones miembro

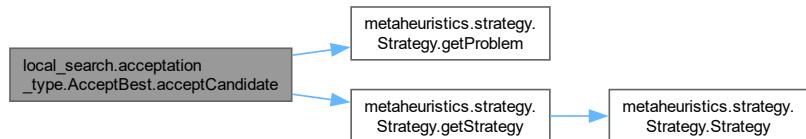
6.3.1.1. acceptCandidate()

```

Boolean local_search.acceptation_type.AcceptBest.acceptCandidate (
    State stateCurrent,
    State stateCandidate) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
  
```

Reimplementado de [local_search.acceptation_type.AcceptableCandidate](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/AcceptBest.java

6.4. Referencia de la clase

local_search.acceptation_type.AcceptMulticase

Implementa la lógica de aceptación de candidatos para el algoritmo de Recocido Simulado Multicaso.

Diagrama de herencia de local_search.acceptation_type.AcceptMulticase

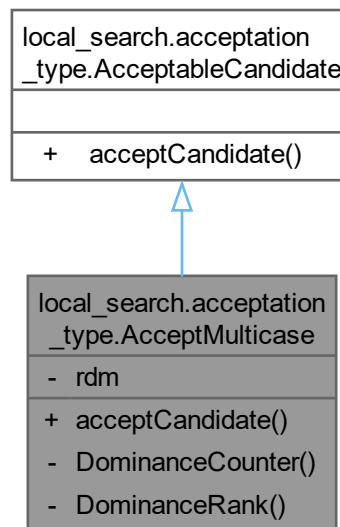
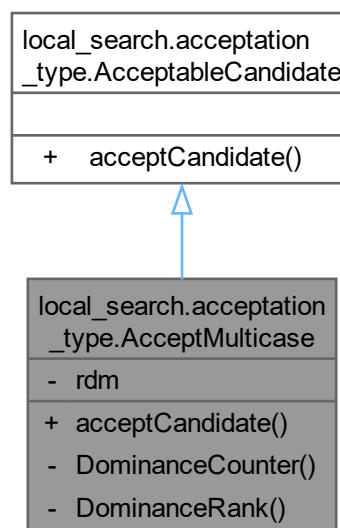


Diagrama de colaboración de local_search.acceptation_type.AcceptMulticase:



Métodos públicos

- Boolean `acceptCandidate (State stateCurrent, State stateCandidate)`
Determina si se acepta una solución candidata.

Métodos privados

- int `DominanceCounter (State stateCandidate, List< State > list)`
Cuenta cuántas soluciones en la lista son dominadas por el candidato.
- int `DominanceRank (State stateCandidate, List< State > list)`
Calcula el rango de dominancia del candidato.

Atributos estáticos privados

- static final Random `rdm = new Random()`
Generador de números aleatorios compartido.

6.4.1. Descripción detallada

Implementa la lógica de aceptación de candidatos para el algoritmo de Recocido Simulado Multicaso.

Esta clase extiende de `AcceptableCandidate` y define las reglas para aceptar o rechazar una solución candidata basándose en criterios de dominancia y probabilidad.

6.4.2. Documentación de funciones miembro

6.4.2.1. acceptCandidate()

```
Boolean local_search.acceptation_type.AcceptMulticase.acceptCandidate (
    State stateCurrent,
    State stateCandidate)
```

Determina si se acepta una solución candidata.

Evalúa la solución candidata frente a la solución actual utilizando criterios de dominancia y una función de probabilidad basada en la temperatura actual del recocido simulado.

Parámetros

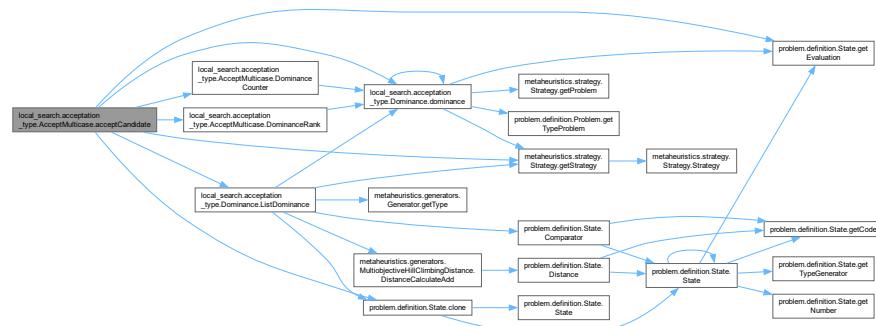
<code>stateCurrent</code>	Estado actual de la solución.
<code>stateCandidate</code>	Estado candidato a ser evaluado.

Devuelve

true si el candidato es aceptado, false en caso contrario.

Reimplementado de [local_search.acceptation_type.AcceptableCandidate](#).

Gráfico de llamadas de esta función:



6.4.2.2. DominanceCounter()

```
int local_search.acceptation_type.AcceptMulticase.DominanceCounter (
    State stateCandidate,
    List< State > list) [private]
```

Cuenta cuántas soluciones en la lista son dominadas por el candidato.

Parámetros

<code>stateCandidate</code>	Estado candidato a evaluar.
<code>list</code>	Lista de soluciones de referencia (ej. frente de Pareto actual).

Devuelve

Número de soluciones en la lista que son dominadas por el candidato.

Gráfico de llamadas de esta función:

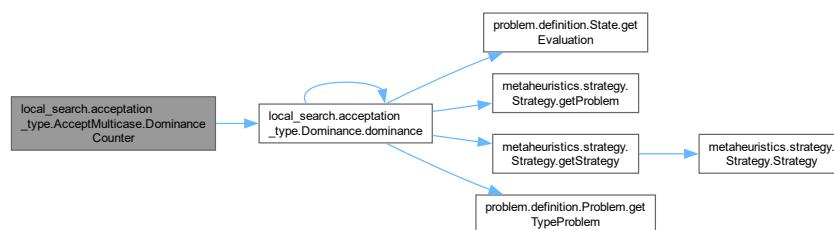
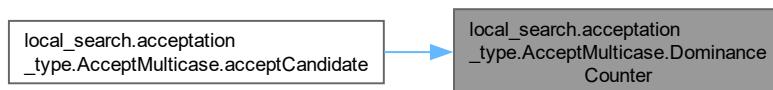


Gráfico de llamadas a esta función:



6.4.2.3. DominanceRank()

```
int local_search.acceptation_type.AcceptMulticase.DominanceRank (
    State stateCandidate,
    List< State > list) [private]
```

Calcula el rango de dominancia del candidato.

Cuenta cuántas soluciones en la lista dominan al candidato.

Parámetros

<code>stateCandidate</code>	Estado candidato a evaluar.
<code>list</code>	Lista de soluciones de referencia.

Devuelve

Número de soluciones en la lista que dominan al candidato.

Gráfico de llamadas de esta función:

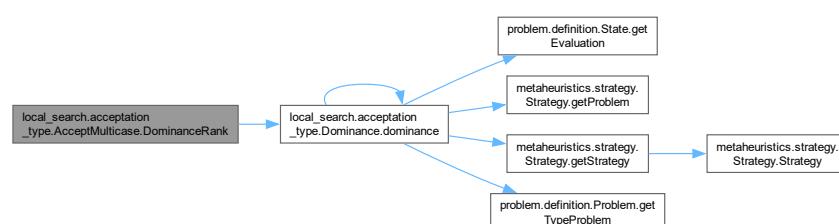


Gráfico de llamadas a esta función:



6.4.3. Documentación de datos miembro

6.4.3.1. rdm

```
final Random local_search.acceptation_type.AcceptMulticase.rdm = new Random() [static], [private]
```

Generador de números aleatorios compartido.

Se utiliza una instancia estática para mejorar la eficiencia y la calidad de la aleatoriedad.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/AcceptMulticase.java

6.5. Referencia de la clase local_search.acceptation_type.AcceptNotBad

Diagrama de herencia de local_search.acceptation_type.AcceptNotBad

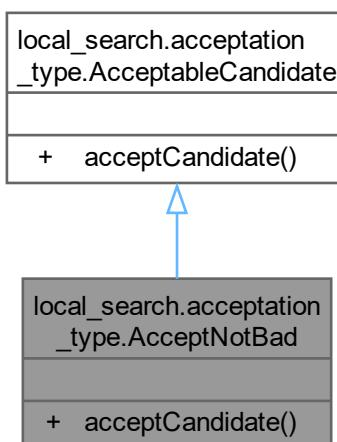
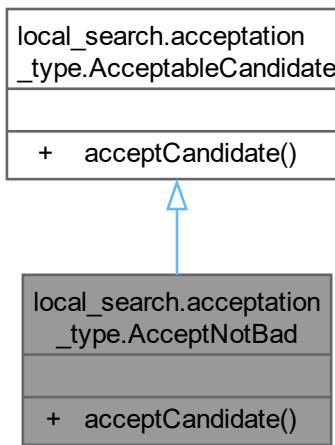


Diagrama de colaboración de local_search.acceptation_type.AcceptNotBad:



Métodos públicos

- Boolean `acceptCandidate (State stateCurrent, State stateCandidate)`

6.5.1. Documentación de funciones miembro

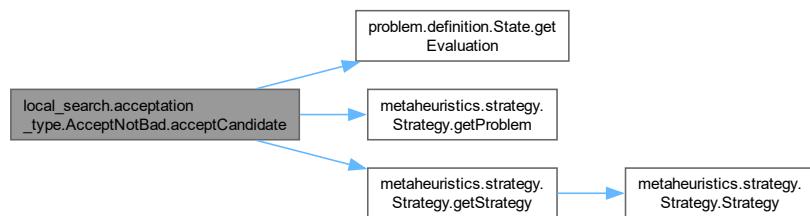
6.5.1.1. acceptCandidate()

```

Boolean local_search.acceptation_type.AcceptNotBad.acceptCandidate (
    State stateCurrent,
    State stateCandidate)
  
```

Reimplementado de `local_search.acceptation_type.AcceptableCandidate`.

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/AcceptNotBad.java

6.6. Referencia de la clase

local_search.acceptation_type.AcceptNotBadT

Diagrama de herencia de local_search.acceptation_type.AcceptNotBadT

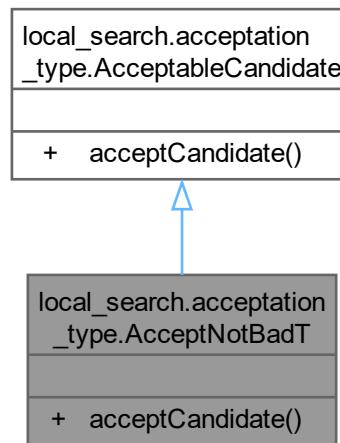
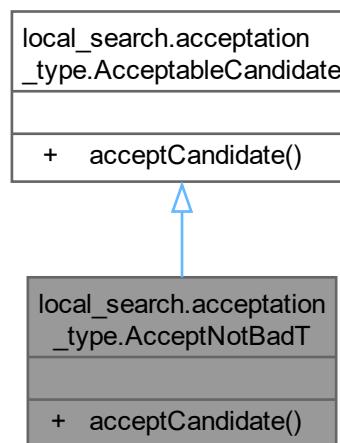


Diagrama de colaboración de local_search.acceptation_type.AcceptNotBadT:



Métodos públicos

- Boolean acceptCandidate (State stateCurrent, State stateCandidate) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

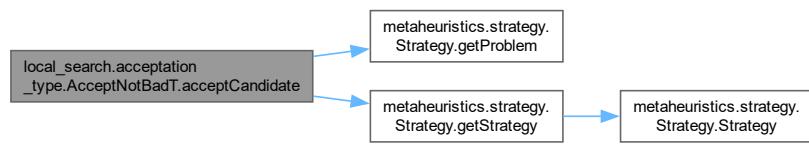
6.6.1. Documentación de funciones miembro

6.6.1.1. acceptCandidate()

```
Boolean local_search.acceptation_type.AcceptNotBadT.acceptCandidate (
    State stateCurrent,
    State stateCandidate) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [local_search.acceptation_type.AcceptableCandidate](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/AcceptNotBadT.java

6.7. Referencia de la clase

local_search.acceptation_type.AcceptNotBadU

Diagrama de herencia de `local_search.acceptation_type.AcceptNotBadU`

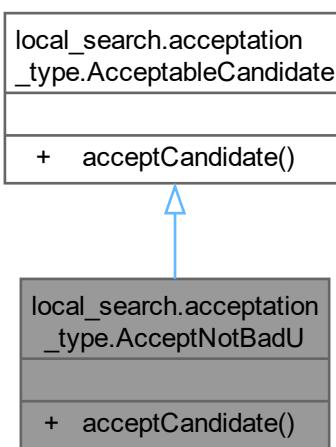
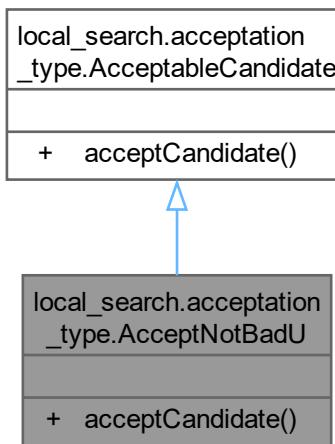


Diagrama de colaboración de local_search.acceptation_type.AcceptNotBadU:



Métodos públicos

- Boolean `acceptCandidate (State stateCurrent, State stateCandidate)` throws `IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.7.1. Documentación de funciones miembro

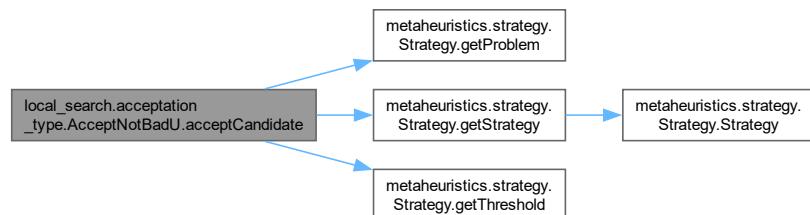
6.7.1.1. acceptCandidate()

```

Boolean local_search.acceptation_type.AcceptNotBadU.acceptCandidate (
    State stateCurrent,
    State stateCandidate) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
  
```

Reimplementado de [local_search.acceptation_type.AcceptableCandidate](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/AcceptNotBadU.java

6.8. Referencia de la clase

local_search.acceptation_type.AcceptNotDominated

Diagrama de herencia de local_search.acceptation_type.AcceptNotDominated

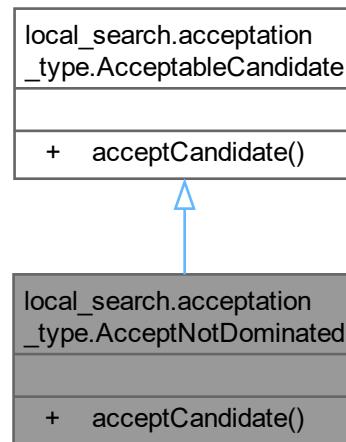
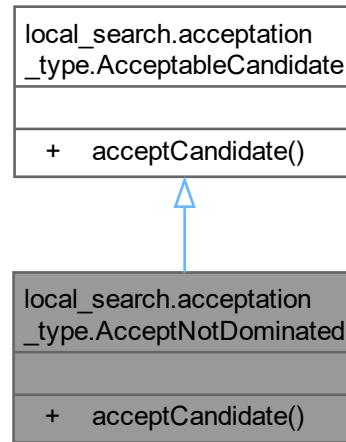


Diagrama de colaboración de local_search.acceptation_type.AcceptNotDominated:



Métodos públicos

- Boolean acceptCandidate (State stateCurrent, State stateCandidate)

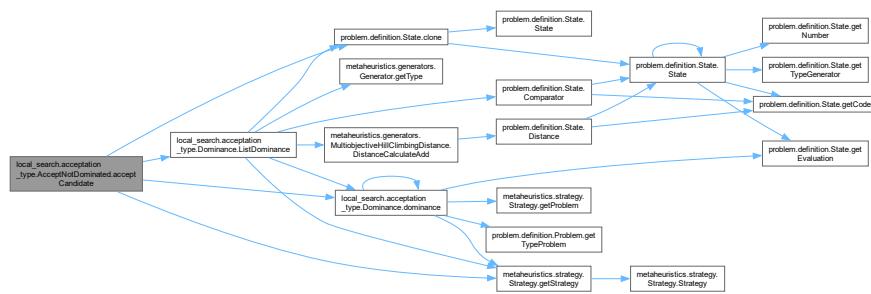
6.8.1. Documentación de funciones miembro

6.8.1.1. acceptCandidate()

```
Boolean local_search.acceptation_type.AcceptNotDominated.acceptCandidate (
    State stateCurrent,
    State stateCandidate)
```

Reimplementado de [local_search.acceptation_type.AcceptableCandidate](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/[AcceptNotDominated.java](#)

6.9. Referencia de la clase

local_search.acceptation_type.AcceptNotDominatedTabu

Diagrama de herencia de local_search.acceptation_type.AcceptNotDominatedTabu

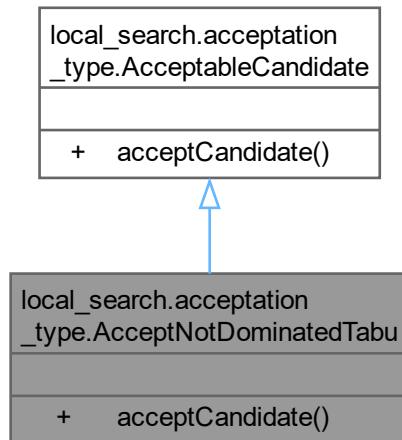
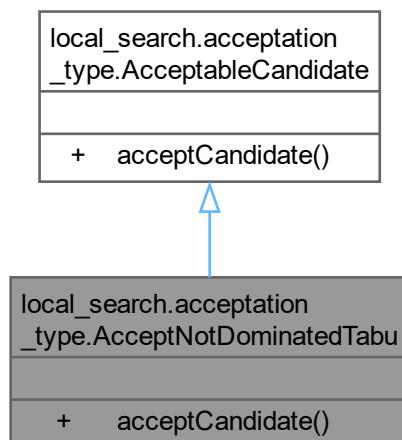


Diagrama de colaboración de local_search.acceptation_type.AcceptNotDominatedTabu:



Métodos públicos

- Boolean `acceptCandidate (State stateCurrent, State stateCandidate)`

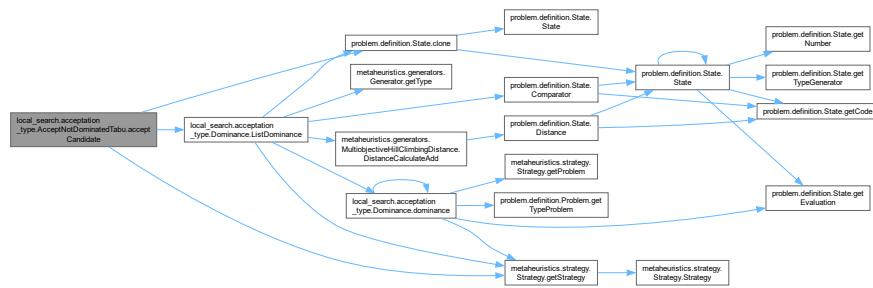
6.9.1. Documentación de funciones miembro

6.9.1.1. acceptCandidate()

```
Boolean local_search.acceptation_type.AcceptNotDominatedTabu.acceptCandidate (
    State stateCurrent,
    State stateCandidate)
```

Reimplementado de [local_search.acceptation_type.AcceptableCandidate](#).

Gráfico de llamadas de esta función:

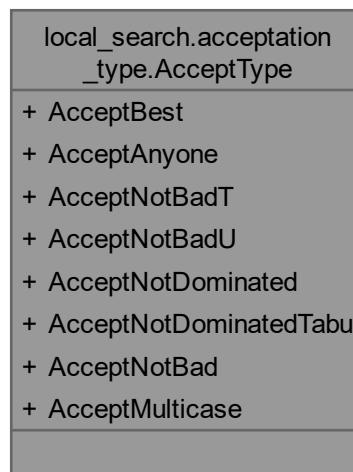


La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/[AcceptNotDominatedTabu.java](#)

6.10. Referencia de la enumeración local_search.acceptation_type.AcceptType

Diagrama de colaboración de local_search.acceptation_type.AcceptType:



Atributos públicos

- [AcceptBest](#)
- [AcceptAnyone](#)
- [AcceptNotBadT](#)
- [AcceptNotBadU](#)
- [AcceptNotDominated](#)
- [AcceptNotDominatedTabu](#)
- [AcceptNotBad](#)
- [AcceptMulticase](#)

6.10.1. Documentación de datos miembro

6.10.1.1. AcceptAnyone

```
local_search.acceptation_type.AcceptType.AcceptAnyone
```

6.10.1.2. AcceptBest

```
local_search.acceptation_type.AcceptType.AcceptBest
```

6.10.1.3. AcceptMulticase

```
local_search.acceptation_type.AcceptType.AcceptMulticase
```

6.10.1.4. AcceptNotBad

```
local_search.acceptation_type.AcceptType.AcceptNotBad
```

6.10.1.5. AcceptNotBadT

```
local_search.acceptation_type.AcceptType.AcceptNotBadT
```

6.10.1.6. AcceptNotBadU

```
local_search.acceptation_type.AcceptType.AcceptNotBadU
```

6.10.1.7. AcceptNotDominated

```
local_search.acceptation_type.AcceptType.AcceptNotDominated
```

6.10.1.8. AcceptNotDominatedTabu

```
local_search.acceptation_type.AcceptType.AcceptNotDominatedTabu
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/local_search/acceptation_type/AcceptType.java](#)

6.11. Referencia de la clase evolutionary_algorithms.complement.AIOMutation

Diagrama de herencia de evolutionary_algorithms.complement.AIOMutation

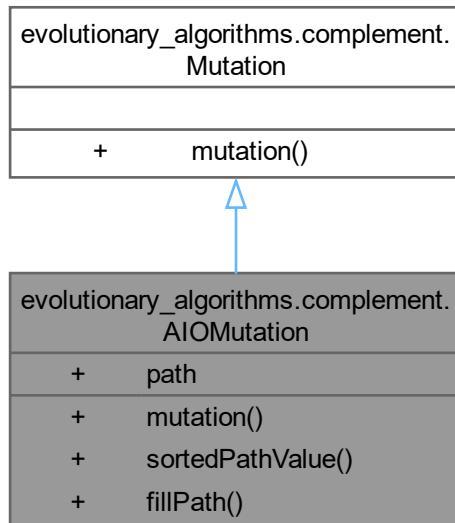
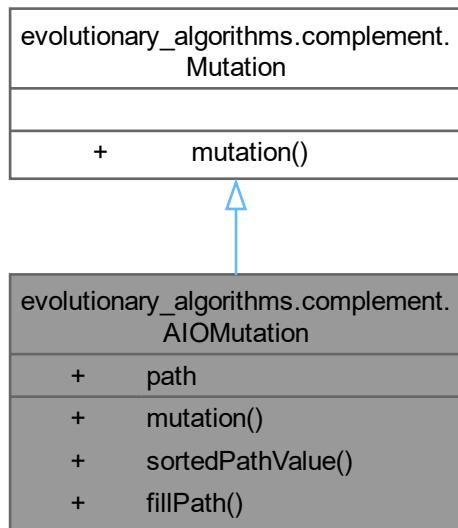


Diagrama de colaboración de `evolutionary_algorithms.complement.AIOMutation`:



Métodos públicos

- `State mutation (State state, double PM)`
- `void sortedPathValue (State state)`

Métodos públicos estáticos

- `static void fillPath ()`

Atributos públicos estáticos

- `static final ArrayList< Object > path = new ArrayList<Object>()`

6.11.1. Documentación de funciones miembro

6.11.1.1. `fillPath()`

```
void evolutionary_algorithms.complement.AIOMutation.fillPath () [static]
```

Gráfico de llamadas de esta función:

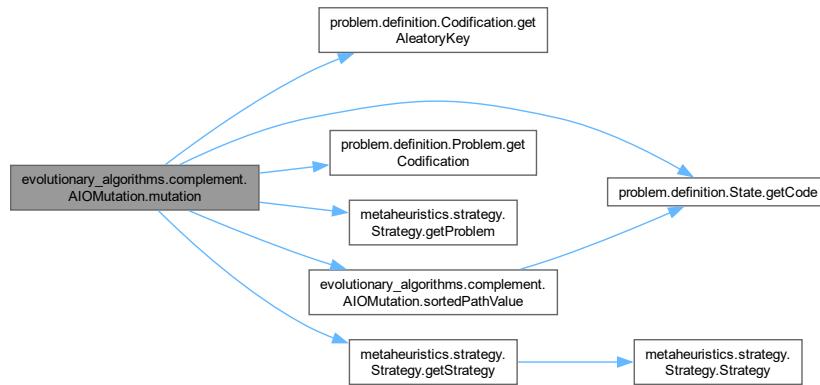


6.11.1.2. mutation()

```
State evolutionary_algorithms.complement.AIOMutation.mutation (
    State state,
    double PM)
```

Reimplementado de [evolutionary_algorithms.complement.Mutation](#).

Gráfico de llamadas de esta función:



6.11.1.3. sortedPathValue()

```
void evolutionary_algorithms.complement.AIOMutation.sortedPathValue (
    State state)
```

Gráfico de llamadas de esta función:



Gráfico de llamadas a esta función:



6.11.2. Documentación de datos miembro

6.11.2.1. path

```
final ArrayList<Object> evolutionary_algorithms.complement.AIOMutation.path = new ArrayList<Object>() [static]
```

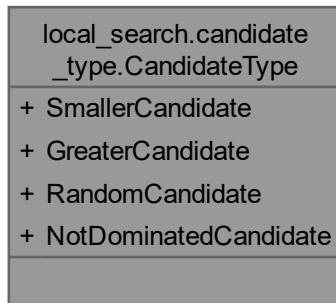
La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/AIOMutation.java](#)

6.12. Referencia de la enumeración

local_search.candidate_type.CandidateType

Diagrama de colaboración de local_search.candidate_type.CandidateType:



Atributos públicos

- [SmallerCandidate](#)
- [GreaterCandidate](#)
- [RandomCandidate](#)
- [NotDominatedCandidate](#)

6.12.1. Documentación de datos miembro

6.12.1.1. GreaterCandidate

```
local_search.candidate_type.CandidateType.GreaterCandidate
```

6.12.1.2. NotDominatedCandidate

```
local_search.candidate_type.CandidateType.NotDominatedCandidate
```

6.12.1.3. RandomCandidate

```
local_search.candidate_type.CandidateType.RandomCandidate
```

6.12.1.4. SmallerCandidate

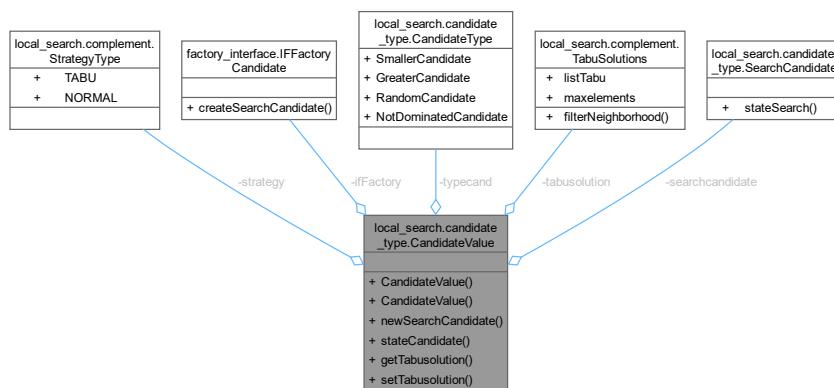
```
local_search.candidate_type.CandidateType.SmallerCandidate
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/local_search/candidate_type/CandidateType.java](#)

6.13. Referencia de la clase local_search.candidate_type.CandidateValue

Diagrama de colaboración de local_search.candidate_type.CandidateValue:



Métodos públicos

- [CandidateValue \(\)](#)
- [CandidateValue \(StrategyType strategy, IFFactoryCandidate ifFactory, CandidateType typecandidate, TabuSolutions tabusolution, SearchCandidate searchcandidate\)](#)
- [SearchCandidate newSearchCandidate \(CandidateType typecandidate\) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException](#)
- [State stateCandidate \(State stateCurrent, CandidateType typeCandidate, StrategyType strategy, Integer operatornumber, List< State > neighborhood\) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException](#)
- [TabuSolutions getTabusolution \(\)](#)
- [void setTabusolution \(TabuSolutions tabusolution\)](#)

Atributos privados

- `StrategyType strategy`
- `IFFactoryCandidate iffFactory`
- `CandidateType typecand`
- `TabuSolutions tabusolution`
- `SearchCandidate searchcandidate`

6.13.1. Documentación de constructores y destructores

6.13.1.1. CandidateValue() [1/2]

```
local_search.candidate_type.CandidateValue.CandidateValue ()
```

Gráfico de llamadas de esta función:

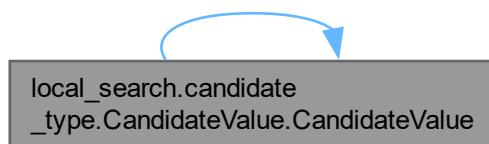


Gráfico de llamadas a esta función:



6.13.1.2. CandidateValue() [2/2]

```
local_search.candidate_type.CandidateValue.CandidateValue (StrategyType strategy,  
IFFactoryCandidate iffFactory,  
CandidateType typecand,  
TabuSolutions tabusolution,  
SearchCandidate searchcandidate)
```

6.13.2. Documentación de funciones miembro

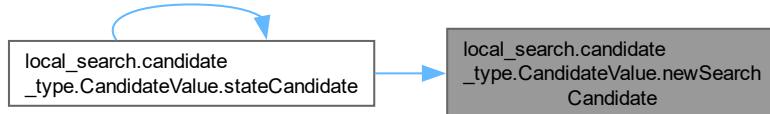
6.13.2.1. getTabusolution()

```
TabuSolutions local_search.candidate_type.CandidateValue.getTabusolution ()
```

6.13.2.2. newSearchCandidate()

```
SearchCandidate local_search.candidate_type.CandidateValue.newSearchCandidate (
    CandidateType typecandidate) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Gráfico de llamadas a esta función:



6.13.2.3. setTabusolution()

```
void local_search.candidate_type.CandidateValue.setTabusolution (
    TabuSolutions tabusolution)
```

6.13.2.4. stateCandidate()

```
State local_search.candidate_type.CandidateValue.stateCandidate (
    State stateCurrent,
    CandidateType typeCandidate,
    StrategyType strategy,
    Integer operatornumber,
    List< State > neighborhood) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Gráfico de llamadas de esta función:

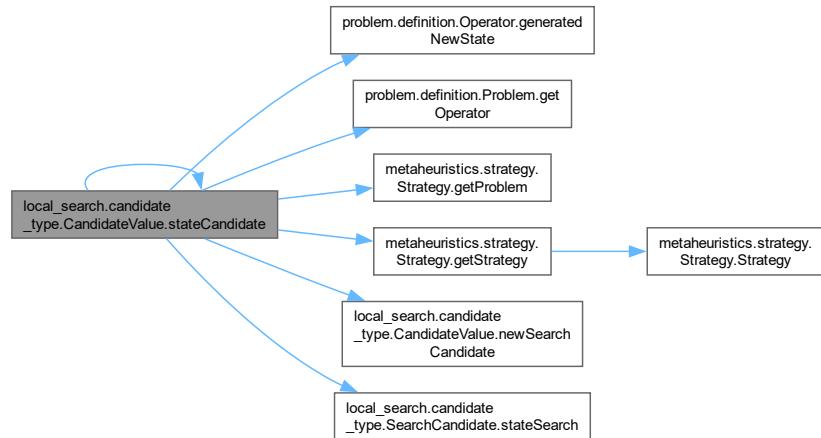


Gráfico de llamadas a esta función:



6.13.3. Documentación de datos miembro

6.13.3.1. ifFactory

```
IFFactoryCandidate local_search.candidate_type.CandidateValue.ifFactory [private]
```

6.13.3.2. searchcandidate

```
SearchCandidate local_search.candidate_type.CandidateValue.searchcandidate [private]
```

6.13.3.3. strategy

```
StrategyType local_search.candidate_type.CandidateValue.strategy [private]
```

6.13.3.4. tabusolution

```
TabuSolutions local_search.candidate_type.CandidateValue.tabusolution [private]
```

6.13.3.5. typecand

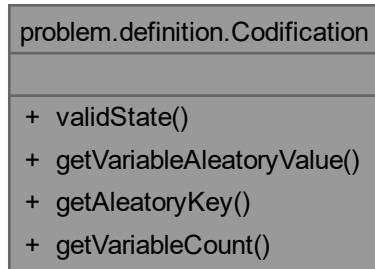
```
CandidateType local_search.candidate_type.CandidateValue.typecand [private]
```

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/candidate_type/CandidateValue.java

6.14. Referencia de la clase problem.definition.Codification

Diagrama de colaboración de problem.definition.Codification:



Métodos públicos

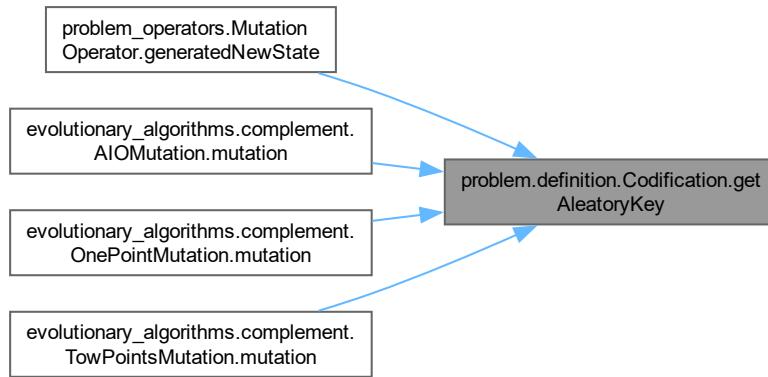
- abstract boolean [validState \(State state\)](#)
- abstract Object [getVariableAleatoryValue \(int key\)](#)
- abstract int [getAleatoryKey \(\)](#)
- abstract int [getVariableCount \(\)](#)

6.14.1. Documentación de funciones miembro

6.14.1.1. getAleatoryKey()

```
abstract int problem.definition.Codification.getAleatoryKey () [abstract]
```

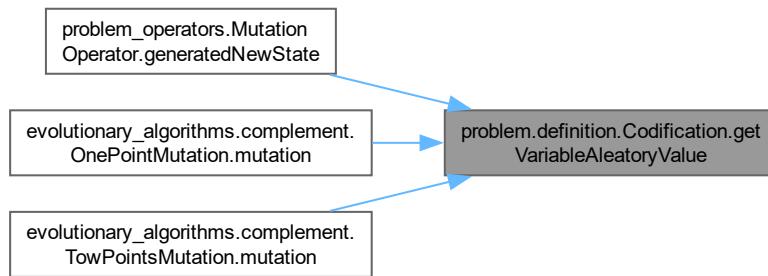
Gráfico de llamadas a esta función:



6.14.1.2. getVariableAleatoryValue()

```
abstract Object problem.definition.Codification.getVariableAleatoryValue (int key) [abstract]
```

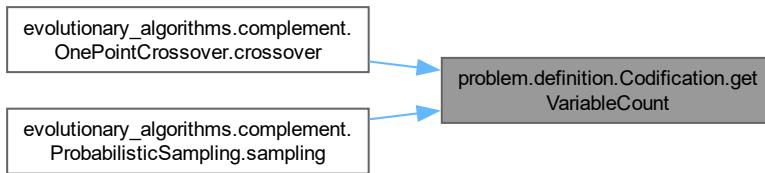
Gráfico de llamadas a esta función:



6.14.1.3. getVariableCount()

```
abstract int problem.definition.Codification.getVariableCount () [abstract]
```

Gráfico de llamadas a esta función:



6.14.1.4. validState()

```
abstract boolean problem.definition.Codification.validState (  
    State state) [abstract]
```

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/problem/definition/Codification.java

6.15. Referencia de la clase evolutionary_algorithms.complement.Crossover

Diagrama de herencia de evolutionary_algorithms.complement.Crossover

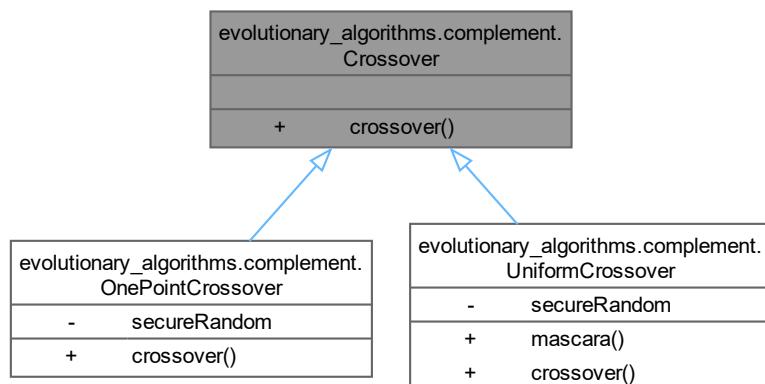
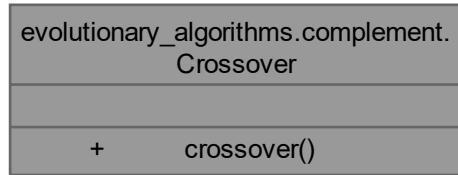


Diagrama de colaboración de `evolutionary_algorithms.complement.Crossover`:



Métodos públicos

- abstract `State crossover (State father1, State father2, double PC)`

6.15.1. Documentación de funciones miembro

6.15.1.1. crossover()

```
abstract State evolutionary_algorithms.complement.Crossover.crossover (
    State father1,
    State father2,
    double PC) [abstract]
```

Reimplementado en [evolutionary_algorithms.complement.OnePointCrossover](#) y [evolutionary_algorithms.complement.UniformCrossover](#)

Gráfico de llamadas a esta función:

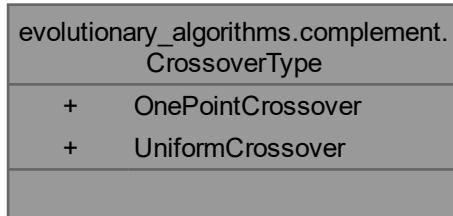


La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/evolutionary_algorithms/complement/Crossover.java`

6.16. Referencia de la enumeración evolutionary_algorithms.complement.CrossoverType

Diagrama de colaboración de evolutionary_algorithms.complement.CrossoverType:



Atributos públicos

- [OnePointCrossover](#)
- [UniformCrossover](#)

6.16.1. Documentación de datos miembro

6.16.1.1. OnePointCrossover

```
evolutionary_algorithms.complement.CrossoverType.OnePointCrossover
```

6.16.1.2. UniformCrossover

```
evolutionary_algorithms.complement.CrossoverType.UniformCrossover
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/CrossoverType.java](#)

6.17. Referencia de la clase

evolutionary_algorithms.complement.Distribution

Diagrama de herencia de evolutionary_algorithms.complement.Distribution

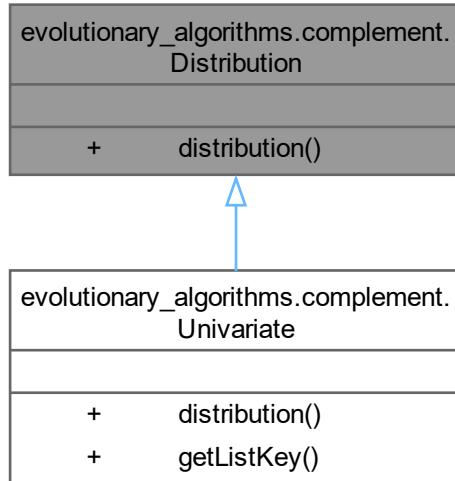
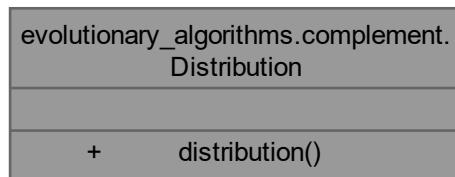


Diagrama de colaboración de `evolutionary_algorithms.complement.Distribution`:



Métodos públicos

- `abstract List< Probability > distribution (List< State > fathers)`

6.17.1. Documentación de funciones miembro

6.17.1.1. distribution()

```
abstract List< Probability > evolutionary_algorithms.complement.Distribution.distribution (   
    List< State > fathers) [abstract]
```

Reimplementado en [evolutionary_algorithms.complement.Univariate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/Distribution.java](#)

6.18. Referencia de la clase

metaheuristics.generators.DistributionEstimationAlgorithm

Diagrama de herencia de metaheuristics.generators.DistributionEstimationAlgorithm

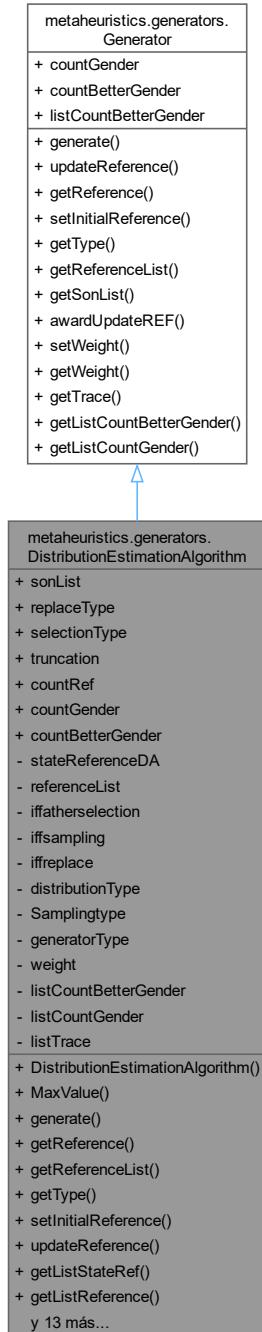
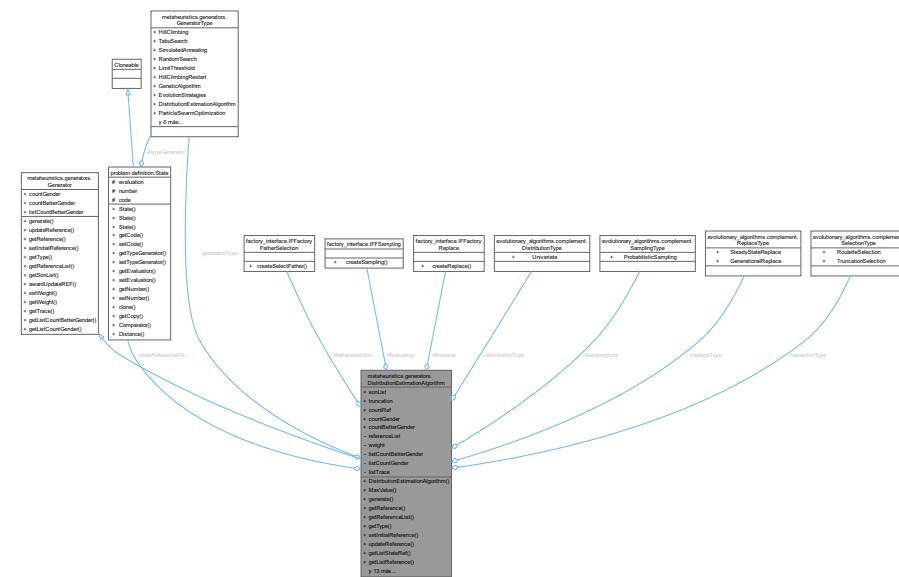


Diagrama de colaboración de metaheuristics.generators.DistributionEstimationAlgorithm:



Métodos públicos

- `DistributionEstimationAlgorithm ()`
- `List< State > maxValue (List< State > listInd)`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `State getReference ()`
- `List< State > getReferenceList ()`
- `GeneratorType getType ()`
- `void setInitialReference (State stateInitialRef)`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `List< State > getListStateRef ()`
- `List< State > getListReference ()`
- `void setListReference (List< State > listReference)`
- `GeneratorType getGeneratorType ()`
- `void setGeneratorType (GeneratorType generatorType)`
- `List< State > getFathersList ()` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `DistributionType getDistributionType ()`
- `void setDistributionType (DistributionType distributionType)`
- `int[] getListCountBetterGender ()`
- `int[] getListCountGender ()`
- `float[] getTrace ()`

Atributos públicos estáticos

- static List< State > sonList = new ArrayList<State>()
- static ReplaceType replaceType
- static SelectionType selectionType
- static int truncation
- static int countRef = 0
- static int countGender = 0
- static int countBetterGender = 0

Atributos privados

- State stateReferenceDA
- List< State > referenceList = new ArrayList<State>()
- IFFactoryFatherSelection iffatherselection
- IFFSampling iffsampling
- IFFactoryReplace iffreplace
- DistributionType distributionType
- SamplingType Samplingtype
- GeneratorType generatorType
- float weight
- int[] listCountBetterGender = new int[10]
- int[] listCountGender = new int[10]
- float[] listTrace = new float[1200000]

Otros miembros heredados

Atributos públicos heredados de `metaheuristics.generators.Generator`

- int countGender
- int countBetterGender
- int[] listCountBetterGender

6.18.1. Documentación de constructores y destructores

6.18.1.1. `DistributionEstimationAlgorithm()`

```
metaheuristics.generators.DistributionEstimationAlgorithm.DistributionEstimationAlgorithm ()
```

Gráfico de llamadas de esta función:

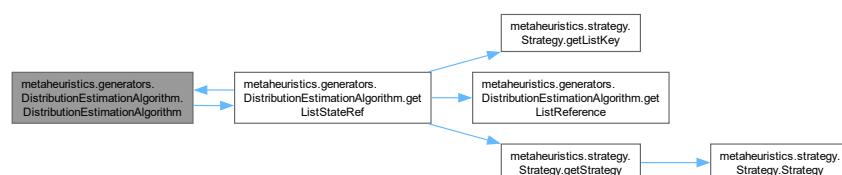


Gráfico de llamadas a esta función:



6.18.2. Documentación de funciones miembro

6.18.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.DistributionEstimationAlgorithm.awardUpdateREF (
    State stateCandidate)
```

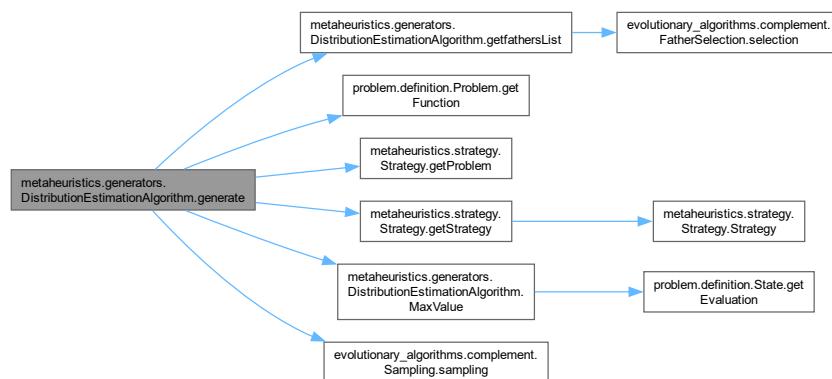
Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.2. generate()

```
State metaheuristics.generators.DistributionEstimationAlgorithm.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.18.2.3. getDistributionType()

```
DistributionType metaheuristics.generators.DistributionEstimationAlgorithm.getDistributionType
()
```

6.18.2.4. `getfathersList()`

```
List< State > metaheuristics.generators.DistributionEstimationAlgorithm.getfathersList ()  
throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException,  
IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Gráfico de llamadas de esta función:



Gráfico de llamadas a esta función:



6.18.2.5. `getGeneratorType()`

```
GeneratorType metaheuristics.generators.DistributionEstimationAlgorithm.getGeneratorType ()
```

6.18.2.6. `getListCountBetterGender()`

```
int[] metaheuristics.generators.DistributionEstimationAlgorithm.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.7. `getListCountGender()`

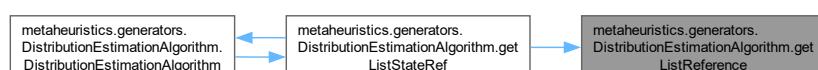
```
int[] metaheuristics.generators.DistributionEstimationAlgorithm.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.8. `getListReference()`

```
List< State > metaheuristics.generators.DistributionEstimationAlgorithm.getListReference ()
```

Gráfico de llamadas a esta función:



6.18.2.9. getListStateRef()

```
List< State > metaheuristics.generators.DistributionEstimationAlgorithm.getListStateRef ()
```

Gráfico de llamadas de esta función:

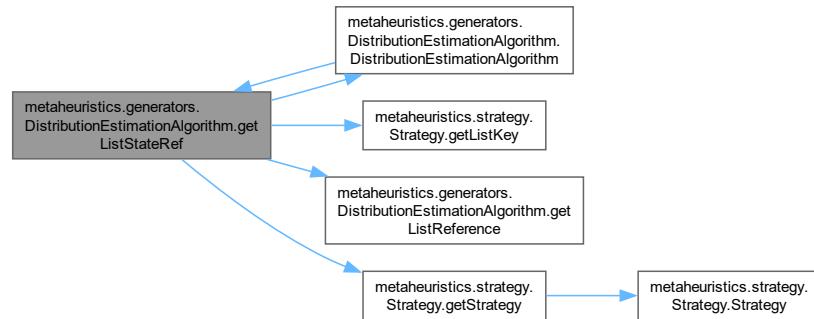
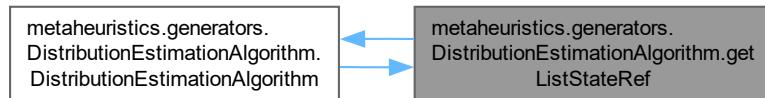


Gráfico de llamadas a esta función:

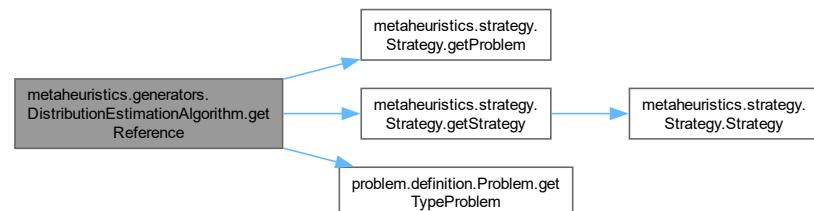


6.18.2.10. getReference()

```
State metaheuristics.generators.DistributionEstimationAlgorithm.getReference ()
```

Reimplementado de `metaheuristics.generators.Generator`.

Gráfico de llamadas de esta función:



6.18.2.11. `getReferenceList()`

```
List< State > metaheuristics.generators.DistributionEstimationAlgorithm.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.12. `getSonList()`

```
List< State > metaheuristics.generators.DistributionEstimationAlgorithm.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.13. `getTrace()`

```
float[ ] metaheuristics.generators.DistributionEstimationAlgorithm.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.14. `getType()`

```
GeneratorType metaheuristics.generators.DistributionEstimationAlgorithm.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.15. `getWeight()`

```
float metaheuristics.generators.DistributionEstimationAlgorithm.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.16. `MaxValue()`

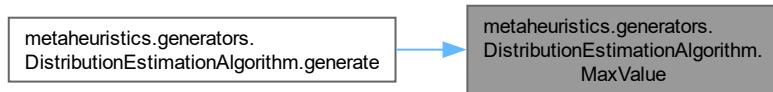
```
State metaheuristics.generators.DistributionEstimationAlgorithm.MaxValue (
```

`List< State > listInd)`

Gráfico de llamadas de esta función:



Gráfico de llamadas a esta función:



6.18.2.17. setDistributionType()

```
void metaheuristics.generators.DistributionEstimationAlgorithm.setDistributionType (   
    DistributionType distributionType)
```

6.18.2.18. setGeneratorType()

```
void metaheuristics.generators.DistributionEstimationAlgorithm.setGeneratorType (   
    GeneratorType generatorType)
```

6.18.2.19. setInitialReference()

```
void metaheuristics.generators.DistributionEstimationAlgorithm.setInitialReference (   
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.20. setListReference()

```
void metaheuristics.generators.DistributionEstimationAlgorithm.setListReference (   
    List< State > listReference)
```

6.18.2.21. setWeight()

```
void metaheuristics.generators.DistributionEstimationAlgorithm.setWeight (   
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.18.2.22. updateReference()

```
void metaheuristics.generators.DistributionEstimationAlgorithm.updateReference (   
    State stateCandidate,   
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,   
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,   
    NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.18.3. Documentación de datos miembro

6.18.3.1. countBetterGender

```
int metaheuristics.generators.DistributionEstimationAlgorithm.countBetterGender = 0 [static]
```

6.18.3.2. countGender

```
int metaheuristics.generators.DistributionEstimationAlgorithm.countGender = 0 [static]
```

6.18.3.3. countRef

```
int metaheuristics.generators.DistributionEstimationAlgorithm.countRef = 0 [static]
```

6.18.3.4. distributionType

```
DistributionType metaheuristics.generators.DistributionEstimationAlgorithm.distributionType  
[private]
```

6.18.3.5. generatorType

```
GeneratorType metaheuristics.generators.DistributionEstimationAlgorithm.generatorType [private]
```

6.18.3.6. iffatherselection

```
IFFFactoryFatherSelection metaheuristics.generators.DistributionEstimationAlgorithm.iffatherselection  
[private]
```

6.18.3.7. iffreplace

```
IFFFactoryReplace metaheuristics.generators.DistributionEstimationAlgorithm.iffreplace [private]
```

6.18.3.8. iffsampling

```
IFFSampling metaheuristics.generators.DistributionEstimationAlgorithm.iffsampling [private]
```

6.18.3.9. listCountBetterGender

```
int [] metaheuristics.generators.DistributionEstimationAlgorithm.listCountBetterGender = new  
int[10] [private]
```

6.18.3.10. listCountGender

```
int [] metaheuristics.generators.DistributionEstimationAlgorithm.listCountGender = new int[10]
[private]
```

6.18.3.11. listTrace

```
float [] metaheuristics.generators.DistributionEstimationAlgorithm.listTrace = new float[1200000]
[private]
```

6.18.3.12. referenceList

```
List<State> metaheuristics.generators.DistributionEstimationAlgorithm.referenceList = new
ArrayList<State>() [private]
```

6.18.3.13. replaceType

```
ReplaceType metaheuristics.generators.DistributionEstimationAlgorithm.replaceType [static]
```

6.18.3.14. Samplingtype

```
SamplingType metaheuristics.generators.DistributionEstimationAlgorithm.Samplingtype [private]
```

6.18.3.15. selectionType

```
SelectionType metaheuristics.generators.DistributionEstimationAlgorithm.selectionType [static]
```

6.18.3.16. sonList

```
List<State> metaheuristics.generators.DistributionEstimationAlgorithm.sonList = new Array←
List<State>() [static]
```

6.18.3.17. stateReferenceDA

```
State metaheuristics.generators.DistributionEstimationAlgorithm.stateReferenceDA [private]
```

6.18.3.18. truncation

```
int metaheuristics.generators.DistributionEstimationAlgorithm.truncation [static]
```

6.18.3.19. weight

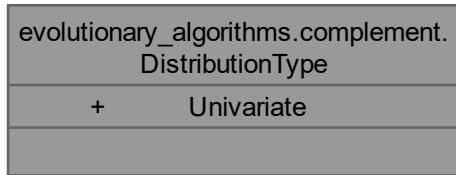
```
float metaheuristics.generators.DistributionEstimationAlgorithm.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/DistributionEstimationAlgorithm.java](#)

6.19. Referencia de la enumeración **evolutionary_algorithms.complement.DistributionType**

Diagrama de colaboración de `evolutionary_algorithms.complement.DistributionType`:



Atributos públicos

- [Univariate](#)

6.19.1. Documentación de datos miembro

6.19.1.1. Univariate

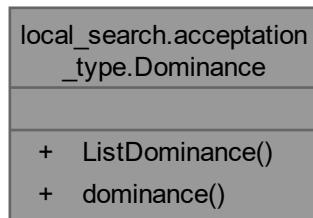
```
evolutionary_algorithms.complement.DistributionType.Univariate
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/DistributionType.java](#)

6.20. Referencia de la clase local_search.acceptation_type.Dominance

Diagrama de colaboración de local_search.acceptation_type.Dominance:



Métodos públicos

- boolean `ListDominance (State solutionX, List< State > list)`
- boolean `dominance (State solutionX, State solutionY)`

6.20.1. Documentación de funciones miembro

6.20.1.1. dominance()

```

boolean local_search.acceptation_type.Dominance.dominance (
    State solutionX,
    State solutionY)
  
```

Gráfico de llamadas de esta función:

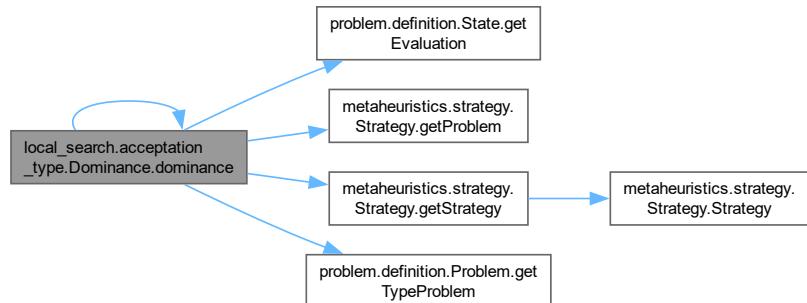
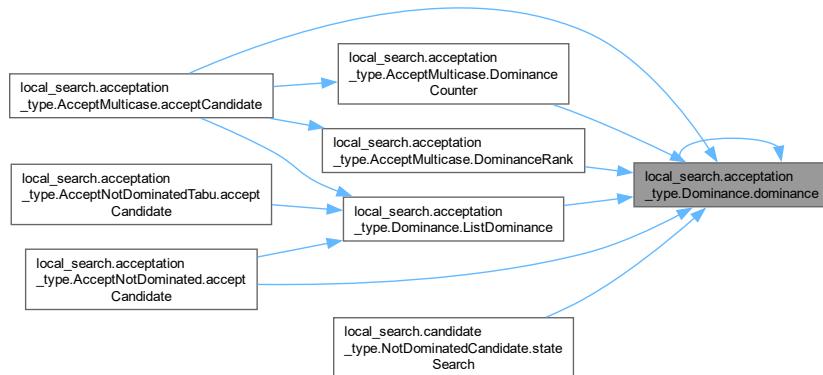


Gráfico de llamadas a esta función:



6.20.1.2. ListDominance()

```
boolean local_search.acceptation_type.Dominance.ListDominance (
    State solutionX,
    List< State > list)
```

Gráfico de llamadas de esta función:

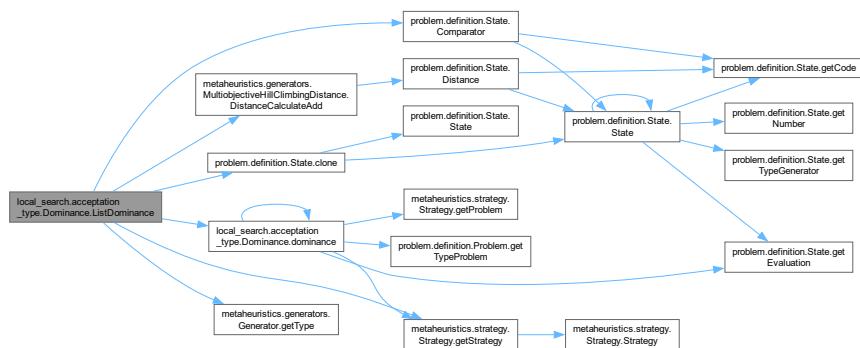
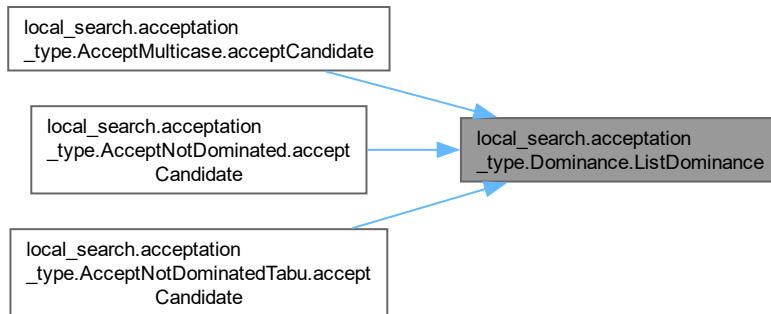


Gráfico de llamadas a esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/local_search/acceptation_type/[Dominance.java](#)

6.21. Referencia de la clase

metaheuristics.generators.EvolutionStrategies

Diagrama de herencia de metaheuristics.generators.EvolutionStrategies

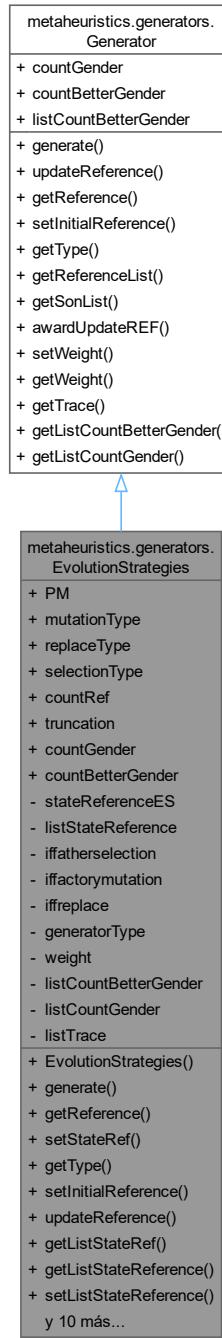
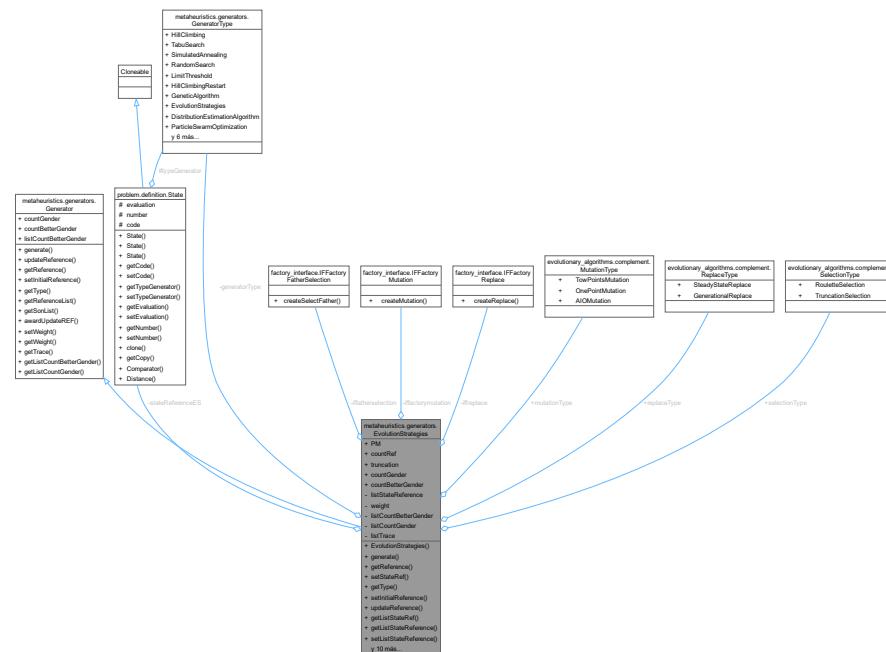


Diagrama de colaboración de metaheuristics.generators.EvolutionStrategies:



Métodos públicos

- **EvolutionStrategies ()**
 - **State generate (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**
 - **State getReference ()**
 - **void setStateRef (State stateRef)**
 - **GeneratorType getType ()**
 - **void setInitialReference (State stateInitialRef)**
 - **void updateReference (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**
 - **List< State > getListStateRef ()**
 - **List< State > getListStateReference ()**
 - **void setListStateReference (List< State > listStateReference)**
 - **GeneratorType getTypeGenerator ()**
 - **void setTypeGenerator (GeneratorType generatorType)**
 - **List< State > getReferenceList ()**
 - **List< State > getSonList ()**
 - **boolean awardUpdateREF (State stateCandidate)**
 - **float getWeight ()**
 - **void setWeight (float weight)**
 - **int[] getListCountBetterGender ()**
 - **int[] getListCountGender ()**
 - **float[] getTrace ()**

Atributos públicos estáticos

- static double PM
- static MutationType mutationType
- static ReplaceType replaceType
- static SelectionType selectionType
- static int countRef = 0
- static int truncation
- static int countGender = 0
- static int countBetterGender = 0

Atributos privados

- State stateReferenceES
- List< State > listStateReference = new ArrayList<State>()
- IFFactoryFatherSelection iffatherselection
- IFFactoryMutation iffactorymutation
- IFFactoryReplace iffreplace
- GeneratorType generatorType
- float weight = 50
- int[] listCountBetterGender = new int[10]
- int[] listCountGender = new int[10]
- float[] listTrace = new float[1200000]

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int countGender
- int countBetterGender
- int[] listCountBetterGender

6.21.1. Documentación de constructores y destructores

6.21.1.1. EvolutionStrategies()

```
metaheuristics.generators.EvolutionStrategies.EvolutionStrategies ()
```

Gráfico de llamadas de esta función:

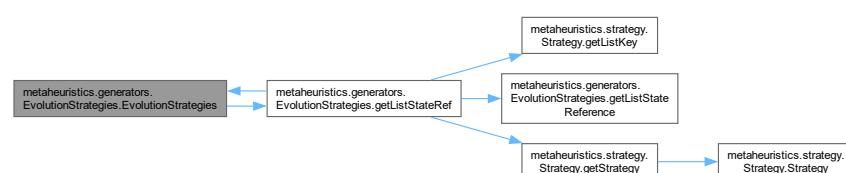
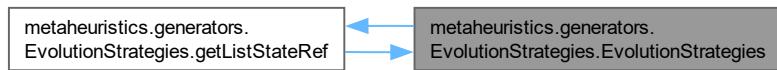


Gráfico de llamadas a esta función:



6.21.2. Documentación de funciones miembro

6.21.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.EvolutionStrategies.awardUpdateREF (
    State stateCandidate)
```

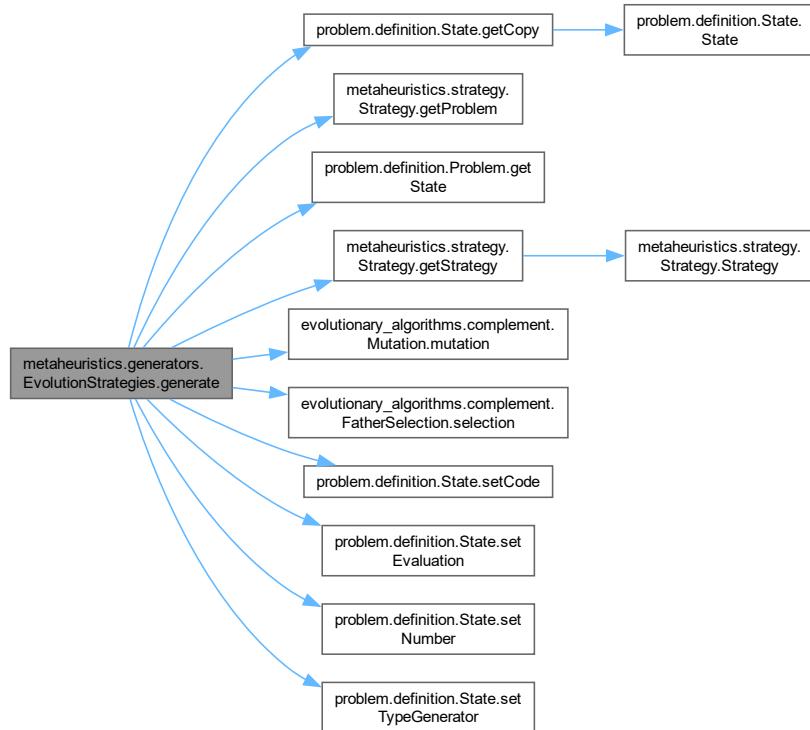
Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.2. generate()

```
State metaheuristics.generators.EvolutionStrategies.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.21.2.3. getListCountBetterGender()

```
int[] metaheuristics.generators.EvolutionStrategies.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.4. getListCountGender()

```
int[] metaheuristics.generators.EvolutionStrategies.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.5. getListStateRef()

```
List< State > metaheuristics.generators.EvolutionStrategies.getListStateRef ()
```

Gráfico de llamadas de esta función:

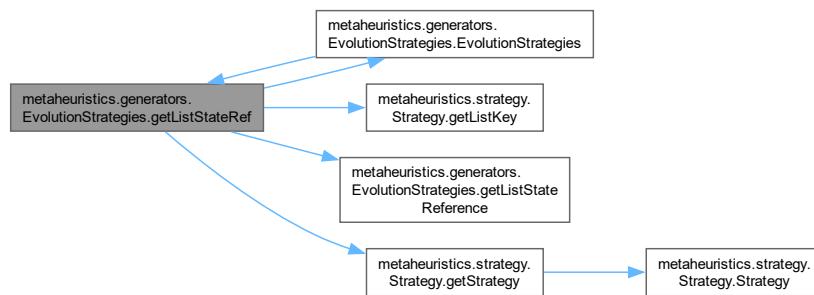
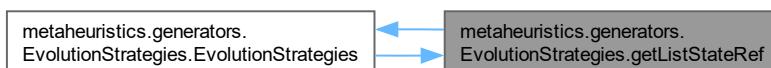


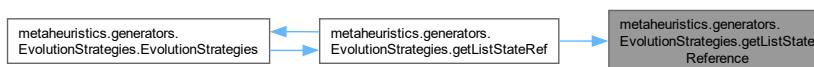
Gráfico de llamadas a esta función:



6.21.2.6. getListStateReference()

```
List< State > metaheuristics.generators.EvolutionStrategies.getListStateReference ()
```

Gráfico de llamadas a esta función:

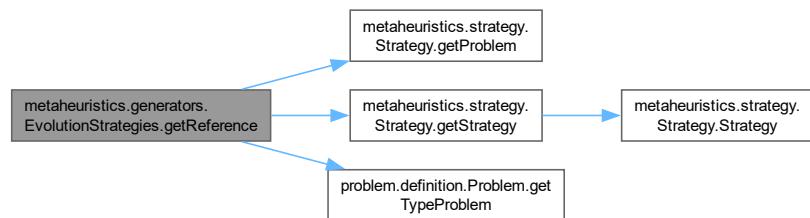


6.21.2.7. `getReference()`

```
State metaheuristics.generators.EvolutionStrategies.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.21.2.8. `getReferenceList()`

```
List< State > metaheuristics.generators.EvolutionStrategies.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.9. `getSonList()`

```
List< State > metaheuristics.generators.EvolutionStrategies.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.10. `getTrace()`

```
float[] metaheuristics.generators.EvolutionStrategies.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.11. `getType()`

```
GeneratorType metaheuristics.generators.EvolutionStrategies.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.12. `getTypeGenerator()`

```
GeneratorType metaheuristics.generators.EvolutionStrategies.getTypeGenerator ()
```

6.21.2.13. `getWeight()`

```
float metaheuristics.generators.EvolutionStrategies.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.14. `setInitialReference()`

```
void metaheuristics.generators.EvolutionStrategies.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.15. `setListStateReference()`

```
void metaheuristics.generators.EvolutionStrategies.setListStateReference (
    List< State > listStateReference)
```

6.21.2.16. `setStateRef()`

```
void metaheuristics.generators.EvolutionStrategies.setStateRef (
    State stateRef)
```

6.21.2.17. `setTypeGenerator()`

```
void metaheuristics.generators.EvolutionStrategies.setTypeGenerator (
    GeneratorType generatorType)
```

6.21.2.18. `setWeight()`

```
void metaheuristics.generators.EvolutionStrategies.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.21.2.19. `updateReference()`

```
void metaheuristics.generators.EvolutionStrategies.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.21.3. Documentación de datos miembro

6.21.3.1. countBetterGender

```
int metaheuristics.generators.EvolutionStrategies.countBetterGender = 0 [static]
```

6.21.3.2. countGender

```
int metaheuristics.generators.EvolutionStrategies.countGender = 0 [static]
```

6.21.3.3. countRef

```
int metaheuristics.generators.EvolutionStrategies.countRef = 0 [static]
```

6.21.3.4. generatorType

```
GeneratorType metaheuristics.generators.EvolutionStrategies.generatorType [private]
```

6.21.3.5. iffactorymutation

```
IFFactoryMutation metaheuristics.generators.EvolutionStrategies.iffactorymutation [private]
```

6.21.3.6. iffatherselection

```
IFFactoryFatherSelection metaheuristics.generators.EvolutionStrategies.iffatherselection [private]
```

6.21.3.7. iffreplace

```
IFFactoryReplace metaheuristics.generators.EvolutionStrategies.iffreplace [private]
```

6.21.3.8. listCountBetterGender

```
int [] metaheuristics.generators.EvolutionStrategies.listCountBetterGender = new int[10] [private]
```

6.21.3.9. listCountGender

```
int [] metaheuristics.generators.EvolutionStrategies.listCountGender = new int[10] [private]
```

6.21.3.10. listStateReference

```
List<State> metaheuristics.generators.EvolutionStrategies.listStateReference = new Array<-->  
List<State>() [private]
```

6.21.3.11. **listTrace**

```
float [] metaheuristics.generators.EvolutionStrategies.listTrace = new float[1200000] [private]
```

6.21.3.12. **mutationType**

```
MutationType metaheuristics.generators.EvolutionStrategies.mutationType [static]
```

6.21.3.13. **PM**

```
double metaheuristics.generators.EvolutionStrategies.PM [static]
```

6.21.3.14. **replaceType**

```
ReplaceType metaheuristics.generators.EvolutionStrategies.replaceType [static]
```

6.21.3.15. **selectionType**

```
SelectionType metaheuristics.generators.EvolutionStrategies.selectionType [static]
```

6.21.3.16. **stateReferenceES**

```
State metaheuristics.generators.EvolutionStrategies.stateReferenceES [private]
```

6.21.3.17. **truncation**

```
int metaheuristics.generators.EvolutionStrategies.truncation [static]
```

6.21.3.18. **weight**

```
float metaheuristics.generators.EvolutionStrategies.weight = 50 [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/EvolutionStrategies.java](#)

6.22. Referencia de la clase problem.extension.FactoresPonderados

Diagrama de herencia de problem.extension.FactoresPonderados

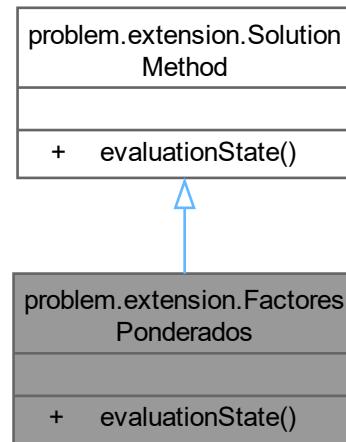
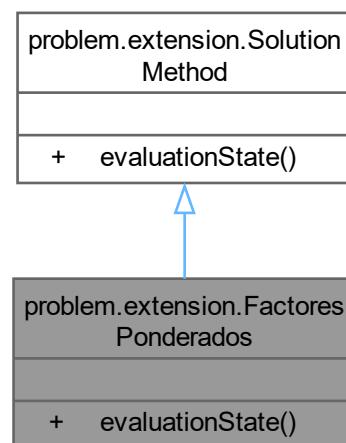


Diagrama de colaboración de problem.extension.FactoresPonderados:



Métodos públicos

- void [evaluationState \(State state\)](#)

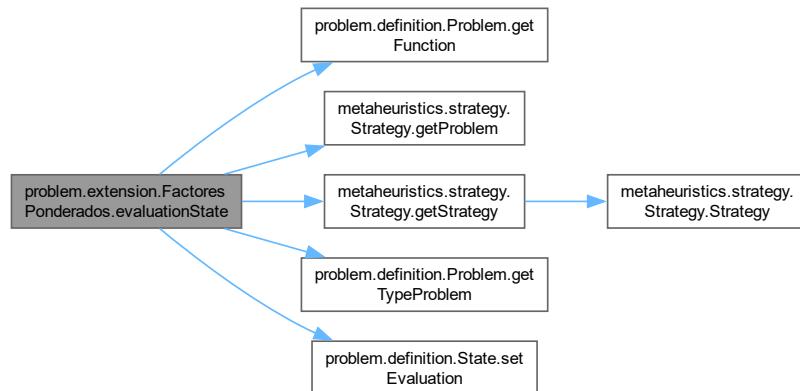
6.22.1. Documentación de funciones miembro

6.22.1.1. evaluationState()

```
void problem.extension.FactoresPonderados.evaluationState (
    State state)
```

Reimplementado de [problem.extension.SolutionMethod](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/extension/FactoresPonderados.java](#)

6.23. Referencia de la clase factory_method.FactoryAcceptCandidate

Diagrama de herencia de `factory_method.FactoryAcceptCandidate`

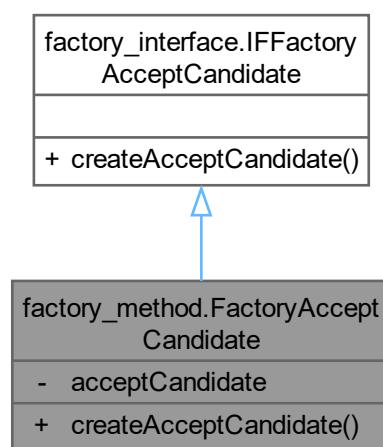
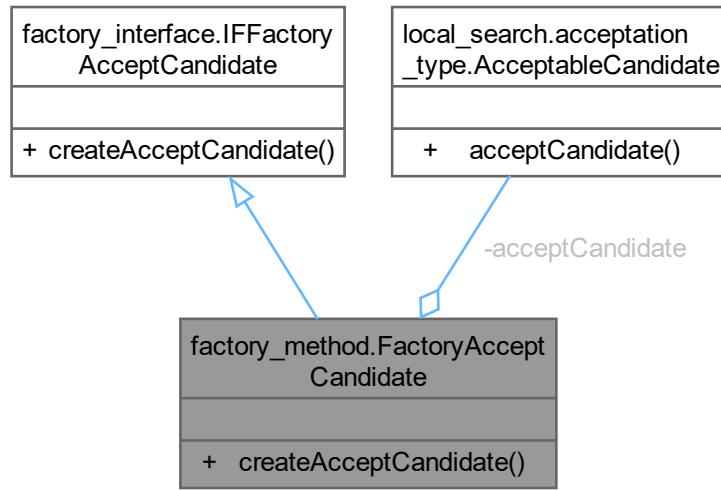


Diagrama de colaboración de factory_method.FactoryAcceptCandidate:



Métodos públicos

- `AcceptableCandidate createAcceptCandidate (AcceptType typeacceptation) throws IllegalArgument←Exception, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

Atributos privados

- `AcceptableCandidate acceptCandidate`

6.23.1. Documentación de funciones miembro

6.23.1.1. createAcceptCandidate()

```

AcceptableCandidate factory_method.FactoryAcceptCandidate.createAcceptCandidate (
    AcceptType typeacceptation) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
  
```

Implementa [factory_interface.IFFactoryAcceptCandidate](#).

Gráfico de llamadas de esta función:



6.23.2. Documentación de datos miembro

6.23.2.1. acceptCandidate

```
AcceptableCandidate factory_method.FactoryAcceptCandidate.acceptCandidate [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactoryAcceptCandidate.java](#)

6.24. Referencia de la clase factory_method.FactoryCandidate

Diagrama de herencia de `factory_method.FactoryCandidate`

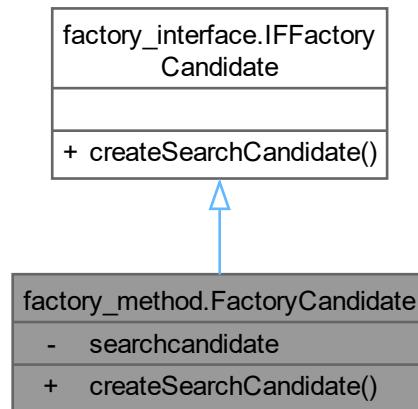
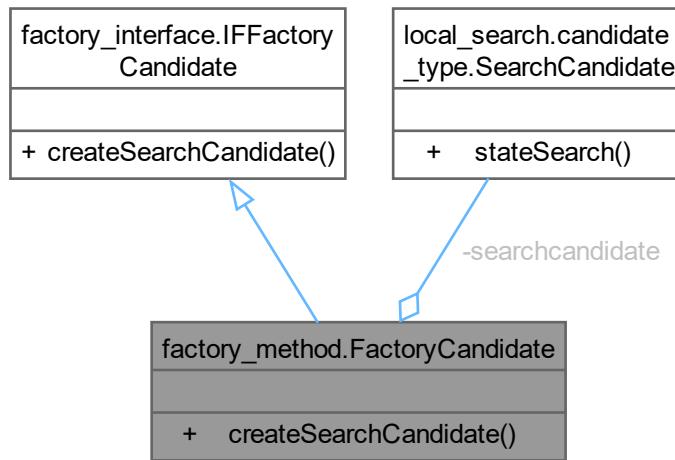


Diagrama de colaboración de factory_method.FactoryCandidate:



Métodos públicos

- `SearchCandidate createSearchCandidate (CandidateType typeCandidate)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

Atributos privados

- `SearchCandidate searchcandidate`

6.24.1. Documentación de funciones miembro

6.24.1.1. createSearchCandidate()

```

SearchCandidate factory_method.FactoryCandidate.createSearchCandidate (
    CandidateType typeCandidate) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
  
```

Implementa `factory_interface.IFFactoryCandidate`.

Gráfico de llamadas de esta función:



6.24.2. Documentación de datos miembro

6.24.2.1. searchcandidate

```
SearchCandidate factory_method.FactoryCandidate.searchcandidate [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactoryCandidate.java](#)

6.25. Referencia de la clase factory_method.FactoryCrossover

Diagrama de herencia de factory_method.FactoryCrossover

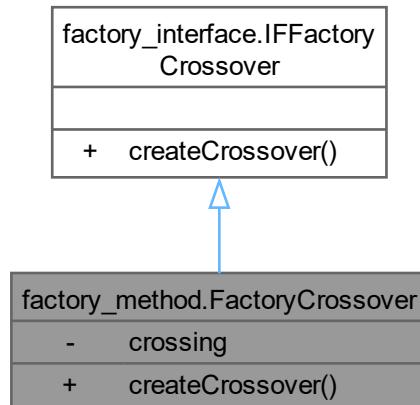
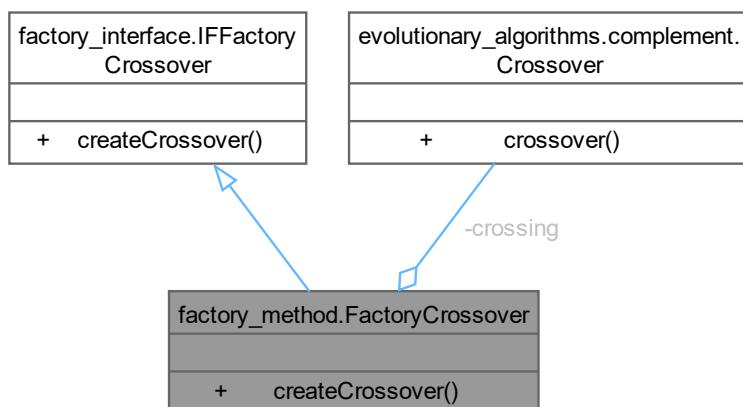


Diagrama de colaboración de factory_method.FactoryCrossover:



Métodos públicos

- `Crossover createCrossover (CrossoverType CrossoverType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

Atributos privados

- `Crossover crossing`

6.25.1. Documentación de funciones miembro

6.25.1.1. createCrossover()

```
Crossover factory_method.FactoryCrossover.createCrossover (
    CrossoverType CrossoverType) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementa `factory_interface.IFFactoryCrossover`.

Gráfico de llamadas de esta función:



6.25.2. Documentación de datos miembro

6.25.2.1. crossing

```
Crossover factory_method.FactoryCrossover.crossing [private]
```

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/factory_method/FactoryCrossover.java`

6.26. Referencia de la clase factory_method.FactoryDistribution

Diagrama de herencia de factory_method.FactoryDistribution

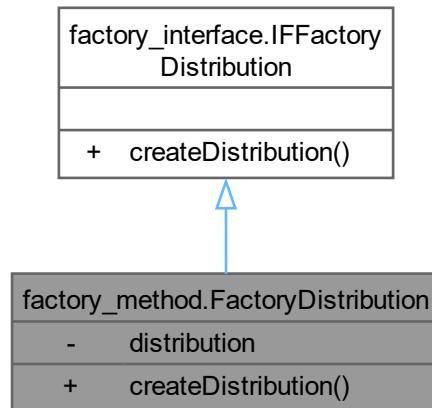
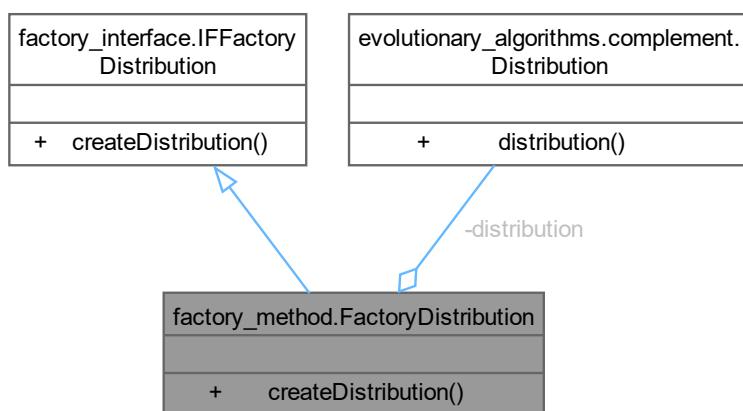


Diagrama de colaboración de factory_method.FactoryDistribution:



Métodos públicos

- `Distribution createDistribution (DistributionType distributiontype)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoMethodException`

Atributos privados

- [Distribution distribution](#)

6.26.1. Documentación de funciones miembro

6.26.1.1. [createDistribution\(\)](#)

```
Distribution factory_method.FactoryDistribution.createDistribution (
    DistributionType distributiontype) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Implementa [factory_interface.IFFactoryDistribution](#).

Gráfico de llamadas de esta función:



6.26.2. Documentación de datos miembro

6.26.2.1. [distribution](#)

```
Distribution factory_method.FactoryDistribution.distribution [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactoryDistribution.java](#)

6.27. Referencia de la clase factory_method.FactoryFatherSelection

Diagrama de herencia de factory_method.FactoryFatherSelection

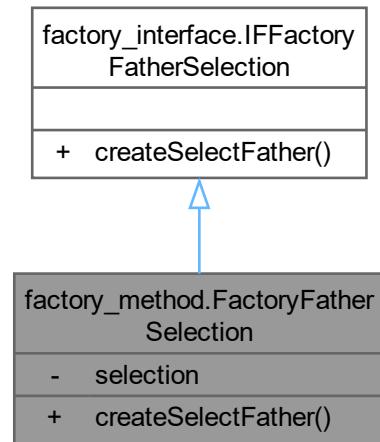
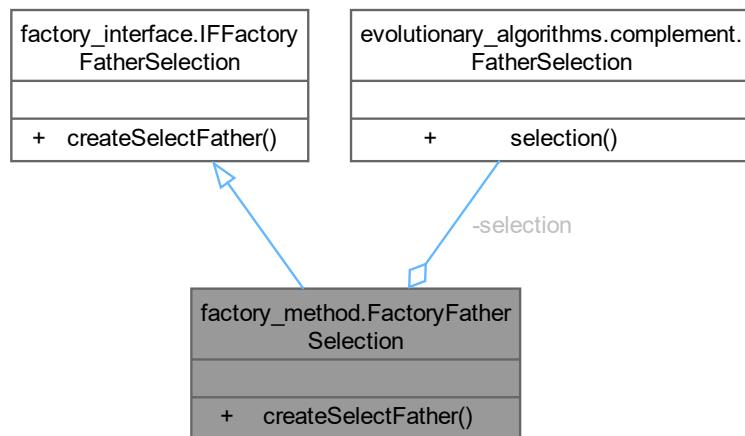


Diagrama de colaboración de factory_method.FactoryFatherSelection:



Métodos públicos

- `FatherSelection createSelectFather (SelectionType selectionType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

Atributos privados

- [FatherSelection selection](#)

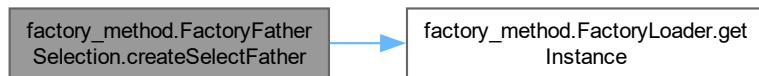
6.27.1. Documentación de funciones miembro

6.27.1.1. [createSelectFather\(\)](#)

```
FatherSelection factory_method.FactoryFatherSelection.createSelectFather (
    SelectionType selectionType) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementa [factory_interface.IFFactoryFatherSelection](#).

Gráfico de llamadas de esta función:



6.27.2. Documentación de datos miembro

6.27.2.1. [selection](#)

```
FatherSelection factory_method.FactoryFatherSelection.selection [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactoryFatherSelection.java](#)

6.28. Referencia de la clase factory_method.FactoryGenerator

Diagrama de herencia de factory_method.FactoryGenerator

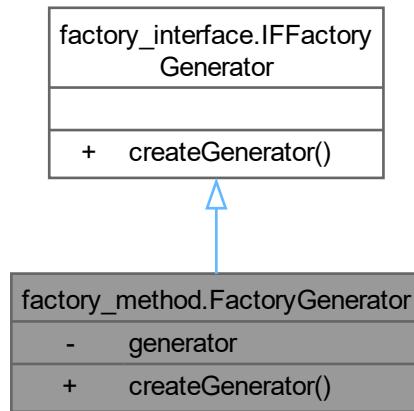
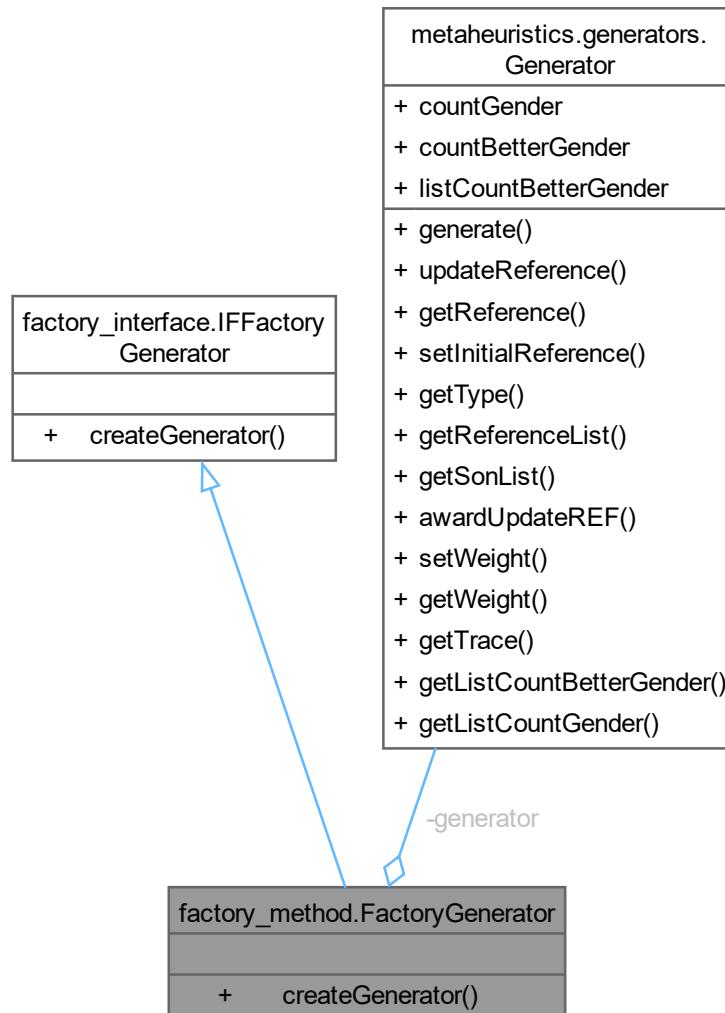


Diagrama de colaboración de factory_method.FactoryGenerator:



Métodos públicos

- `Generator createGenerator (GeneratorType generatorType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

Atributos privados

- `Generator generator`

6.28.1. Documentación de funciones miembro

6.28.1.1. createGenerator()

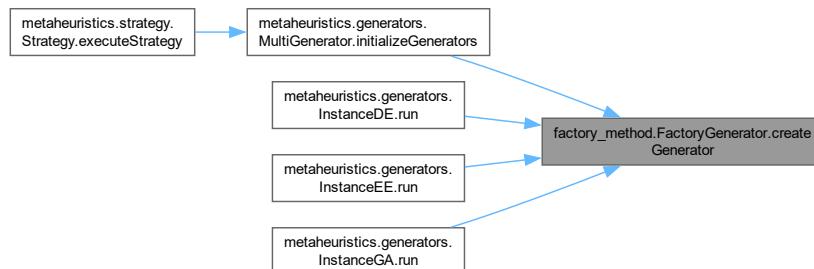
```
Generator factory_method.FactoryGenerator.createGenerator (
    GeneratorType generatorType) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementa [factory_interface.IFFactoryGenerator](#).

Gráfico de llamadas de esta función:



Gráfico de llamadas a esta función:



6.28.2. Documentación de datos miembro

6.28.2.1. generator

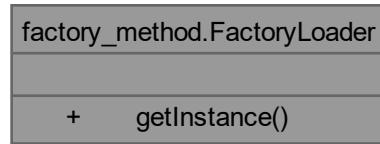
```
Generator factory_method.FactoryGenerator.generator [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactoryGenerator.java](#)

6.29. Referencia de la clase factory_method.FactoryLoader

Diagrama de colaboración de factory_method.FactoryLoader:



Métodos públicos estáticos

- static Object [getINSTANCE](#) (String className) throws ClassNotFoundException, IllegalArgumentException, SecurityException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

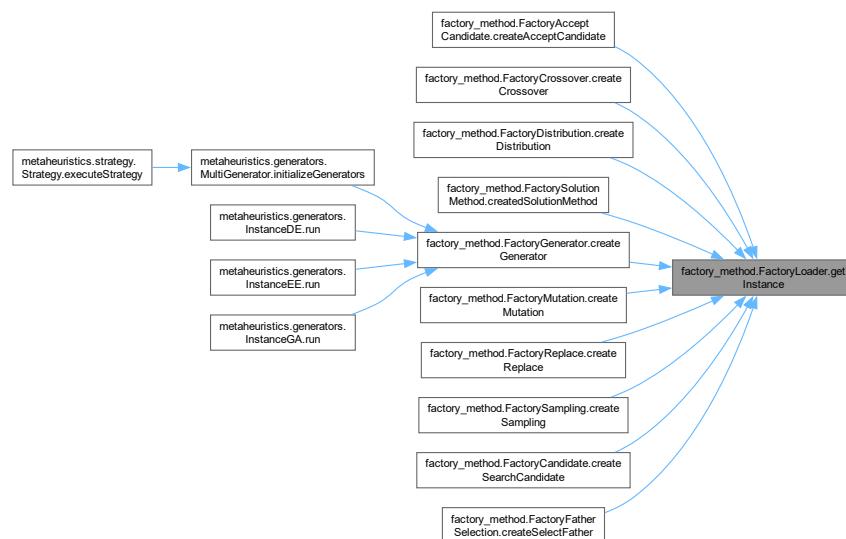
6.29.1. Documentación de funciones miembro

6.29.1.1. [getINSTANCE\(\)](#)

```

Object factory_method.FactoryLoader.getINSTANCE (
    String className) throws ClassNotFoundException, IllegalArgumentException, SecurityException,
    InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException [static]
  
```

Gráfico de llamadas a esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/factory_method/[FactoryLoader.java](#)

6.30. Referencia de la clase factory_method.FactoryMutation

Diagrama de herencia de factory_method.FactoryMutation

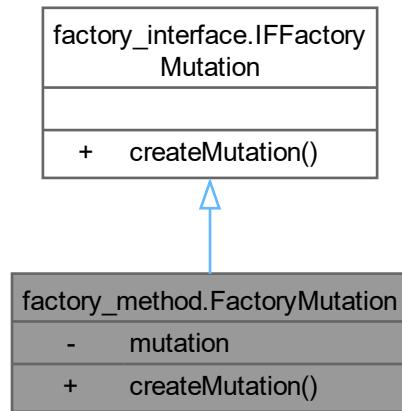
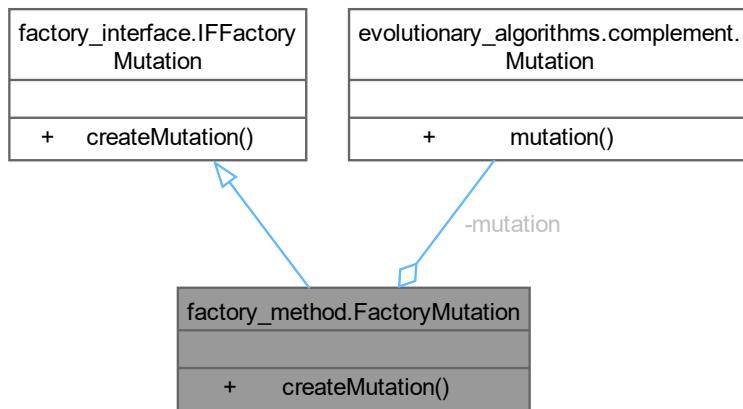


Diagrama de colaboración de factory_method.FactoryMutation:



Métodos públicos

- **Mutation createMutation (MutationType typeMutation)** throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

Atributos privados

- [Mutation mutation](#)

6.30.1. Documentación de funciones miembro

6.30.1.1. [createMutation\(\)](#)

```
Mutation factory_method.FactoryMutation.createMutation (
    MutationType typeMutation) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementa [factory_interface.IFFactoryMutation](#).

Gráfico de llamadas de esta función:



6.30.2. Documentación de datos miembro

6.30.2.1. [mutation](#)

```
Mutation factory_method.FactoryMutation.mutation [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactoryMutation.java](#)

6.31. Referencia de la clase factory_method.FactoryReplace

Diagrama de herencia de factory_method.FactoryReplace

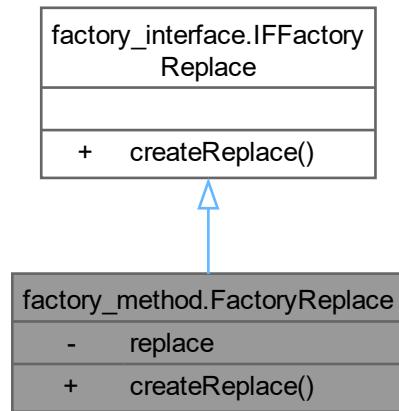
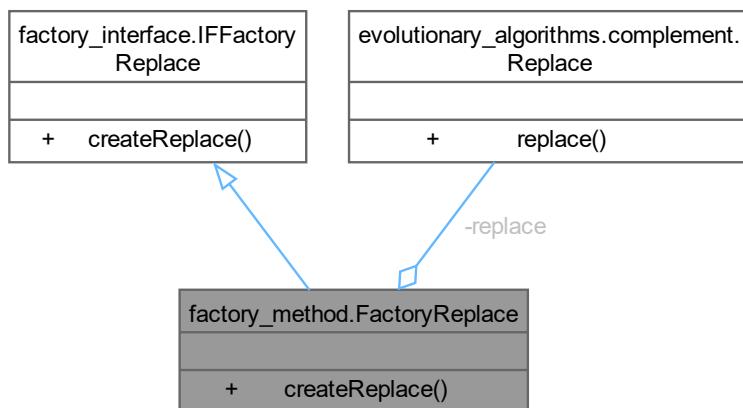


Diagrama de colaboración de factory_method.FactoryReplace:



Métodos públicos

- `Replace createReplace (ReplaceType typereplace)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

Atributos privados

- Replace replace

6.31.1. Documentación de funciones miembro

6.31.1.1. createReplace()

```
Replace factory_method.FactoryReplace.createReplace (
    ReplaceType typereplace) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementa [factory_interface.IFFactoryReplace](#).

Gráfico de llamadas de esta función:



6.31.2. Documentación de datos miembro

6.31.2.1. replace

```
Replace factory_method.FactoryReplace.replace [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactoryReplace.java](#)

6.32. Referencia de la clase factory_method.FactorySampling

Diagrama de herencia de factory_method.FactorySampling

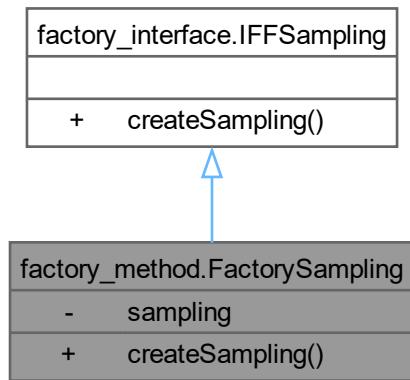
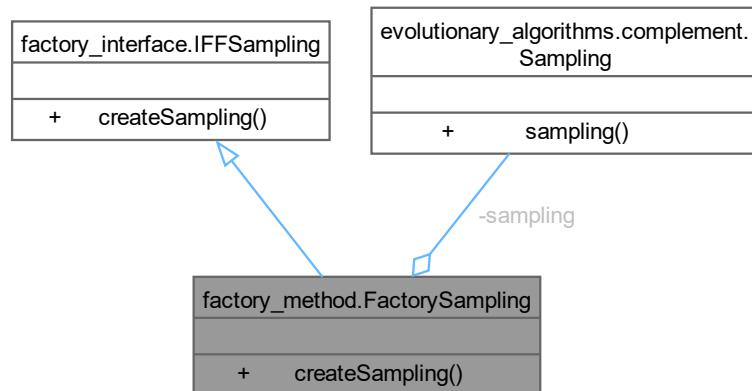


Diagrama de colaboración de factory_method.FactorySampling:



Métodos públicos

- `Sampling createSampling (SamplingType typesampling) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

Atributos privados

- `Sampling sampling`

6.32.1. Documentación de funciones miembro

6.32.1.1. createSampling()

```
Sampling factory_method.FactorySampling.createSampling (
    SamplingType typesampling) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementa [factory_interface.IFFSampling](#).

Gráfico de llamadas de esta función:



6.32.2. Documentación de datos miembro

6.32.2.1. sampling

```
Sampling factory_method.FactorySampling.sampling [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactorySampling.java](#)

6.33. Referencia de la clase factory_method.FactorySolutionMethod

Diagrama de herencia de `factory_method.FactorySolutionMethod`

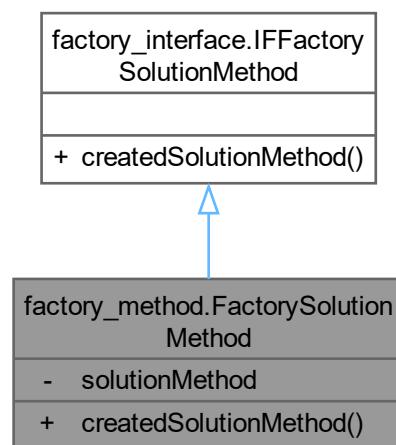
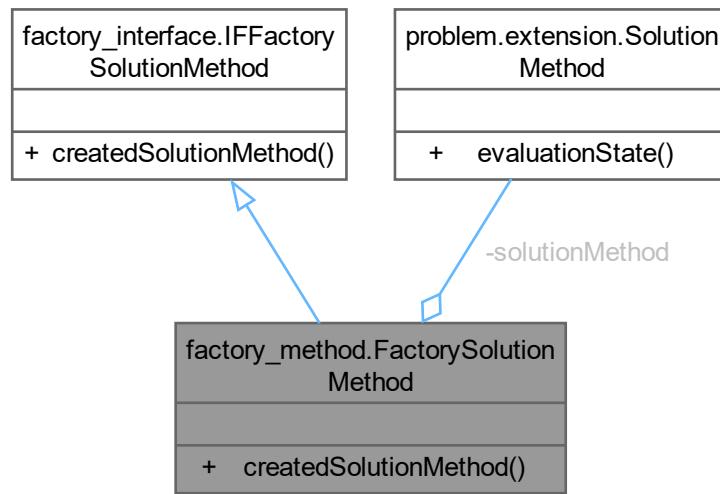


Diagrama de colaboración de factory_method.FactorySolutionMethod:



Métodos públicos

- `SolutionMethod createdSolutionMethod (TypeSolutionMethod method)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

Atributos privados

- `SolutionMethod solutionMethod`

6.33.1. Documentación de funciones miembro

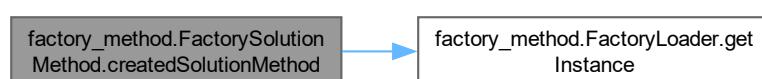
6.33.1.1. `createdSolutionMethod()`

```

SolutionMethod factory_method.FactorySolutionMethod.createdSolutionMethod (
    TypeSolutionMethod method) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
  
```

Implementa [factory_interface.IFFactorySolutionMethod](#).

Gráfico de llamadas de esta función:



6.33.2. Documentación de datos miembro

6.33.2.1. solutionMethod

```
SolutionMethod factory_method.FactorySolutionMethod.solutionMethod [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/factory_method/FactorySolutionMethod.java](#)

6.34. Referencia de la clase evolutionary_algorithms.complement.FatherSelection

Diagrama de herencia de `evolutionary_algorithms.complement.FatherSelection`

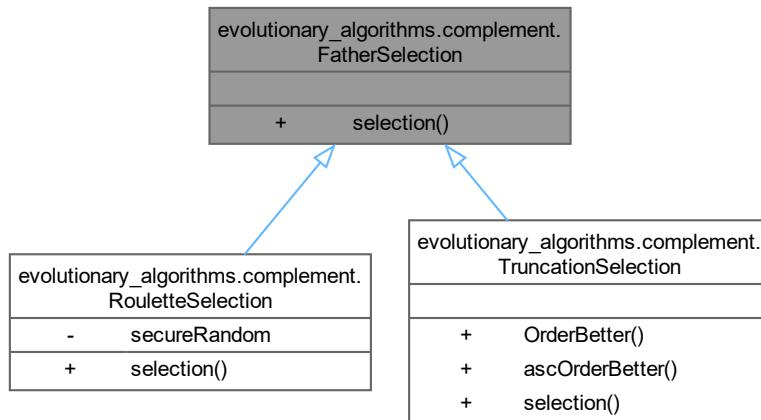
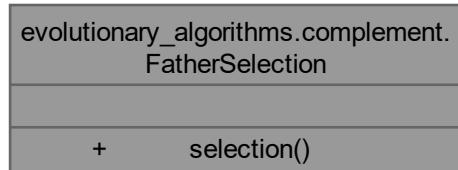


Diagrama de colaboración de `evolutionary_algorithms.complement.FatherSelection`:



Métodos públicos

- abstract List< State > selection (List< State > listState, int truncation)

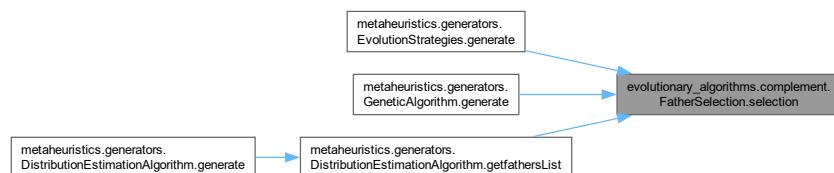
6.34.1. Documentación de funciones miembro

6.34.1.1. selection()

```
abstract List< State > evolutionary_algorithms.complement.FatherSelection.selection (
    List< State > listState,
    int truncation) [abstract]
```

Reimplementado en [evolutionary_algorithms.complement.RouletteSelection](#) y [evolutionary_algorithms.complement.TruncationSelection](#)

Gráfico de llamadas a esta función:



La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/FatherSelection.java](#)

6.35. Referencia de la clase

evolutionary_algorithms.complement.GenerationalReplace

Diagrama de herencia de `evolutionary_algorithms.complement.GenerationalReplace`

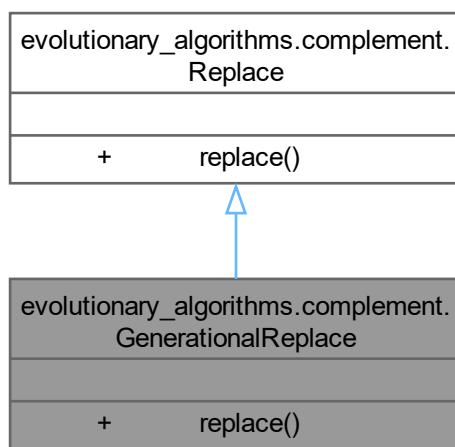
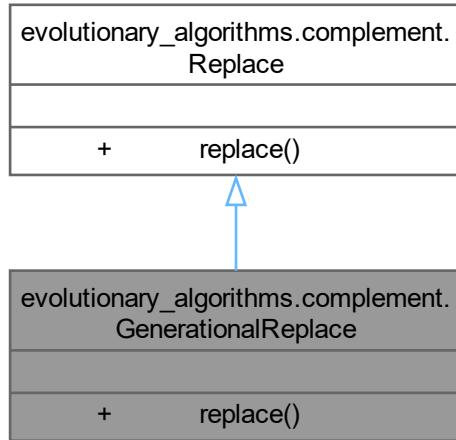


Diagrama de colaboración de evolutionary_algorithms.complement.GenerationalReplace:



Métodos públicos

- `List< State > replace (State stateCandidate, List< State > listState) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.35.1. Documentación de funciones miembro

6.35.1.1. replace()

```
List< State > evolutionary_algorithms.complement.GenerationalReplace.replace (
    State stateCandidate,
    List< State > listState) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [evolutionary_algorithms.complement.Replace](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/evolutionary_algorithms/complement/GenerationalReplace.java`

6.36. Referencia de la clase metaheuristics.generators.Generator

Diagrama de herencia de metaheuristics.generators.Generator

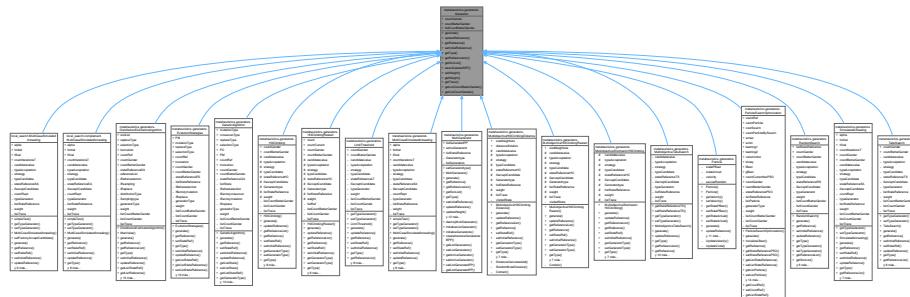
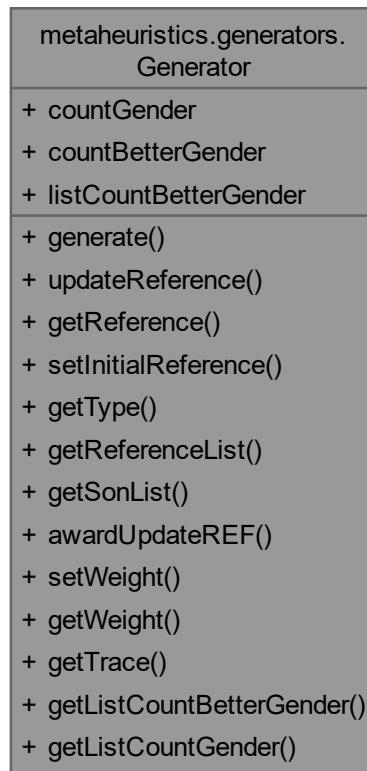


Diagrama de colaboración de metaheuristics.generators.Generator:



Métodos públicos

- abstract `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

- abstract void `updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- abstract `State getReference ()`
- abstract void `setInitialReference (State stateInitialRef)`
- abstract `GeneratorType getType ()`
- abstract `List< State > getReferenceList ()`
- abstract `List< State > getSonList ()`
- abstract boolean `awardUpdateREF (State stateCandidate)`
- abstract void `setWeight (float weight)`
- abstract float `getWeight ()`
- abstract float[] `getTrace ()`
- abstract int[] `getListCountBetterGender ()`
- abstract int[] `getListCountGender ()`

Atributos públicos

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.36.1. Documentación de funciones miembro

6.36.1.1. awardUpdateREF()

```
abstract boolean metaheuristics.generators.Generator.awardUpdateREF (
    State stateCandidate) [abstract]
```

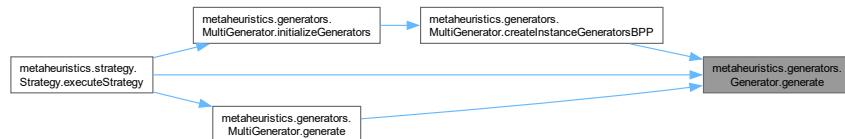
Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

6.36.1.2. generate()

```
abstract State metaheuristics.generators.Generator.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException [abstract]
```

Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

Gráfico de llamadas a esta función:



6.36.1.3. getListCountBetterGender()

```
abstract int[] metaheuristics.generators.Generator.getListCountBetterGender () [abstract]
```

Reimplementado en [local_search.complement.MultiCaseSimulatedAnnealing](#), [local_search.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.DistributionEstimationAlgorithm](#), [metaheuristics.generators.EvolutionStrategies](#), [metaheuristics.generators.HillClimbing](#), [metaheuristics.generators.HillClimbingRestart](#), [metaheuristics.generators.LimitThreshold](#), [metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.MultiGenerator](#), [metaheuristics.generators.MultiobjectiveHillClimbing](#), [metaheuristics.generators.MultiobjectiveHillClimbingRestart](#), [metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [metaheuristics.generators.MultiobjectiveTabuSearch](#), [metaheuristics.generators.Particle](#), [metaheuristics.generators.ParticleSwarmOptimization](#), [metaheuristics.generators.RandomSearch](#), [metaheuristics.generators.SimulatedAnnealing](#) y [metaheuristics.generators.TabuSearch](#).

Gráfico de llamadas a esta función:

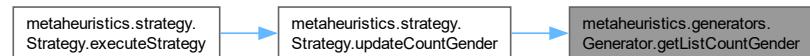


6.36.1.4. getListCountGender()

```
abstract int[] metaheuristics.generators.Generator.getListCountGender () [abstract]
```

Reimplementado en [local_search.complement.MultiCaseSimulatedAnnealing](#), [local_search.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.DistributionEstimationAlgorithm](#), [metaheuristics.generators.EvolutionStrategies](#), [metaheuristics.generators.HillClimbing](#), [metaheuristics.generators.HillClimbingRestart](#), [metaheuristics.generators.LimitThreshold](#), [metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.MultiGenerator](#), [metaheuristics.generators.MultiobjectiveHillClimbing](#), [metaheuristics.generators.MultiobjectiveHillClimbingRestart](#), [metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [metaheuristics.generators.MultiobjectiveTabuSearch](#), [metaheuristics.generators.Particle](#), [metaheuristics.generators.ParticleSwarmOptimization](#), [metaheuristics.generators.RandomSearch](#), [metaheuristics.generators.SimulatedAnnealing](#) y [metaheuristics.generators.TabuSearch](#).

Gráfico de llamadas a esta función:



6.36.1.5. `getReference()`

```
abstract State metaheuristics.generators.Generator.getReference () [abstract]
```

Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

6.36.1.6. `getReferenceList()`

```
abstract List< State > metaheuristics.generators.Generator.getReferenceList () [abstract]
```

Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

6.36.1.7. `getSonList()`

```
abstract List< State > metaheuristics.generators.Generator.getSonList () [abstract]
```

Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

6.36.1.8. `getTrace()`

```
abstract float[] metaheuristics.generators.Generator.getTrace () [abstract]
```

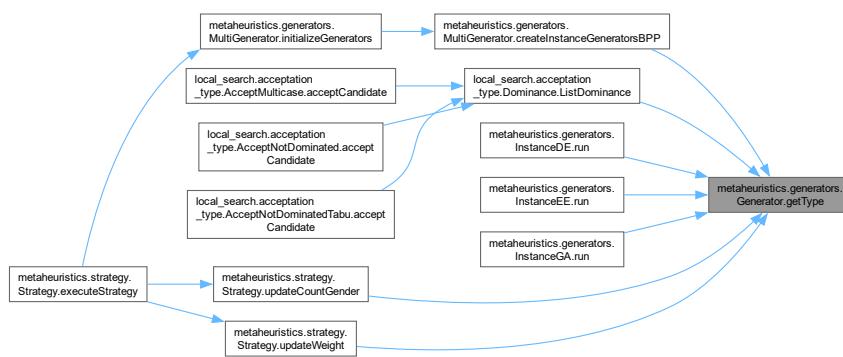
Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

6.36.1.9. `getType()`

```
abstract GeneratorType metaheuristics.generators.Generator.getType () [abstract]
```

Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

Gráfico de llamadas a esta función:



6.36.1.10. `getWeight()`

```
abstract float metaheuristics.generators.Generator.getWeight () [abstract]
```

Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

6.36.1.11. `setInitialReference()`

```
abstract void metaheuristics.generators.Generator.setInitialReference (
    State stateInitialRef) [abstract]
```

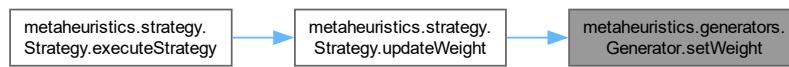
Reimplementado en `local_search.complement.MultiCaseSimulatedAnnealing`, `local_search.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.DistributionEstimationAlgorithm`, `metaheuristics.generators.EvolutionStrategies`, `metaheuristics.generators.metaheuristics.generators.HillClimbing`, `metaheuristics.generators.HillClimbingRestart`, `metaheuristics.generators.LimitThreshold`, `metaheuristics.generators.MultiCaseSimulatedAnnealing`, `metaheuristics.generators.MultiGenerator`, `metaheuristics.generators.MultiobjectiveHillClimbing`, `metaheuristics.generators.MultiobjectiveHillClimbingRestart`, `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`, `metaheuristics.generators.MultiobjectiveTabuSearch`, `metaheuristics.generators.Particle`, `metaheuristics.generators.ParticleSwarmOptimization`, `metaheuristics.generators.RandomSearch`, `metaheuristics.generators.SimulatedAnnealing` y `metaheuristics.generators.TabuSearch`.

6.36.1.12. setWeight()

```
abstract void metaheuristics.generators.Generator.setWeight (
    float weight) [abstract]
```

Reimplementado en [local_search.complement.MultiCaseSimulatedAnnealing](#), [local_search.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.DistributionEstimationAlgorithm](#), [metaheuristics.generators.EvolutionStrategies](#), [metaheuristics.generators.metaheuristics.generators.HillClimbing](#), [metaheuristics.generators.HillClimbingRestart](#), [metaheuristics.generators.LimitThreshold](#), [metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.MultiGenerator](#), [metaheuristics.generators.MultiobjectiveHillClimbingRestart](#), [metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [metaheuristics.generators.MultiobjectiveTabuSearch](#), [metaheuristics.generators.Particle](#), [metaheuristics.generators.ParticleSwarmOptimization](#), [metaheuristics.generators.RandomSearch](#), [metaheuristics.generators.SimulatedAnnealing](#) y [metaheuristics.generators.TabuSearch](#).

Gráfico de llamadas a esta función:



6.36.1.13. updateReference()

```
abstract void metaheuristics.generators.Generator.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException [abstract]
```

Reimplementado en [local_search.complement.MultiCaseSimulatedAnnealing](#), [local_search.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.DistributionEstimationAlgorithm](#), [metaheuristics.generators.EvolutionStrategies](#), [metaheuristics.generators.metaheuristics.generators.HillClimbing](#), [metaheuristics.generators.HillClimbingRestart](#), [metaheuristics.generators.LimitThreshold](#), [metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [metaheuristics.generators.MultiGenerator](#), [metaheuristics.generators.MultiobjectiveHillClimbingRestart](#), [metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [metaheuristics.generators.MultiobjectiveTabuSearch](#), [metaheuristics.generators.Particle](#), [metaheuristics.generators.ParticleSwarmOptimization](#), [metaheuristics.generators.RandomSearch](#), [metaheuristics.generators.SimulatedAnnealing](#) y [metaheuristics.generators.TabuSearch](#).

6.36.2. Documentación de datos miembro

6.36.2.1. countBetterGender

```
int metaheuristics.generators.Generator.countBetterGender
```

6.36.2.2. countGender

```
int metaheuristics.generators.Generator.countGender
```

6.36.2.3. `listCountBetterGender`

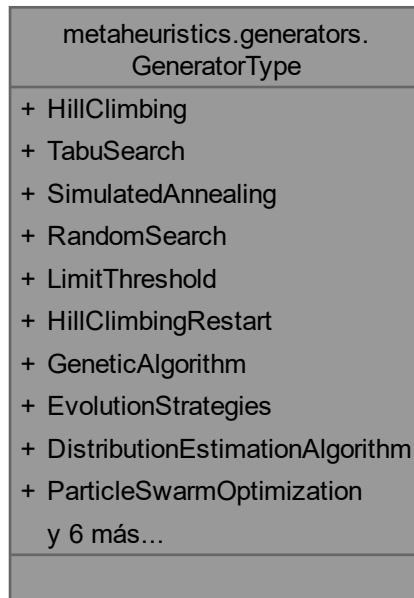
```
int [ ] metaheuristics.generators.Generator.listCountBetterGender
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/Generator.java](#)

6.37. Referencia de la enumeración `metaheuristics.generators.GeneratorType`

Diagrama de colaboración de `metaheuristics.generators.GeneratorType`:



Atributos públicos

- [HillClimbing](#)
- [TabuSearch](#)
- [SimulatedAnnealing](#)
- [RandomSearch](#)
- [LimitThreshold](#)
- [HillClimbingRestart](#)
- [GeneticAlgorithm](#)
- [EvolutionStrategies](#)
- [DistributionEstimationAlgorithm](#)
- [ParticleSwarmOptimization](#)

- [MultiGenerator](#)
- [MultiobjectiveTabuSearch](#)
- [MultiobjectiveStochasticHillClimbing](#)
- [MultiCaseSimulatedAnnealing](#)
- [MultiobjectiveHillClimbingRestart](#)
- [MultiobjectiveHillClimbingDistance](#)

6.37.1. Documentación de datos miembro

6.37.1.1. DistributionEstimationAlgorithm

```
metaheuristics.generators.GeneratorType.DistributionEstimationAlgorithm
```

6.37.1.2. EvolutionStrategies

```
metaheuristics.generators.GeneratorType.EvolutionStrategies
```

6.37.1.3. GeneticAlgorithm

```
metaheuristics.generators.GeneratorType.GeneticAlgorithm
```

6.37.1.4. HillClimbing

```
metaheuristics.generators.GeneratorType.HillClimbing
```

6.37.1.5. HillClimbingRestart

```
metaheuristics.generators.GeneratorType.HillClimbingRestart
```

6.37.1.6. LimitThreshold

```
metaheuristics.generators.GeneratorType.LimitThreshold
```

6.37.1.7. MultiCaseSimulatedAnnealing

```
metaheuristics.generators.GeneratorType.MultiCaseSimulatedAnnealing
```

6.37.1.8. MultiGenerator

```
metaheuristics.generators.GeneratorType.MultiGenerator
```

6.37.1.9. MultiobjectiveHillClimbingDistance

```
metaheuristics.generators.GeneratorType.MultiobjectiveHillClimbingDistance
```

6.37.1.10. MultiobjectiveHillClimbingRestart

```
metaheuristics.generators.GeneratorType.MultiobjectiveHillClimbingRestart
```

6.37.1.11. MultiobjectiveStochasticHillClimbing

```
metaheuristics.generators.GeneratorType.MultiobjectiveStochasticHillClimbing
```

6.37.1.12. MultiobjectiveTabuSearch

```
metaheuristics.generators.GeneratorType.MultiobjectiveTabuSearch
```

6.37.1.13. ParticleSwarmOptimization

```
metaheuristics.generators.GeneratorType.ParticleSwarmOptimization
```

6.37.1.14. RandomSearch

```
metaheuristics.generators.GeneratorType.RandomSearch
```

6.37.1.15. SimulatedAnnealing

```
metaheuristics.generators.GeneratorType.SimulatedAnnealing
```

6.37.1.16. TabuSearch

```
metaheuristics.generators.GeneratorType.TabuSearch
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/GeneratorType.java](#)

6.38. Referencia de la clase metaheuristics.generators.GeneticAlgorithm

Diagrama de herencia de metaheuristics.generators.GeneticAlgorithm

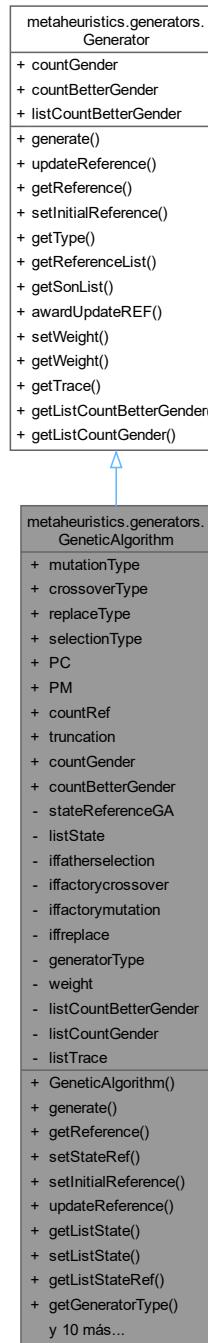
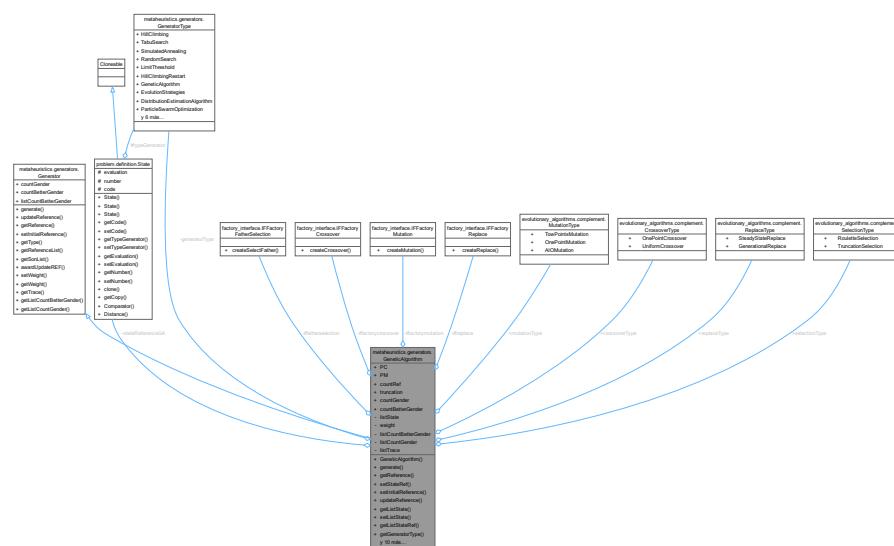


Diagrama de colaboración de metaheuristics.generators.GeneticAlgorithm:



Métodos públicos

- `GeneticAlgorithm ()`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `State getReference ()`
- `void setStateRef (State stateRef)`
- `void setInitialReference (State statelInitialRef)`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `List< State > getListState ()`
- `void setListState (List< State > listState)`
- `List< State > getListStateRef ()`
- `GeneratorType getGeneratorType ()`
- `void setGeneratorType (GeneratorType generatorType)`
- `GeneratorType getType ()`
- `List< State > getReferenceList ()`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `int[] getListCountBetterGender ()`
- `int[] getListCountGender ()`
- `float[] getTrace ()`

Atributos públicos estáticos

- static `MutationType mutationType`
- static `CrossoverType crossoverType`
- static `ReplaceType replaceType`

- static SelectionType selectionType
- static double PC
- static double PM
- static int countRef = 0
- static int truncation
- static int countGender = 0
- static int countBetterGender = 0

Atributos privados

- State stateReferenceGA
- List< State > listState = new ArrayList<State>()
- IFFactoryFatherSelection iffatherselection
- IFFactoryCrossover iffactorycrossover
- IFFactoryMutation iffactorymutation
- IFFactoryReplace iffreplace
- GeneratorType generatorType
- float weight
- int[] listCountBetterGender = new int[10]
- int[] listCountGender = new int[10]
- float[] listTrace = new float[1200000]

Otros miembros heredados

Atributos públicos heredados de metaheuristics.generators.Generator

- int countGender
- int countBetterGender
- int[] listCountBetterGender

6.38.1. Documentación de constructores y destructores

6.38.1.1. GeneticAlgorithm()

```
metaheuristics.generators.GeneticAlgorithm.GeneticAlgorithm ()
```

Gráfico de llamadas de esta función:

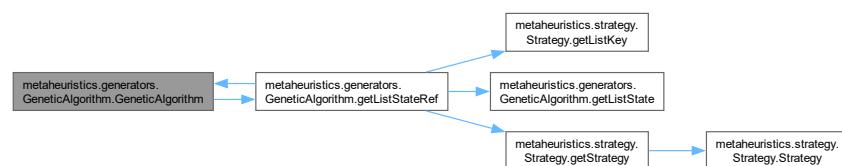
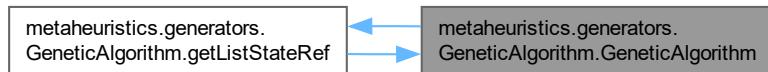


Gráfico de llamadas a esta función:



6.38.2. Documentación de funciones miembro

6.38.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.GeneticAlgorithm.awardUpdateREF (\n    State stateCandidate)
```

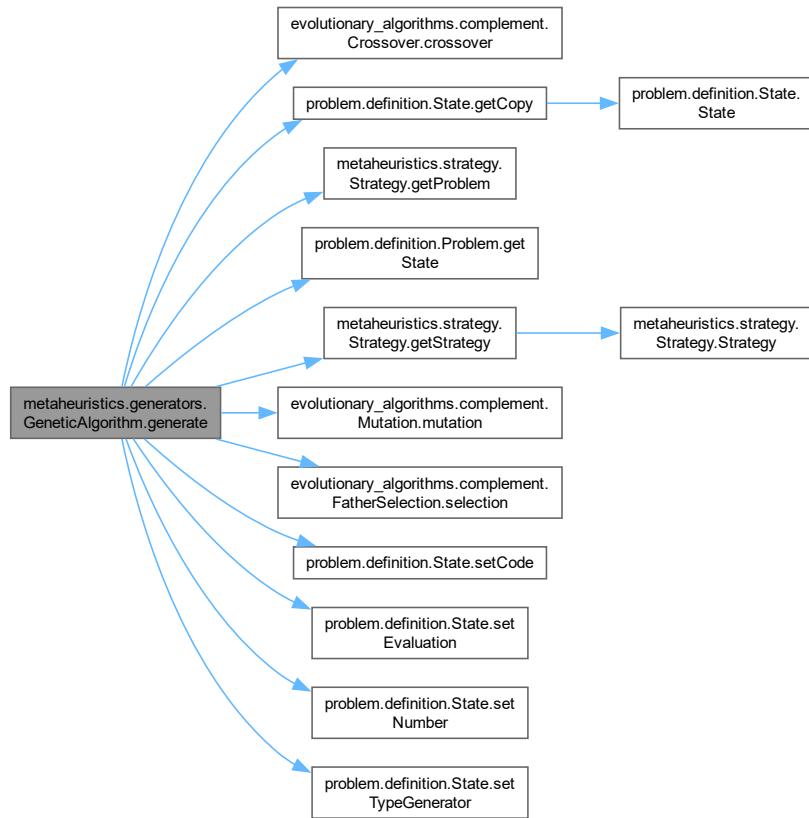
Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.2. generate()

```
State metaheuristics.generators.GeneticAlgorithm.generate (\n    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←\nNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,\nNoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.38.2.3. `getGeneratorType()`

```
GeneratorType metaheuristics.generators.GeneticAlgorithm.getGeneratorType ()
```

6.38.2.4. `getListCountBetterGender()`

```
int[] metaheuristics.generators.GeneticAlgorithm.getListCountBetterGender ()
```

Reimplementado de `metaheuristics.generators.Generator`.

6.38.2.5. `getListCountGender()`

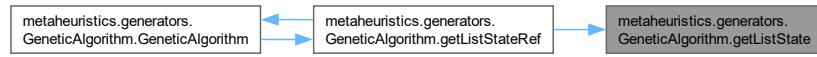
```
int[] metaheuristics.generators.GeneticAlgorithm.getListCountGender ()
```

Reimplementado de `metaheuristics.generators.Generator`.

6.38.2.6. getListState()

```
List< State > metaheuristics.generators.GeneticAlgorithm.getListState ()
```

Gráfico de llamadas a esta función:



6.38.2.7. getListStateRef()

```
List< State > metaheuristics.generators.GeneticAlgorithm.getListStateRef ()
```

Gráfico de llamadas de esta función:

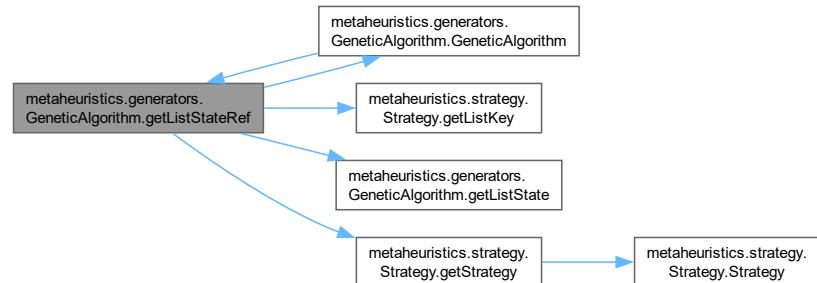
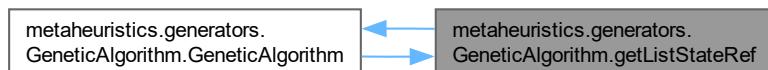


Gráfico de llamadas a esta función:

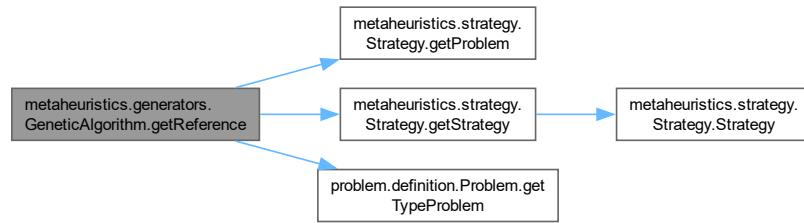


6.38.2.8. getReference()

```
State metaheuristics.generators.GeneticAlgorithm.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.38.2.9. `getReferenceList()`

```
List< State > metaheuristics.generators.GeneticAlgorithm.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.10. `getSonList()`

```
List< State > metaheuristics.generators.GeneticAlgorithm.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.11. `getTrace()`

```
float[ ] metaheuristics.generators.GeneticAlgorithm.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.12. `getType()`

```
GeneratorType metaheuristics.generators.GeneticAlgorithm.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.13. `getWeight()`

```
float metaheuristics.generators.GeneticAlgorithm.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.14. `setGeneratorType()`

```
void metaheuristics.generators.GeneticAlgorithm.setGeneratorType (
    GeneratorType generatorType)
```

6.38.2.15. `setInitialReference()`

```
void metaheuristics.generators.GeneticAlgorithm.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.16. `setListState()`

```
void metaheuristics.generators.GeneticAlgorithm.setListState (
    List< State > listState)
```

6.38.2.17. `setStateRef()`

```
void metaheuristics.generators.GeneticAlgorithm.setStateRef (
    State stateRef)
```

6.38.2.18. `setWeight()`

```
void metaheuristics.generators.GeneticAlgorithm.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.38.2.19. `updateReference()`

```
void metaheuristics.generators.GeneticAlgorithm.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.38.3. Documentación de datos miembro

6.38.3.1. `countBetterGender`

```
int metaheuristics.generators.GeneticAlgorithm.countBetterGender = 0 [static]
```

6.38.3.2. countGender

```
int metaheuristics.generators.GeneticAlgorithm.countGender = 0 [static]
```

6.38.3.3. countRef

```
int metaheuristics.generators.GeneticAlgorithm.countRef = 0 [static]
```

6.38.3.4. crossoverType

```
CrossoverType metaheuristics.generators.GeneticAlgorithm.crossoverType [static]
```

6.38.3.5. generatorType

```
GeneratorType metaheuristics.generators.GeneticAlgorithm.generatorType [private]
```

6.38.3.6. iffactorycrossover

```
IFFactoryCrossover metaheuristics.generators.GeneticAlgorithm.iffactorycrossover [private]
```

6.38.3.7. iffactorymutation

```
IFFactoryMutation metaheuristics.generators.GeneticAlgorithm.iffactorymutation [private]
```

6.38.3.8. iffatherselection

```
IFFactoryFatherSelection metaheuristics.generators.GeneticAlgorithm.iffatherselection [private]
```

6.38.3.9. iffreplace

```
IFFactoryReplace metaheuristics.generators.GeneticAlgorithm.iffreplace [private]
```

6.38.3.10. listCountBetterGender

```
int [] metaheuristics.generators.GeneticAlgorithm.listCountBetterGender = new int[10] [private]
```

6.38.3.11. listCountGender

```
int [] metaheuristics.generators.GeneticAlgorithm.listCountGender = new int[10] [private]
```

6.38.3.12. **listState**

```
List<State> metaheuristics.generators.GeneticAlgorithm.listState = new ArrayList<State>()  
[private]
```

6.38.3.13. **listTrace**

```
float [] metaheuristics.generators.GeneticAlgorithm.listTrace = new float[1200000] [private]
```

6.38.3.14. **mutationType**

```
MutationType metaheuristics.generators.GeneticAlgorithm.mutationType [static]
```

6.38.3.15. **PC**

```
double metaheuristics.generators.GeneticAlgorithm.PC [static]
```

6.38.3.16. **PM**

```
double metaheuristics.generators.GeneticAlgorithm.PM [static]
```

6.38.3.17. **replaceType**

```
ReplaceType metaheuristics.generators.GeneticAlgorithm.replaceType [static]
```

6.38.3.18. **selectionType**

```
SelectionType metaheuristics.generators.GeneticAlgorithm.selectionType [static]
```

6.38.3.19. **stateReferenceGA**

```
State metaheuristics.generators.GeneticAlgorithm.stateReferenceGA [private]
```

6.38.3.20. **truncation**

```
int metaheuristics.generators.GeneticAlgorithm.truncation [static]
```

6.38.3.21. **weight**

```
float metaheuristics.generators.GeneticAlgorithm.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/GeneticAlgorithm.java](#)

6.39. Referencia de la clase local_search.candidate_type.GreaterCandidate

Diagrama de herencia de local_search.candidate_type.GreaterCandidate

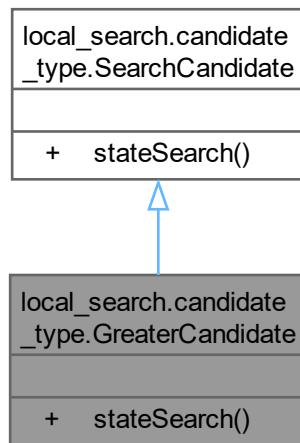
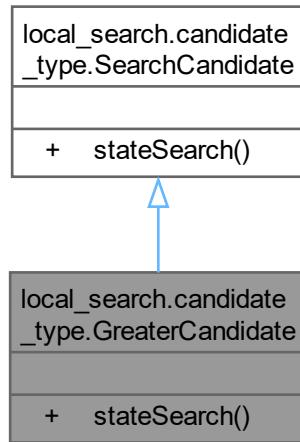


Diagrama de colaboración de local_search.candidate_type.GreaterCandidate:



Métodos públicos

- **State stateSearch (List< State > listNeighborhood) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**

6.39.1. Documentación de funciones miembro

6.39.1.1. stateSearch()

```
State local_search.candidate_type.GreaterCandidate.stateSearch (
    List< State > listNeighborhood) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [local_search.candidate_type.SearchCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/candidate_type/GreaterCandidate.java](#)

6.40. Referencia de la clase metaheuristics.generators.HillClimbing

Diagrama de herencia de metaheuristics.generators.HillClimbing

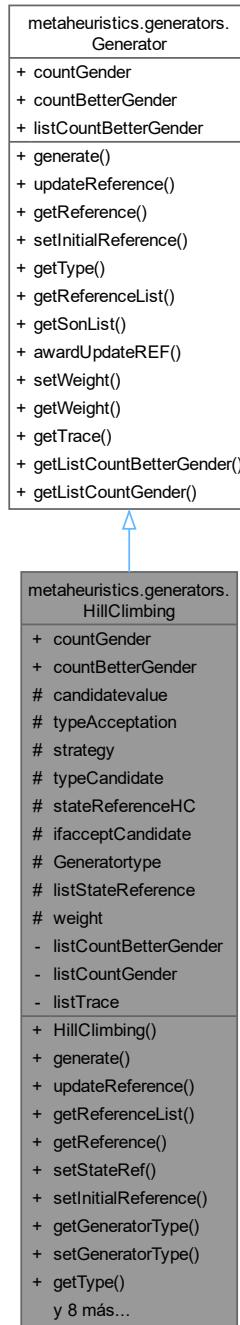
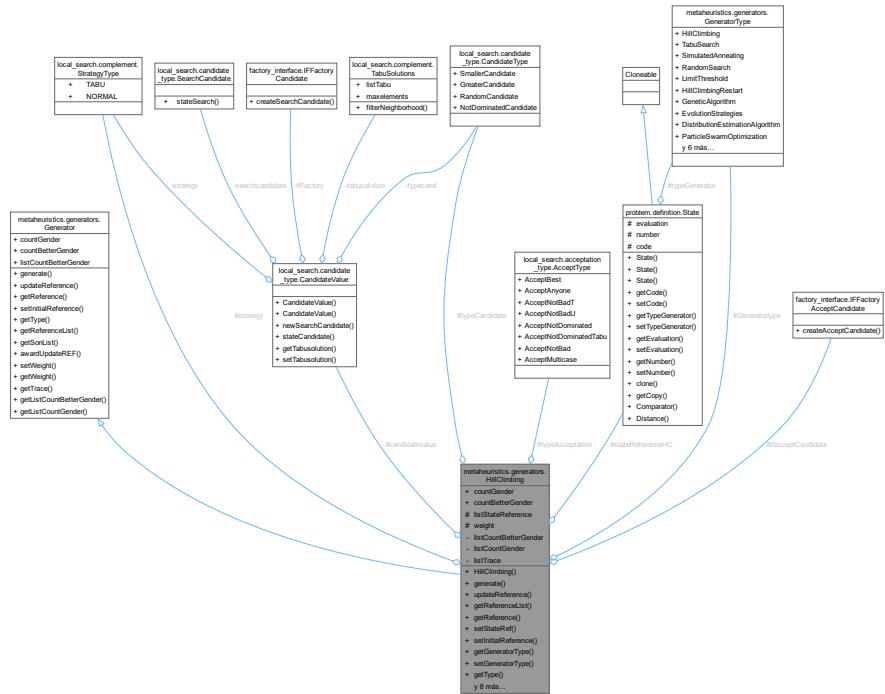


Diagrama de colaboración de metaheuristics.generators.HillClimbing:



Métodos públicos

- **HillClimbing ()**
 - **State generate (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**
 - **void updateReference (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**
 - **List< State > getReferenceList ()**
 - **State getReference ()**
 - **void setStateRef (State stateRef)**
 - **void setInitialReference (State stateInitialRef)**
 - **GeneratorType getGeneratorType ()**
 - **void setGeneratorType (GeneratorType Generatortype)**
 - **GeneratorType getType ()**
 - **List< State > getSonList ()**
 - **void setTypeCandidate (CandidateType typeCandidate)**
 - **boolean awardUpdateREF (State stateCandidate)**
 - **float getWeight ()**
 - **void setWeight (float weight)**
 - **int[] getListCountBetterGender ()**
 - **int[] getListCountGender ()**
 - **float[] getTrace ()**

Atributos públicos estáticos

- static int countGender = 0
 - static int countBetterGender = 0

Atributos protegidos

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceHC`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- `GeneratorType Generatortype`
- `List< State > listStateReference = new ArrayList<State>()`
- `float weight`

Atributos privados

- `int[] listCountBetterGender = new int[10]`
- `int[] listCountGender = new int[10]`
- `float[] listTrace = new float[1200000]`

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

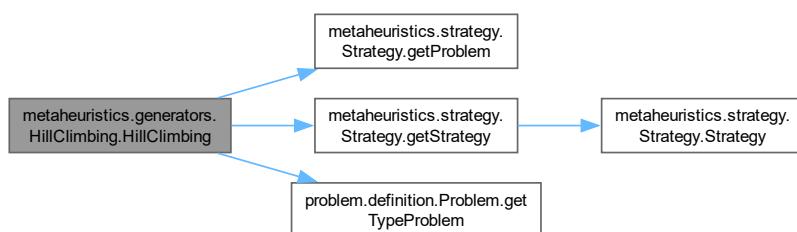
- `int countGender`
- `int countBetterGender`
- `int[] listCountBetterGender`

6.40.1. Documentación de constructores y destructores

6.40.1.1. [HillClimbing\(\)](#)

```
metaheuristics.generators.HillClimbing.HillClimbing ()
```

Gráfico de llamadas de esta función:



6.40.2. Documentación de funciones miembro

6.40.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.HillClimbing.awardUpdateREF (
    State stateCandidate)
```

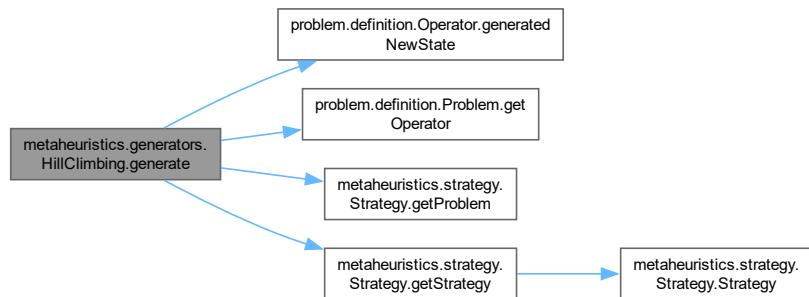
Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.2. generate()

```
State metaheuristics.generators.HillClimbing.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.40.2.3. getGeneratorType()

```
GeneratorType metaheuristics.generators.HillClimbing.getGeneratorType ()
```

6.40.2.4. getListCountBetterGender()

```
int[] metaheuristics.generators.HillClimbing.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.5. getListCountGender()

```
int[] metaheuristics.generators.HillClimbing.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.6. `getReference()`

```
State metaheuristics.generators.HillClimbing.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.7. `getReferenceList()`

```
List< State > metaheuristics.generators.HillClimbing.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.8. `getSonList()`

```
List< State > metaheuristics.generators.HillClimbing.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.9. `getTrace()`

```
float[] metaheuristics.generators.HillClimbing.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.10. `getType()`

```
GeneratorType metaheuristics.generators.HillClimbing.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.11. `getWeight()`

```
float metaheuristics.generators.HillClimbing.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.12. `setGeneratorType()`

```
void metaheuristics.generators.HillClimbing.setGeneratorType (
    GeneratorType Generatortype)
```

6.40.2.13. `setInitialReference()`

```
void metaheuristics.generators.HillClimbing.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.14. `setStateRef()`

```
void metaheuristics.generators.HillClimbing.setStateRef (
    State stateRef)
```

6.40.2.15. `setTypeCandidate()`

```
void metaheuristics.generators.HillClimbing.setTypeCandidate (
    CandidateType typeCandidate)
```

6.40.2.16. `setWeight()`

```
void metaheuristics.generators.HillClimbing.setWeight (
    float weight)
```

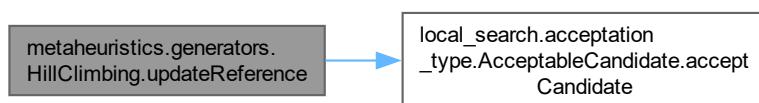
Reimplementado de [metaheuristics.generators.Generator](#).

6.40.2.17. `updateReference()`

```
void metaheuristics.generators.HillClimbing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.40.3. Documentación de datos miembro

6.40.3.1. `candidatevalue`

```
CandidateValue metaheuristics.generators.HillClimbing.candidatevalue [protected]
```

6.40.3.2. `countBetterGender`

```
int metaheuristics.generators.HillClimbing.countBetterGender = 0 [static]
```

6.40.3.3. countGender

```
int metaheuristics.generators.HillClimbing.countGender = 0 [static]
```

6.40.3.4. GeneratorType

```
GeneratorType metaheuristics.generators.HillClimbing.GeneratorType [protected]
```

6.40.3.5. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.HillClimbing.ifacceptCandidate [protected]
```

6.40.3.6. listCountBetterGender

```
int [] metaheuristics.generators.HillClimbing.listCountBetterGender = new int[10] [private]
```

6.40.3.7. listCountGender

```
int [] metaheuristics.generators.HillClimbing.listCountGender = new int[10] [private]
```

6.40.3.8. listStateReference

```
List<State> metaheuristics.generators.HillClimbing.listStateReference = new ArrayList<State>()  
[protected]
```

6.40.3.9. listTrace

```
float [] metaheuristics.generators.HillClimbing.listTrace = new float[1200000] [private]
```

6.40.3.10. stateReferenceHC

```
State metaheuristics.generators.HillClimbing.stateReferenceHC [protected]
```

6.40.3.11. strategy

```
StrategyType metaheuristics.generators.HillClimbing.strategy [protected]
```

6.40.3.12. typeAcceptation

```
AcceptType metaheuristics.generators.HillClimbing.typeAcceptation [protected]
```

6.40.3.13. typeCandidate

```
CandidateType metaheuristics.generators.HillClimbing.typeCandidate [protected]
```

6.40.3.14. weight

```
float metaheuristics.generators.HillClimbing.weight [protected]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/HillClimbing.java](#)

6.41. Referencia de la clase metaheuristics.generators.HillClimbingRestart

Diagrama de herencia de metaheuristics.generators.HillClimbingRestart

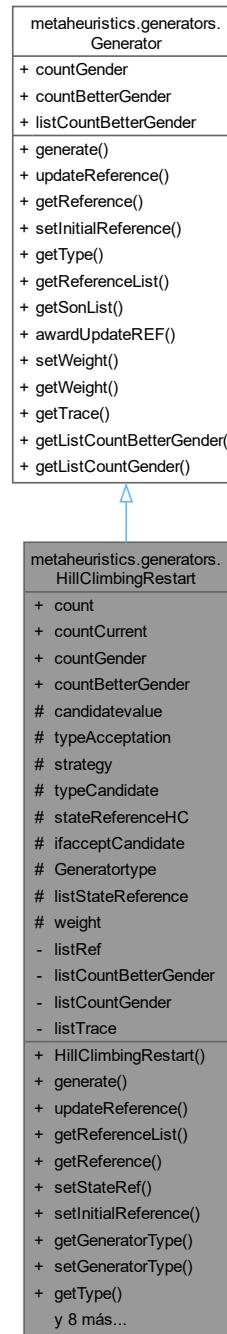
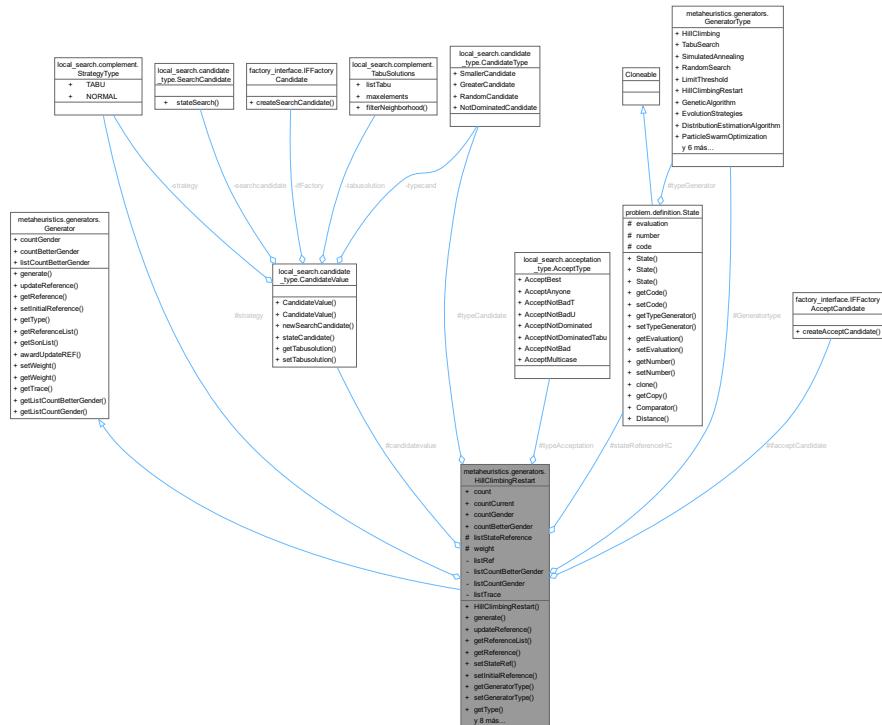


Diagrama de colaboración de metaheuristics.generators.HillClimbingRestart:



Métodos públicos

- **HillClimbingRestart ()**
- **State generate (Integer operatornumber)** throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- **void updateReference (State stateCandidate, Integer countIterationsCurrent)** throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- **List< State > getReferenceList ()**
- **State getReference ()**
- **void setStateRef (State stateRef)**
- **void setInitialReference (State statelInitialRef)**
- **GeneratorType getGeneratorType ()**
- **void setGeneratorType (GeneratorType Generatortype)**
- **GeneratorType getType ()**
- **List< State > getSonList ()**
- **void setTypeCandidate (CandidateType typeCandidate)**
- **boolean awardUpdateREF (State stateCandidate)**
- **float getWeight ()**
- **void setWeight (float weight)**
- **int[] getListCountBetterGender ()**
- **int[] getListCountGender ()**
- **float[] getTrace ()**

Atributos públicos estáticos

- static int `count`
- static int `countCurrent`
- static int `countGender` = 0
- static int `countBetterGender` = 0

Atributos protegidos

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceHC`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- `GeneratorType Generatortype`
- `List< State > listStateReference = new ArrayList<State>()`
- float `weight`

Atributos privados

- `List< State > listRef = new ArrayList<State>()`
- `int[] listCountBetterGender = new int[10]`
- `int[] listCountGender = new int[10]`
- `float[] listTrace = new float[1200000]`

Otros miembros heredados

Atributos públicos heredados de `metaheuristics.generators.Generator`

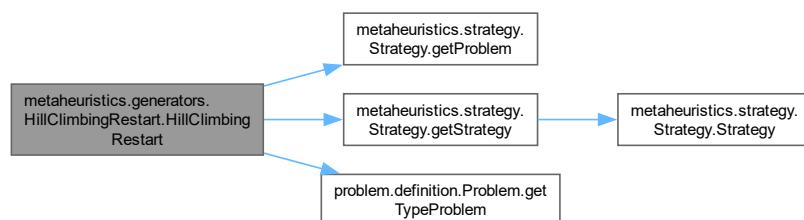
- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.41.1. Documentación de constructores y destructores

6.41.1.1. `HillClimbingRestart()`

```
metaheuristics.generators.HillClimbingRestart.HillClimbingRestart ()
```

Gráfico de llamadas de esta función:



6.41.2. Documentación de funciones miembro

6.41.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.HillClimbingRestart.awardUpdateREF (
    State stateCandidate)
```

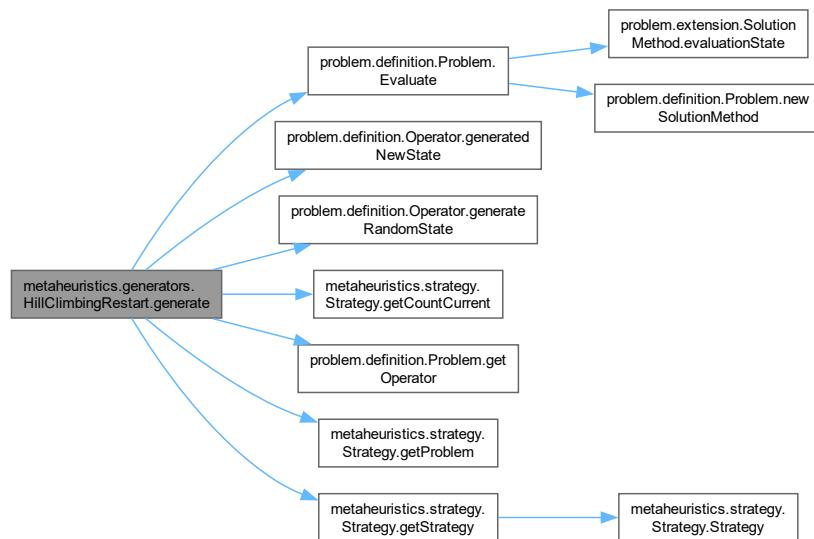
Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.2. generate()

```
State metaheuristics.generators.HillClimbingRestart.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.41.2.3. getGeneratorType()

```
GeneratorType metaheuristics.generators.HillClimbingRestart.getGeneratorType ()
```

6.41.2.4. getListCountBetterGender()

```
int[] metaheuristics.generators.HillClimbingRestart.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.5. getListCountGender()

```
int[ ] metaheuristics.generators.HillClimbingRestart.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.6. getReference()

```
State metaheuristics.generators.HillClimbingRestart.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.7. getReferenceList()

```
List< State > metaheuristics.generators.HillClimbingRestart.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.8. getSonList()

```
List< State > metaheuristics.generators.HillClimbingRestart.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.9. getTrace()

```
float[ ] metaheuristics.generators.HillClimbingRestart.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.10. getType()

```
GeneratorType metaheuristics.generators.HillClimbingRestart.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.11. getWeight()

```
float metaheuristics.generators.HillClimbingRestart.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.12. setGeneratorType()

```
void metaheuristics.generators.HillClimbingRestart.setGeneratorType (
    GeneratorType Generatortype)
```

6.41.2.13. `setInitialReference()`

```
void metaheuristics.generators.HillClimbingRestart.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.14. `setStateRef()`

```
void metaheuristics.generators.HillClimbingRestart.setStateRef (
    State stateRef)
```

6.41.2.15. `setTypeCandidate()`

```
void metaheuristics.generators.HillClimbingRestart.setTypeCandidate (
    CandidateType typeCandidate)
```

6.41.2.16. `setWeight()`

```
void metaheuristics.generators.HillClimbingRestart.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.41.2.17. `updateReference()`

```
void metaheuristics.generators.HillClimbingRestart.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.41.3. Documentación de datos miembro

6.41.3.1. candidatevalue

```
CandidateValue metaheuristics.generators.HillClimbingRestart.candidatevalue [protected]
```

6.41.3.2. count

```
int metaheuristics.generators.HillClimbingRestart.count [static]
```

6.41.3.3. countBetterGender

```
int metaheuristics.generators.HillClimbingRestart.countBetterGender = 0 [static]
```

6.41.3.4. countCurrent

```
int metaheuristics.generators.HillClimbingRestart.countCurrent [static]
```

6.41.3.5. countGender

```
int metaheuristics.generators.HillClimbingRestart.countGender = 0 [static]
```

6.41.3.6. Generatortype

```
GeneratorType metaheuristics.generators.HillClimbingRestart.Generatortype [protected]
```

6.41.3.7. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.HillClimbingRestart.ifacceptCandidate [protected]
```

6.41.3.8. listCountBetterGender

```
int [] metaheuristics.generators.HillClimbingRestart.listCountBetterGender = new int[10] [private]
```

6.41.3.9. listCountGender

```
int [] metaheuristics.generators.HillClimbingRestart.listCountGender = new int[10] [private]
```

6.41.3.10. listRef

```
List<State> metaheuristics.generators.HillClimbingRestart.listRef = new ArrayList<State>()  
[private]
```

6.41.3.11. **listStateReference**

```
List<State> metaheuristics.generators.HillClimbingRestart.listStateReference = new Array←  
List<State>() [protected]
```

6.41.3.12. **listTrace**

```
float [] metaheuristics.generators.HillClimbingRestart.listTrace = new float[1200000] [private]
```

6.41.3.13. **stateReferenceHC**

```
State metaheuristics.generators.HillClimbingRestart.stateReferenceHC [protected]
```

6.41.3.14. **strategy**

```
StrategyType metaheuristics.generators.HillClimbingRestart.strategy [protected]
```

6.41.3.15. **typeAcceptation**

```
AcceptType metaheuristics.generators.HillClimbingRestart.typeAcceptation [protected]
```

6.41.3.16. **typeCandidate**

```
CandidateType metaheuristics.generators.HillClimbingRestart.typeCandidate [protected]
```

6.41.3.17. **weight**

```
float metaheuristics.generators.HillClimbingRestart.weight [protected]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/HillClimbingRestart.java](#)

6.42. Referencia de la interface factory_interface.IFFactoryAcceptCandidate

Diagrama de herencia de factory_interface.IFFactoryAcceptCandidate

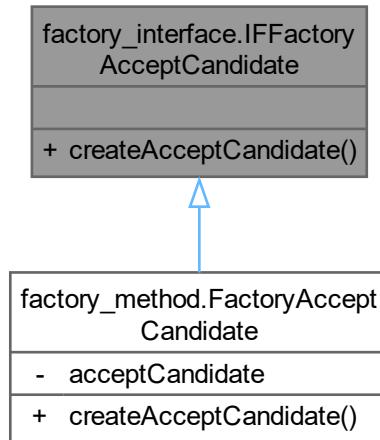
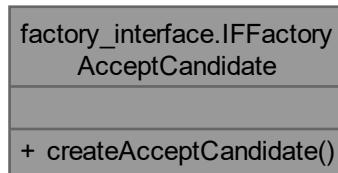


Diagrama de colaboración de factory_interface.IFFactoryAcceptCandidate:



Métodos públicos

- `AcceptableCandidate createAcceptCandidate (AcceptType typeacceptation) throws IllegalArgument←Exception, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.42.1. Documentación de funciones miembro

6.42.1.1. createAcceptCandidate()

```
AcceptableCandidate factory_interface.IFFactoryAcceptCandidate.createAcceptCandidate (
    AcceptType typeacceptation) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementado en [factory_method.FactoryAcceptCandidate](#).

La documentación de esta interfaz está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFactoryAcceptCandidate.java](#)

6.43. Referencia de la interface factory_interface.IFFactoryCandidate

Diagrama de herencia de `factory_interface.IFFactoryCandidate`

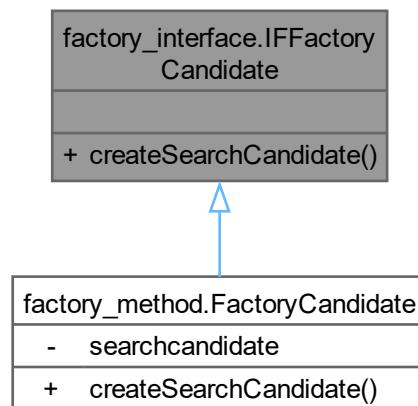
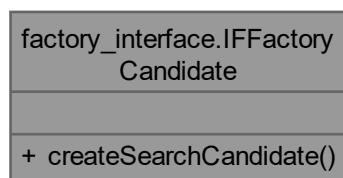


Diagrama de colaboración de `factory_interface.IFFactoryCandidate`:



Métodos públicos

- `SearchCandidate createSearchCandidate (CandidateType typeCandidate) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.43.1. Documentación de funciones miembro

6.43.1.1. `createSearchCandidate()`

```
SearchCandidate factory_interface.IFFactoryCandidate.createSearchCandidate (
    CandidateType typeCandidate) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementado en [factory_method.FactoryCandidate](#).

La documentación de esta interfaz está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFactoryCandidate.java](#)

6.44. Referencia de la interface factory_interface.IFFactoryCrossover

Diagrama de herencia de `factory_interface.IFFactoryCrossover`

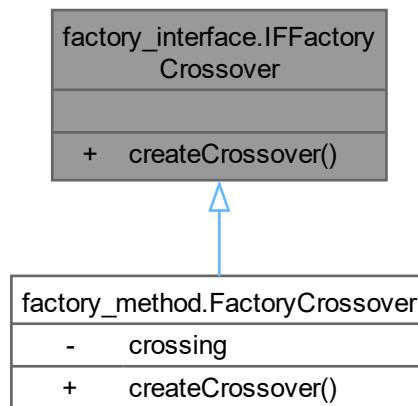
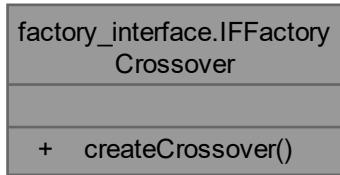


Diagrama de colaboración de factory_interface.IFFactoryCrossover:



Métodos públicos

- `Crossover createCrossover (CrossoverType CrossoverType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.44.1. Documentación de funciones miembro

6.44.1.1. createCrossover()

```
Crossover factory_interface.IFFactoryCrossover.createCrossover (
    CrossoverType CrossoverType) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementado en [factory_method.FactoryCrossover](#).

La documentación de esta interfaz está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFactoryCrossover.java](#)

6.45. Referencia de la interface factory_interface.IFFactoryDistribution

Diagrama de herencia de factory_interface.IFFactoryDistribution

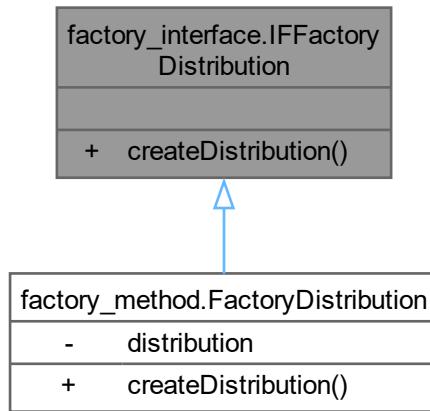
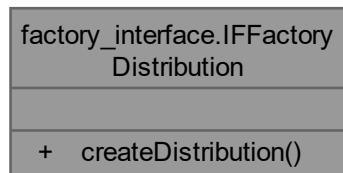


Diagrama de colaboración de factory_interface.IFFactoryDistribution:



Métodos públicos

- `Distribution createDistribution (DistributionType typedistribution) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.45.1. Documentación de funciones miembro

6.45.1.1. createDistribution()

```

Distribution factory_interface.IFFactoryDistribution.createDistribution (
    DistributionType typedistribution) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
  
```

Implementado en [factory_method.FactoryDistribution](#).

La documentación de esta interface está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFFactoryFatherSelection.java](#)

6.46. Referencia de la interface **factory_interface.IFFFactoryFatherSelection**

Diagrama de herencia de factory_interface.IFFFactoryFatherSelection

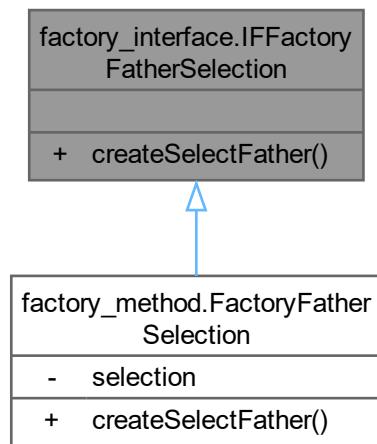
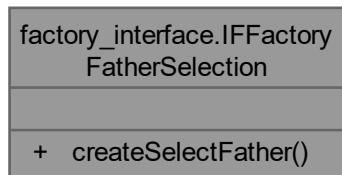


Diagrama de colaboración de factory_interface.IFFFactoryFatherSelection:



Métodos públicos

- `FatherSelection createSelectFather (SelectionType selectionType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.46.1. Documentación de funciones miembro

6.46.1.1. createSelectFather()

```
FatherSelection factory_interface.IFFactoryFatherSelection.createSelectFather (
    SelectionType selectionType) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
Exception, NoSuchMethodException
```

Implementado en [factory_method.FactoryFatherSelection](#).

La documentación de esta interfaz está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFactoryFatherSelection.java](#)

6.47. Referencia de la interface factory_interface.IFFactoryGenerator

Diagrama de herencia de factory_interface.IFFactoryGenerator

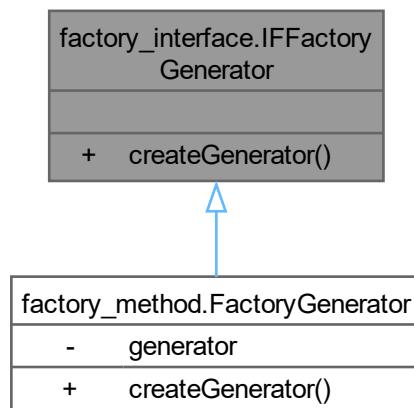
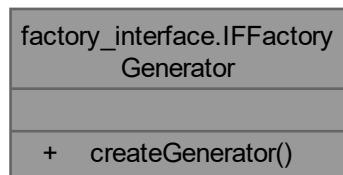


Diagrama de colaboración de factory_interface.IFFactoryGenerator:



Métodos públicos

- `Generator createGenerator (GeneratorType Generatortype) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.47.1. Documentación de funciones miembro

6.47.1.1. `createGenerator()`

```
Generator factory_interface.IFFactoryGenerator.createGenerator (
    GeneratorType Generatortype) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementado en [factory_method.FactoryGenerator](#).

La documentación de esta interfaz está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFactoryGenerator.java](#)

6.48. Referencia de la interface `factory_interface.IFFactoryMutation`

Diagrama de herencia de `factory_interface.IFFactoryMutation`

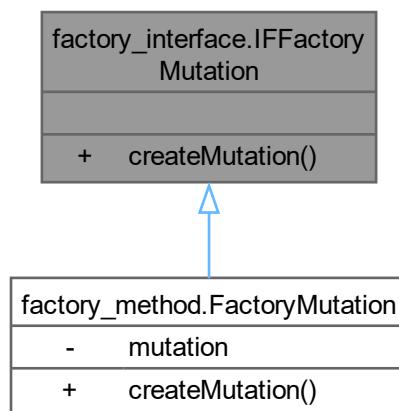
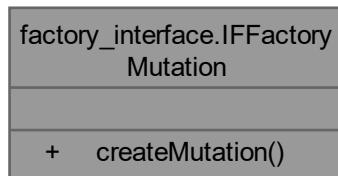


Diagrama de colaboración de factory_interface.IFFactoryMutation:



Métodos públicos

- **Mutation createMutation (MutationType typeMutation)** throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.48.1. Documentación de funciones miembro

6.48.1.1. createMutation()

```
Mutation factory_interface.IFFactoryMutation.createMutation (
    MutationType typeMutation) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementado en [factory_method.FactoryMutation](#).

La documentación de esta interfaz está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFactoryMutation.java](#)

6.49. Referencia de la interface factory_interface.IFFactoryReplace

Diagrama de herencia de factory_interface.IFFactoryReplace

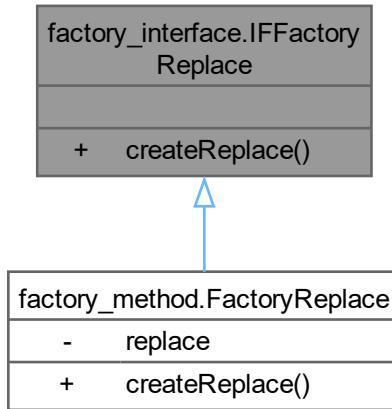
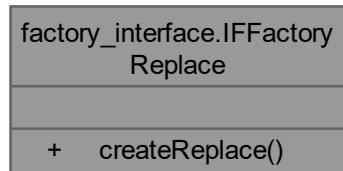


Diagrama de colaboración de factory_interface.IFFactoryReplace:



Métodos públicos

- `Replace createReplace (ReplaceType typereplace) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.49.1. Documentación de funciones miembro

6.49.1.1. createReplace()

```

Replace factory_interface.IFFactoryReplace.createReplace (
    ReplaceType typereplace) throws IllegalArgumentException, SecurityException,
  
```

```
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException<br>Exception, NoSuchMethodException
```

Implementado en [factory_method.FactoryReplace](#).

La documentación de esta interfaz está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFactoryReplace.java](#)

6.50. Referencia de la interface factory_interface.IFFactorySolutionMethod

Diagrama de herencia de factory_interface.IFFactorySolutionMethod

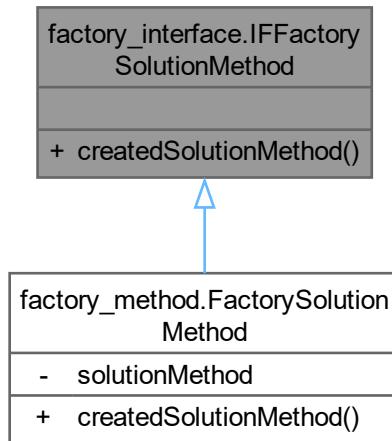
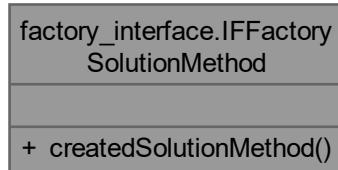


Diagrama de colaboración de factory_interface.IFFactorySolutionMethod:



Métodos públicos

- `SolutionMethod createdSolutionMethod (TypeSolutionMethod method) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.50.1. Documentación de funciones miembro

6.50.1.1. `createdSolutionMethod()`

```
SolutionMethod factory_interface.IFFactorySolutionMethod.createdSolutionMethod (
    TypeSolutionMethod method) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementado en [factory_method.FactorySolutionMethod](#).

La documentación de esta interfaz está generada del siguiente archivo:

- `src/main/java/factory_interface/IFFactorySolutionMethod.java`

6.51. Referencia de la interface `factory_interface.IFFSampling`

Diagrama de herencia de `factory_interface.IFFSampling`

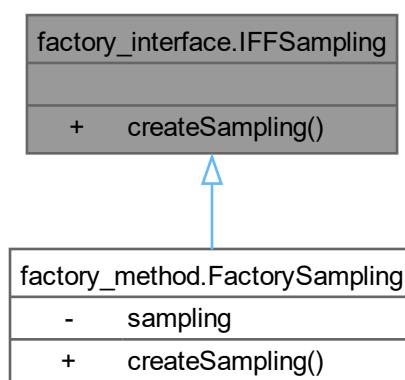
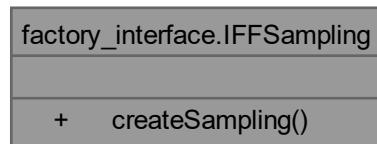


Diagrama de colaboración de factory_interface.IFFSampling:



Métodos públicos

- `Sampling createSampling (SamplingType typesampling) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.51.1. Documentación de funciones miembro

6.51.1.1. createSampling()

```
Sampling factory_interface.IFFSampling.createSampling (
    SamplingType typesampling) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Implementado en [factory_method.FactorySampling](#).

La documentación de esta interface está generada del siguiente archivo:

- [src/main/java/factory_interface/IFFSampling.java](#)

6.52. Referencia de la clase metaheuristics.generators.InstanceDE

Diagrama de herencia de metaheuristics.generators.InstanceDE

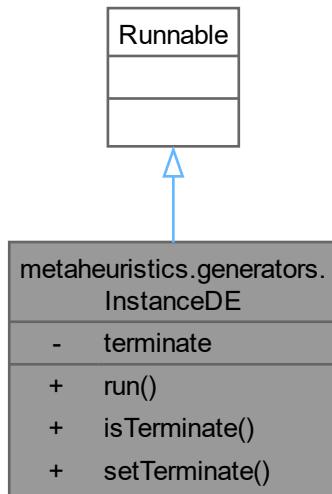
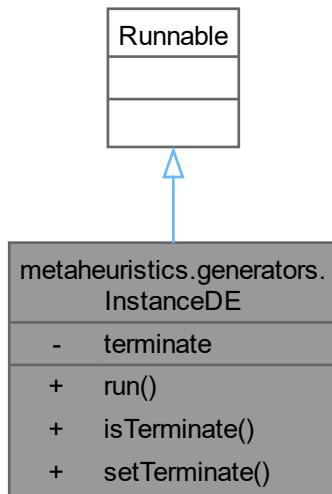


Diagrama de colaboración de metaheuristics.generators.InstanceDE:



Métodos públicos

- `void run ()`

- boolean [isTerminate \(\)](#)
- void [setTerminate \(boolean terminate\)](#)

Atributos privados

- boolean [terminate = false](#)

6.52.1. Documentación de funciones miembro

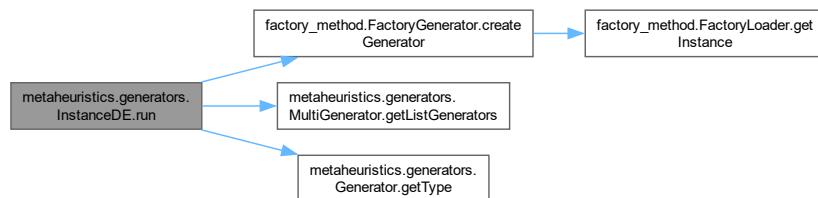
6.52.1.1. [isTerminate\(\)](#)

```
boolean metaheuristics.generators.InstanceDE.isTerminate ()
```

6.52.1.2. [run\(\)](#)

```
void metaheuristics.generators.InstanceDE.run ()
```

Gráfico de llamadas de esta función:



6.52.1.3. [setTerminate\(\)](#)

```
void metaheuristics.generators.InstanceDE.setTerminate (
    boolean terminate)
```

6.52.2. Documentación de datos miembro

6.52.2.1. [terminate](#)

```
boolean metaheuristics.generators.InstanceDE.terminate = false [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/InstanceDE.java](#)

6.53. Referencia de la clase metaheuristics.generators.InstanceEE

Diagrama de herencia de metaheuristics.generators.InstanceEE

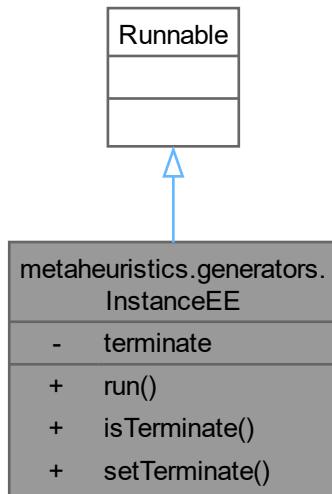
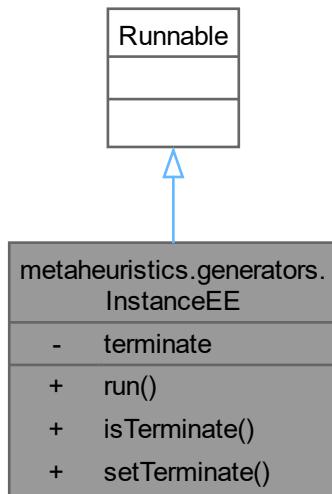


Diagrama de colaboración de metaheuristics.generators.InstanceEE:



Métodos públicos

- `void run ()`

- boolean [isTerminate \(\)](#)
- void [setTerminate \(boolean terminate\)](#)

Atributos privados

- boolean [terminate = false](#)

6.53.1. Documentación de funciones miembro

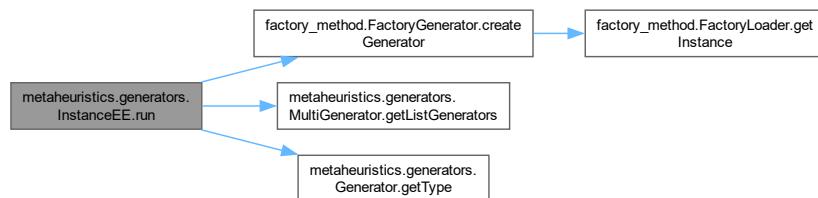
6.53.1.1. [isTerminate\(\)](#)

```
boolean metaheuristics.generators.InstanceEE.isTerminate ()
```

6.53.1.2. [run\(\)](#)

```
void metaheuristics.generators.InstanceEE.run ()
```

Gráfico de llamadas de esta función:



6.53.1.3. [setTerminate\(\)](#)

```
void metaheuristics.generators.InstanceEE.setTerminate (
    boolean terminate)
```

6.53.2. Documentación de datos miembro

6.53.2.1. [terminate](#)

```
boolean metaheuristics.generators.InstanceEE.terminate = false [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/InstanceEE.java](#)

6.54. Referencia de la clase metaheuristics.generators.InstanceGA

Diagrama de herencia de metaheuristics.generators.InstanceGA

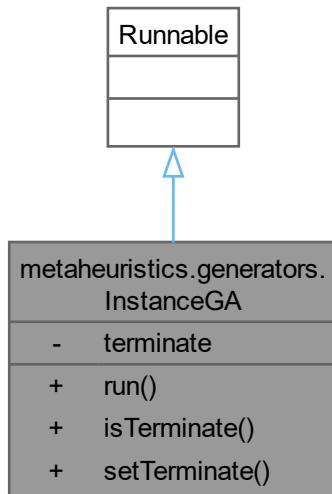
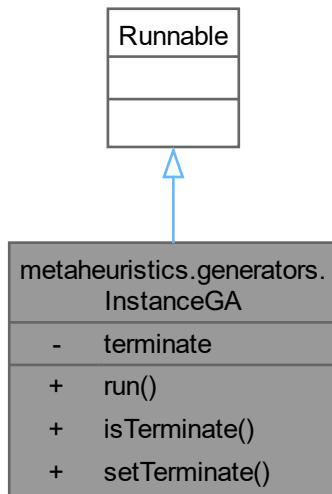


Diagrama de colaboración de metaheuristics.generators.InstanceGA:



Métodos públicos

- void `run ()`

- boolean [isTerminate \(\)](#)
- void [setTerminate \(boolean terminate\)](#)

Atributos privados

- boolean [terminate = false](#)

6.54.1. Documentación de funciones miembro

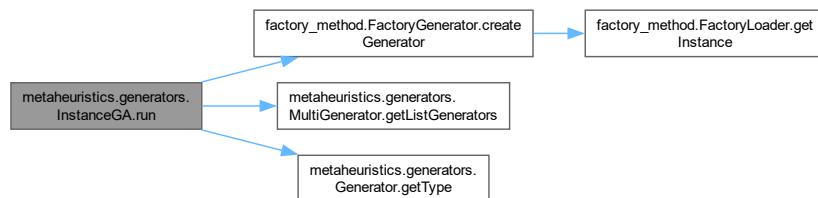
6.54.1.1. [isTerminate\(\)](#)

```
boolean metaheuristics.generators.InstanceGA.isTerminate ()
```

6.54.1.2. [run\(\)](#)

```
void metaheuristics.generators.InstanceGA.run ()
```

Gráfico de llamadas de esta función:



6.54.1.3. [setTerminate\(\)](#)

```
void metaheuristics.generators.InstanceGA.setTerminate (
    boolean terminate)
```

6.54.2. Documentación de datos miembro

6.54.2.1. [terminate](#)

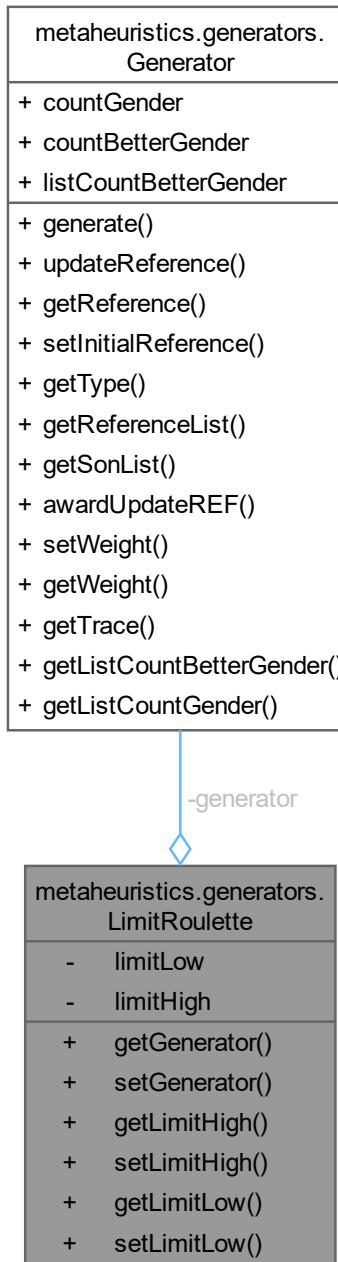
```
boolean metaheuristics.generators.InstanceGA.terminate = false [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/InstanceGA.java](#)

6.55. Referencia de la clase metaheuristics.generators.LimitRoulette

Diagrama de colaboración de metaheuristics.generators.LimitRoulette:



Métodos públicos

- [Generator getGenerator \(\)](#)
- [void setGenerator \(Generator generator\)](#)

- float `getLimitHigh ()`
- void `setLimitHigh (float limitHigh)`
- float `getLimitLow ()`
- void `setLimitLow (float limitLow)`

Atributos privados

- float `limitLow`
- float `limitHigh`
- Generator `generator`

6.55.1. Documentación de funciones miembro

6.55.1.1. `getGenerator()`

```
Generator metaheuristics.generators.LimitRoulette.getGenerator ()
```

6.55.1.2. `getLimitHigh()`

```
float metaheuristics.generators.LimitRoulette.getLimitHigh ()
```

6.55.1.3. `getLimitLow()`

```
float metaheuristics.generators.LimitRoulette.getLimitLow ()
```

6.55.1.4. `setGenerator()`

```
void metaheuristics.generators.LimitRoulette.setGenerator (
    Generator generator)
```

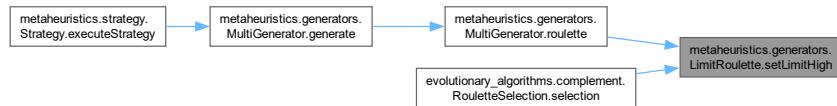
Gráfico de llamadas a esta función:



6.55.1.5. setLimitHigh()

```
void metaheuristics.generators.LimitRoulette.setLimitHigh (
    float limitHigh)
```

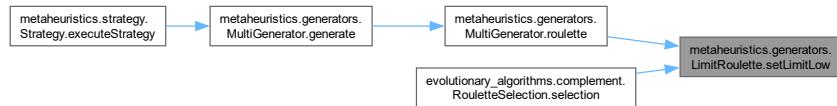
Gráfico de llamadas a esta función:



6.55.1.6. setLimitLow()

```
void metaheuristics.generators.LimitRoulette.setLimitLow (
    float limitLow)
```

Gráfico de llamadas a esta función:



6.55.2. Documentación de datos miembro

6.55.2.1. generator

```
Generator metaheuristics.generators.LimitRoulette.generator [private]
```

6.55.2.2. limitHigh

```
float metaheuristics.generators.LimitRoulette.limitHigh [private]
```

6.55.2.3. limitLow

```
float metaheuristics.generators.LimitRoulette.limitLow [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/LimitRoulette.java](#)

6.56. Referencia de la clase metaheuristics.generators.LimitThreshold

Diagrama de herencia de metaheuristics.generators.LimitThreshold

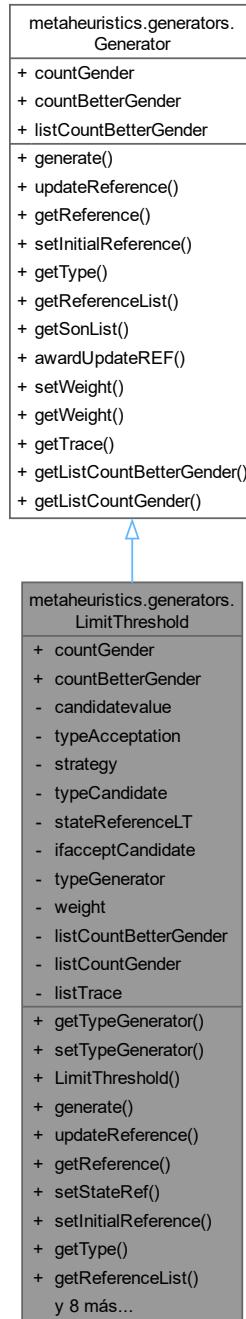
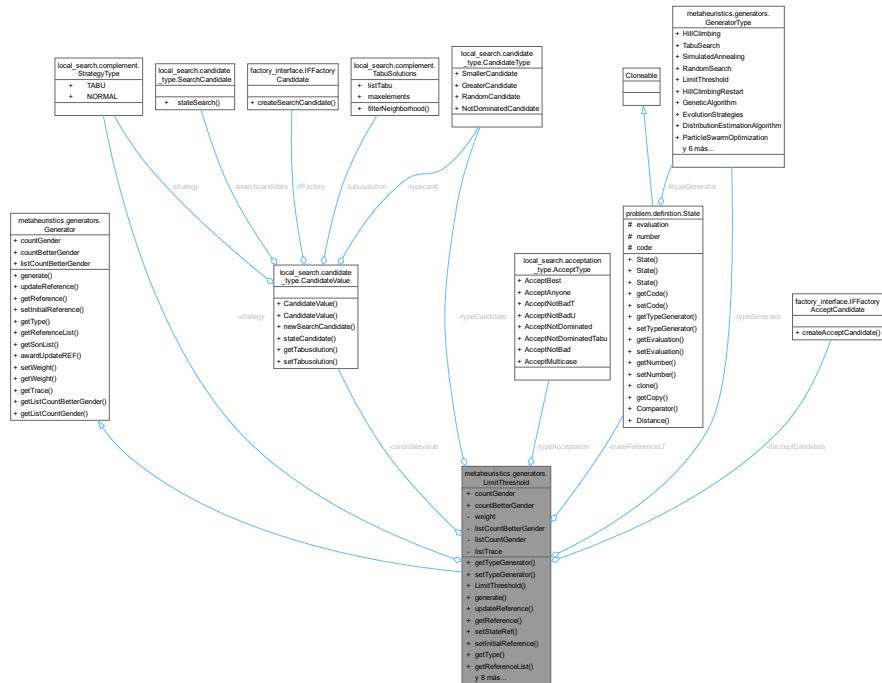


Diagrama de colaboración de `metaheuristics.generators.LimitThreshold`:



Métodos públicos

- `GeneratorType getTypeGenerator ()`
- `void setTypeGenerator (GeneratorType typeGenerator)`
- `LimitThreshold ()`
- `State generate (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `State getReference ()`
- `void setStateRef (State stateRef)`
- `void setInitialReference (State statelInitialRef)`
- `GeneratorType getType ()`
- `List< State > getReferenceList ()`
- `List< State > getSonList ()`
- `void setTypeCandidate (CandidateType typeCandidate)`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `int[] getCountBetterGender ()`
- `int[] getListCountGender ()`
- `float[] getTrace ()`

Atributos públicos estáticos

- `static int countGender = 0`
- `static int countBetterGender = 0`

Atributos privados

- CandidateValue candidatevalue
- AcceptType typeAcceptation
- StrategyType strategy
- CandidateType typeCandidate
- State stateReferenceLT
- IFFactoryAcceptCandidate ifacceptCandidate
- GeneratorType typeGenerator
- float weight
- int[] listCountBetterGender = new int[10]
- int[] listCountGender = new int[10]
- float[] listTrace = new float[1200000]

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

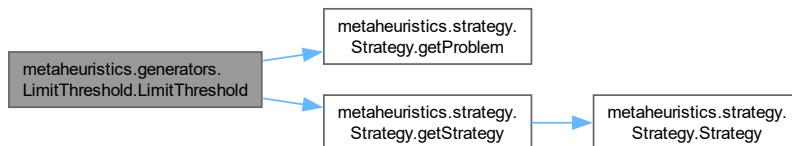
- int countGender
- int countBetterGender
- int[] listCountBetterGender

6.56.1. Documentación de constructores y destructores

6.56.1.1. LimitThreshold()

```
metaheuristics.generators.LimitThreshold.LimitThreshold ()
```

Gráfico de llamadas de esta función:



6.56.2. Documentación de funciones miembro

6.56.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.LimitThreshold.awardUpdateREF (
    State stateCandidate)
```

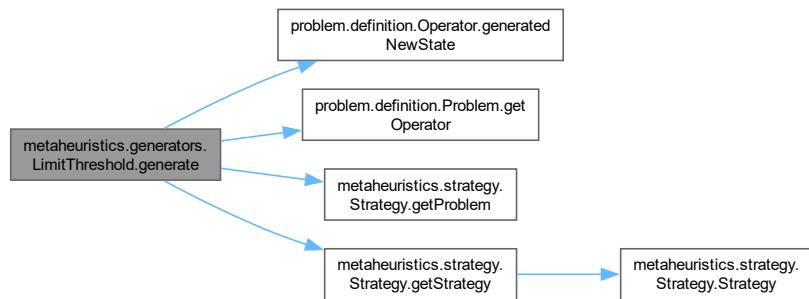
Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.2. generate()

```
State metaheuristics.generators.LimitThreshold.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.56.2.3. getListCountBetterGender()

```
int[] metaheuristics.generators.LimitThreshold.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.4. getListCountGender()

```
int[] metaheuristics.generators.LimitThreshold.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.5. getReference()

```
State metaheuristics.generators.LimitThreshold.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.6. getReferenceList()

```
List< State > metaheuristics.generators.LimitThreshold.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.7. getSonList()

```
List< State > metaheuristics.generators.LimitThreshold.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.8. getTrace()

```
float[ ] metaheuristics.generators.LimitThreshold.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.9. getType()

```
GeneratorType metaheuristics.generators.LimitThreshold.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.10. getTypeGenerator()

```
GeneratorType metaheuristics.generators.LimitThreshold.getTypeGenerator ()
```

6.56.2.11. getWeight()

```
float metaheuristics.generators.LimitThreshold.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.12. setInitialReference()

```
void metaheuristics.generators.LimitThreshold.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.13. setStateRef()

```
void metaheuristics.generators.LimitThreshold.setStateRef (
    State stateRef)
```

6.56.2.14. setTypeCandidate()

```
void metaheuristics.generators.LimitThreshold.setTypeCandidate (
    CandidateType typeCandidate)
```

6.56.2.15. `setTypeGenerator()`

```
void metaheuristics.generators.LimitThreshold.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.56.2.16. `setWeight()`

```
void metaheuristics.generators.LimitThreshold.setWeight (
    float weight)
```

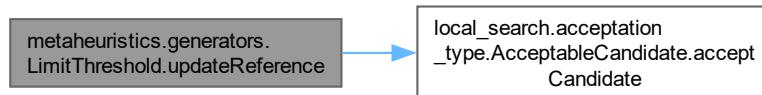
Reimplementado de [metaheuristics.generators.Generator](#).

6.56.2.17. `updateReference()`

```
void metaheuristics.generators.LimitThreshold.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.56.3. Documentación de datos miembro

6.56.3.1. `candidatevalue`

```
CandidateValue metaheuristics.generators.LimitThreshold.candidatevalue [private]
```

6.56.3.2. `countBetterGender`

```
int metaheuristics.generators.LimitThreshold.countBetterGender = 0 [static]
```

6.56.3.3. `countGender`

```
int metaheuristics.generators.LimitThreshold.countGender = 0 [static]
```

6.56.3.4. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.LimitThreshold.ifacceptCandidate [private]
```

6.56.3.5. listCountBetterGender

```
int [] metaheuristics.generators.LimitThreshold.listCountBetterGender = new int[10] [private]
```

6.56.3.6. listCountGender

```
int [] metaheuristics.generators.LimitThreshold.listCountGender = new int[10] [private]
```

6.56.3.7. listTrace

```
float [] metaheuristics.generators.LimitThreshold.listTrace = new float[1200000] [private]
```

6.56.3.8. stateReferenceLT

```
State metaheuristics.generators.LimitThreshold.stateReferenceLT [private]
```

6.56.3.9. strategy

```
StrategyType metaheuristics.generators.LimitThreshold.strategy [private]
```

6.56.3.10. typeAcceptation

```
AcceptType metaheuristics.generators.LimitThreshold.typeAcceptation [private]
```

6.56.3.11. typeCandidate

```
CandidateType metaheuristics.generators.LimitThreshold.typeCandidate [private]
```

6.56.3.12. typeGenerator

```
GeneratorType metaheuristics.generators.LimitThreshold.typeGenerator [private]
```

6.56.3.13. weight

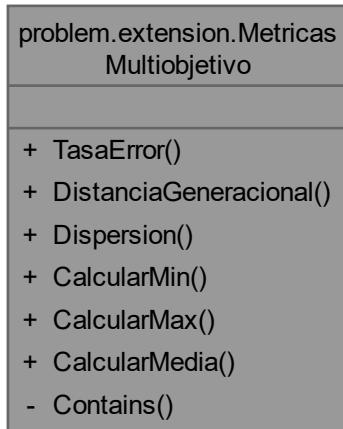
```
float metaheuristics.generators.LimitThreshold.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/LimitThreshold.java](#)

6.57. Referencia de la clase problem.extension.MetricasMultiobjetivo

Diagrama de colaboración de problem.extension.MetricasMultiobjetivo:



Métodos públicos

- double **TasaError** (List< State > solutionsFPcurrent, List< State > solutionsFPtrue) throws BiffException, IOException
- double **DistanciaGeneracional** (List< State > solutionsFPcurrent, List< State > solutionsFPtrue) throws BiffException, IOException
- double **Dispersion** (ArrayList< State > solutions) throws BiffException, IOException
- double **CalcularMin** (ArrayList< Double > allMetrics)
- double **CalcularMax** (ArrayList< Double > allMetrics)
- double **CalcularMedia** (ArrayList< Double > allMetrics)

Métodos privados

- boolean **Contains** (State solA, List< State > solutions)

6.57.1. Documentación de funciones miembro

6.57.1.1. CalcularMax()

```
double problem.extension.MetricasMultiobjetivo.CalcularMax (
    ArrayList< Double > allMetrics)
```

6.57.1.2. CalcularMedia()

```
double problem.extension.MetricasMultiobjetivo.CalcularMedia (
    ArrayList< Double > allMetrics)
```

6.57.1.3. CalcularMin()

```
double problem.extension.MetricasMultiobjetivo.CalcularMin (ArrayList< Double > allMetrics)
```

6.57.1.4. Contains()

```
boolean problem.extension.MetricasMultiobjetivo.Contains (State sola, List< State > solutions) [private]
```

Gráfico de llamadas de esta función:

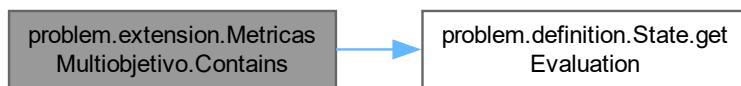
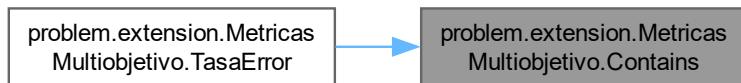


Gráfico de llamadas a esta función:



6.57.1.5. Dispersion()

```
double problem.extension.MetricasMultiobjetivo.Dispersion (ArrayList< State > solutions) throws BiffException, IOException
```

Gráfico de llamadas de esta función:



6.57.1.6. DistanciaGeneracional()

```
double problem.extension.MetricasMultiobjetivo.DistanciaGeneracional (
    List< State > solutionsFPcurrent,
    List< State > solutionsFPtrue) throws BiffException, IOException
```

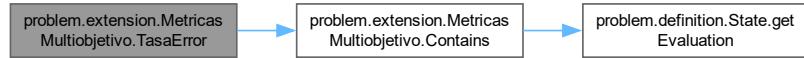
Gráfico de llamadas de esta función:



6.57.1.7. TasaError()

```
double problem.extension.MetricasMultiobjetivo.TasaError (
    List< State > solutionsFPcurrent,
    List< State > solutionsFPtrue) throws BiffException, IOException
```

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/extension/MetricasMultiobjetivo.java](#)

6.58. Referencia de la clase local_search.complement.MultiCaseSimulatedAnnealing

Diagrama de herencia de local_search.complement.MultiCaseSimulatedAnnealing

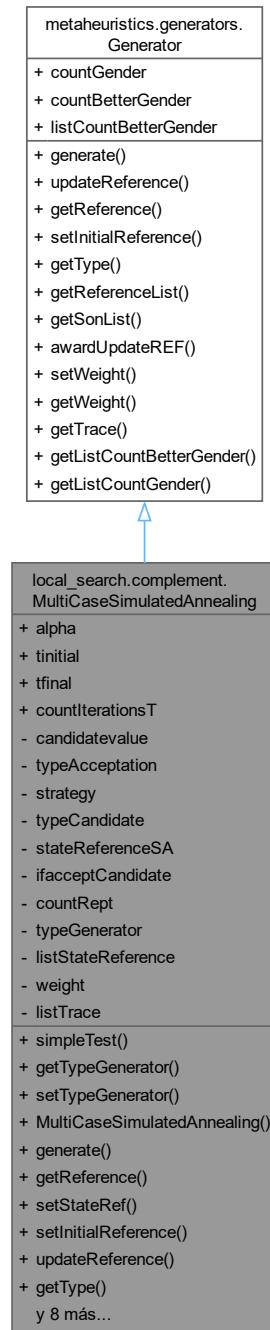
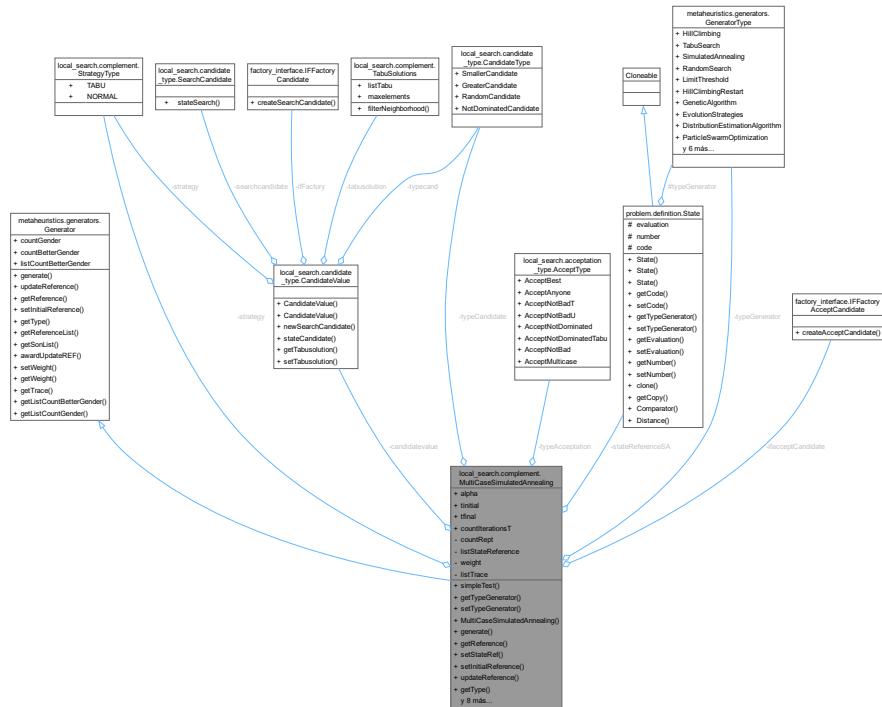


Diagrama de colaboración de local_search.complement.MultiCaseSimulatedAnnealing:



Métodos públicos

- int **simpleTest ()**
- GeneratorType **getTypeGenerator ()**
- void **setTypeGenerator (GeneratorType typeGenerator)**
- MultiCaseSimulatedAnnealing ()
- State **generate (Integer operatornumber)** throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- State **getReference ()**
- void **setStateRef (State stateRef)**
- void **setInitialState (State statelInitialRef)**
- void **updateReference (State stateCandidate, Integer countIterationsCurrent)** throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- GeneratorType **getType ()**
- List< State > **getReferenceList ()**
- List< State > **getSonList ()**
- boolean **awardUpdateREF (State stateCandidate)**
- float **getWeight ()**
- void **setWeight (float weight)**
- int[] **getListCountBetterGender ()**
- int[] **getListCountGender ()**
- float[] **getTrace ()**

Atributos públicos estáticos

- static Double [alpha](#)
- static Double [tinitial](#)
- static Double [tfinal](#)
- static int [countIterationsT](#)

Atributos privados

- [CandidateValue candidatevalue](#)
- [AcceptType typeAcceptation](#)
- [StrategyType strategy](#)
- [CandidateType typeCandidate](#)
- [State stateReferenceSA](#)
- [IFFactoryAcceptCandidate ifacceptCandidate](#)
- int [countRept](#)
- [GeneratorType typeGenerator](#)
- List< [State](#) > [listStateReference](#) = new ArrayList<[State](#)>()
- float [weight](#)
- List< [Float](#) > [listTrace](#) = new ArrayList<[Float](#)>()

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.58.1. Documentación de constructores y destructores

6.58.1.1. MultiCaseSimulatedAnnealing()

```
local_search.complement.MultiCaseSimulatedAnnealing.MultiCaseSimulatedAnnealing ()
```

6.58.2. Documentación de funciones miembro

6.58.2.1. awardUpdateREF()

```
boolean local_search.complement.MultiCaseSimulatedAnnealing.awardUpdateREF (   
    State stateCandidate)
```

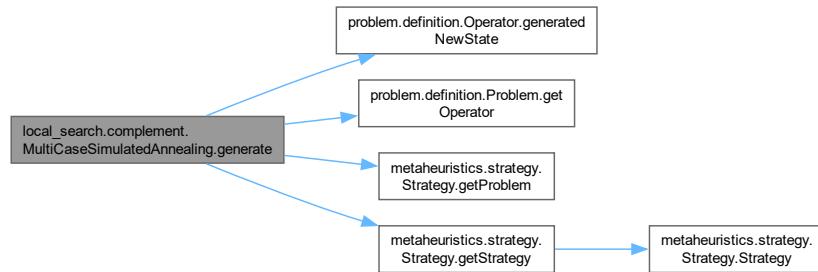
Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.2. generate()

```
State local_search.complement.MultiCaseSimulatedAnnealing.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.58.2.3. getListCountBetterGender()

```
int[] local_search.complement.MultiCaseSimulatedAnnealing.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.4. getListCountGender()

```
int[] local_search.complement.MultiCaseSimulatedAnnealing.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.5. getReference()

```
State local_search.complement.MultiCaseSimulatedAnnealing.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.6. `getReferenceList()`

```
List< State > local_search.complement.MultiCaseSimulatedAnnealing.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.58.2.7. `getSonList()`

```
List< State > local_search.complement.MultiCaseSimulatedAnnealing.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.8. `getTrace()`

```
float[ ] local_search.complement.MultiCaseSimulatedAnnealing.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.9. `getType()`

```
GeneratorType local_search.complement.MultiCaseSimulatedAnnealing.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.10. `getTypeGenerator()`

```
GeneratorType local_search.complement.MultiCaseSimulatedAnnealing.getTypeGenerator ()
```

6.58.2.11. `getWeight()`

```
float local_search.complement.MultiCaseSimulatedAnnealing.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.12. `setInitialReference()`

```
void local_search.complement.MultiCaseSimulatedAnnealing.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.13. `setStateRef()`

```
void local_search.complement.MultiCaseSimulatedAnnealing.setStateRef (
    State stateRef)
```

6.58.2.14. `setTypeGenerator()`

```
void local_search.complement.MultiCaseSimulatedAnnealing.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.58.2.15. `setWeight()`

```
void local_search.complement.MultiCaseSimulatedAnnealing.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.58.2.16. `simpleTest()`

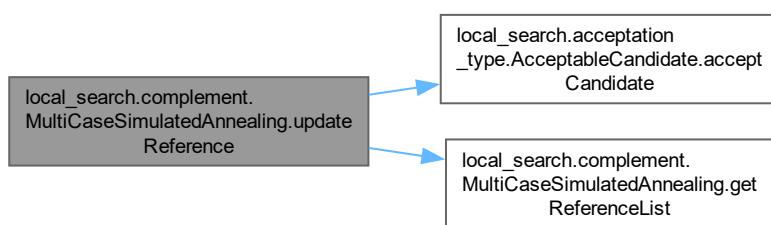
```
int local_search.complement.MultiCaseSimulatedAnnealing.simpleTest ()
```

6.58.2.17. `updateReference()`

```
void local_search.complement.MultiCaseSimulatedAnnealing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.58.3. Documentación de datos miembro

6.58.3.1. alpha

```
Double local_search.complement.MultiCaseSimulatedAnnealing.alpha [static]
```

6.58.3.2. candidatevalue

```
CandidateValue local_search.complement.MultiCaseSimulatedAnnealing.candidatevalue [private]
```

6.58.3.3. countIterationsT

```
int local_search.complement.MultiCaseSimulatedAnnealing.countIterationsT [static]
```

6.58.3.4. countRept

```
int local_search.complement.MultiCaseSimulatedAnnealing.countRept [private]
```

6.58.3.5. ifacceptCandidate

```
IFFactoryAcceptCandidate local_search.complement.MultiCaseSimulatedAnnealing.ifacceptCandidate  
[private]
```

6.58.3.6. listStateReference

```
List<State> local_search.complement.MultiCaseSimulatedAnnealing.listStateReference = new  
ArrayList<State>() [private]
```

6.58.3.7. listTrace

```
List<Float> local_search.complement.MultiCaseSimulatedAnnealing.listTrace = new ArrayList<Float>()  
[private]
```

6.58.3.8. stateReferenceSA

```
State local_search.complement.MultiCaseSimulatedAnnealing.stateReferenceSA [private]
```

6.58.3.9. strategy

```
StrategyType local_search.complement.MultiCaseSimulatedAnnealing.strategy [private]
```

6.58.3.10. tfinal

```
Double local_search.complement.MultiCaseSimulatedAnnealing.tfinal [static]
```

6.58.3.11. tinitial

```
Double local_search.complement.MultiCaseSimulatedAnnealing.tinitial [static]
```

6.58.3.12. typeAcceptation

```
AcceptType local_search.complement.MultiCaseSimulatedAnnealing.typeAcceptation [private]
```

6.58.3.13. typeCandidate

```
CandidateType local_search.complement.MultiCaseSimulatedAnnealing.typeCandidate [private]
```

6.58.3.14. typeGenerator

```
GeneratorType local_search.complement.MultiCaseSimulatedAnnealing.typeGenerator [private]
```

6.58.3.15. weight

```
float local_search.complement.MultiCaseSimulatedAnnealing.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/complement/MultiCaseSimulatedAnnealing.java](#)

6.59. Referencia de la clase local_search.MultiCaseSimulatedAnnealing

Diagrama de herencia de local_search.MultiCaseSimulatedAnnealing

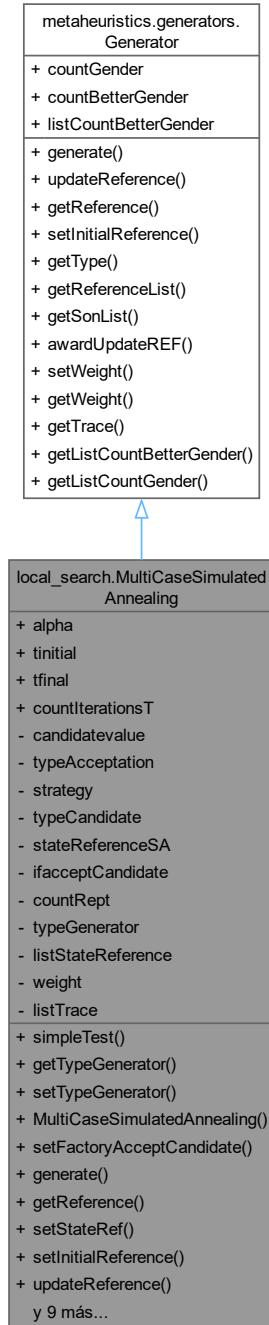
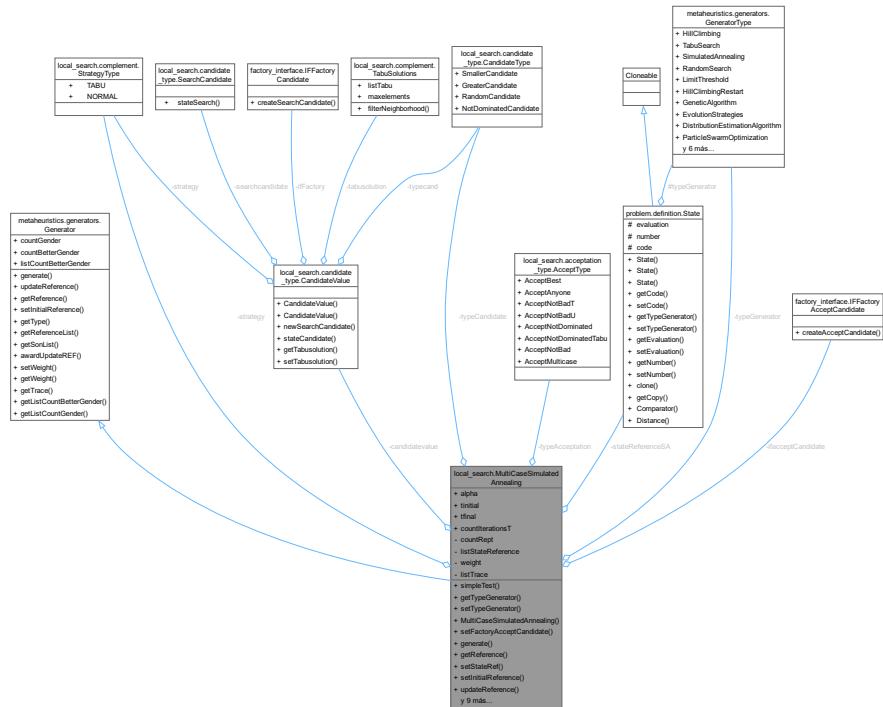


Diagrama de colaboración de local_search.MultiCaseSimulatedAnnealing:



Métodos públicos

- int **simpleTest ()**
- GeneratorType **getTypeGenerator ()**
- void **setTypeGenerator (GeneratorType typeGenerator)**
- MultiCaseSimulatedAnnealing ()
- void **setFactoryAcceptCandidate (IFFactoryAcceptCandidate ifacceptCandidate)**
- State **generate (Integer operatormumber)** throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- State **getReference ()**
- void **setStateRef (State stateRef)**
- void **setInitialReference (State statelInitialRef)**
- void **updateReference (State stateCandidate, Integer countIterationsCurrent)** throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- GeneratorType **getType ()**
- List< State > **getReferenceList ()**
- List< State > **getSonList ()**
- boolean **awardUpdateREF (State stateCandidate)**
- float **getWeight ()**
- void **setWeight (float weight)**
- int[] **getListCountBetterGender ()**
- int[] **getListCountGender ()**
- float[] **getTrace ()**

Atributos públicos estáticos

- static Double `alpha`
- static Double `tinitial`
- static Double `tfinal`
- static int `countIterationsT`

Atributos privados

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceSA`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- int `countRept`
- `GeneratorType typeGenerator`
- List< `State` > `listStateReference` = new ArrayList<`State`>()
- float `weight`
- List< `Float` > `listTrace` = new ArrayList<`Float`>()

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.59.1. Documentación de constructores y destructores

6.59.1.1. MultiCaseSimulatedAnnealing()

```
local_search.MultiCaseSimulatedAnnealing.MultiCaseSimulatedAnnealing ()
```

6.59.2. Documentación de funciones miembro

6.59.2.1. awardUpdateREF()

```
boolean local_search.MultiCaseSimulatedAnnealing.awardUpdateREF (
    State stateCandidate)
```

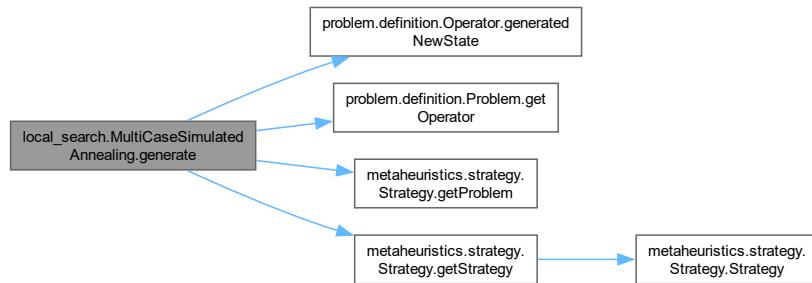
Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.2. generate()

```
State local_search.MultiCaseSimulatedAnnealing.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.59.2.3. getListCountBetterGender()

```
int[] local_search.MultiCaseSimulatedAnnealing.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.4. getListCountGender()

```
int[] local_search.MultiCaseSimulatedAnnealing.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.5. getReference()

```
State local_search.MultiCaseSimulatedAnnealing.getReference ()
```

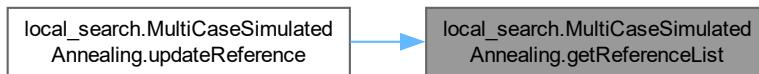
Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.6. `getReferenceList()`

```
List< State > local_search.MultiCaseSimulatedAnnealing.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.59.2.7. `getSonList()`

```
List< State > local_search.MultiCaseSimulatedAnnealing.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.8. `getTrace()`

```
float[ ] local_search.MultiCaseSimulatedAnnealing.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.9. `getType()`

```
GeneratorType local_search.MultiCaseSimulatedAnnealing.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.10. `getTypeGenerator()`

```
GeneratorType local_search.MultiCaseSimulatedAnnealing.getTypeGenerator ()
```

6.59.2.11. `getWeight()`

```
float local_search.MultiCaseSimulatedAnnealing.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.12. `setFactoryAcceptCandidate()`

```
void local_search.MultiCaseSimulatedAnnealing.setFactoryAcceptCandidate (
    IFFactoryAcceptCandidate ifacceptCandidate)
```

6.59.2.13. `setInitialReference()`

```
void local_search.MultiCaseSimulatedAnnealing.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.14. `setStateRef()`

```
void local_search.MultiCaseSimulatedAnnealing.setStateRef (
    State stateRef)
```

6.59.2.15. `setTypeGenerator()`

```
void local_search.MultiCaseSimulatedAnnealing.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.59.2.16. `setWeight()`

```
void local_search.MultiCaseSimulatedAnnealing.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.59.2.17. `simpleTest()`

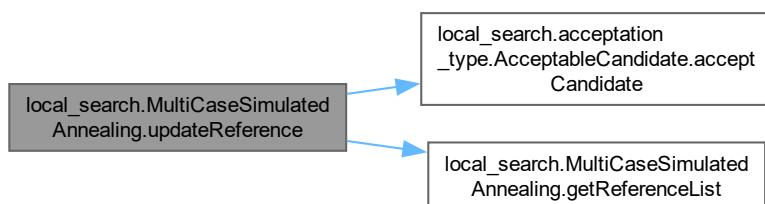
```
int local_search.MultiCaseSimulatedAnnealing.simpleTest ()
```

6.59.2.18. `updateReference()`

```
void local_search.MultiCaseSimulatedAnnealing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.59.3. Documentación de datos miembro

6.59.3.1. alpha

```
Double local_search.MultiCaseSimulatedAnnealing.alpha [static]
```

6.59.3.2. candidatevalue

```
CandidateValue local_search.MultiCaseSimulatedAnnealing.candidatevalue [private]
```

6.59.3.3. countIterationsT

```
int local_search.MultiCaseSimulatedAnnealing.countIterationsT [static]
```

6.59.3.4. countRept

```
int local_search.MultiCaseSimulatedAnnealing.countRept [private]
```

6.59.3.5. ifacceptCandidate

```
IFFactoryAcceptCandidate local_search.MultiCaseSimulatedAnnealing.ifacceptCandidate [private]
```

6.59.3.6. listStateReference

```
List<State> local_search.MultiCaseSimulatedAnnealing.listStateReference = new ArrayList<State>()  
[private]
```

6.59.3.7. listTrace

```
List<Float> local_search.MultiCaseSimulatedAnnealing.listTrace = new ArrayList<Float>() [private]
```

6.59.3.8. stateReferenceSA

```
State local_search.MultiCaseSimulatedAnnealing.stateReferenceSA [private]
```

6.59.3.9. strategy

```
StrategyType local_search.MultiCaseSimulatedAnnealing.strategy [private]
```

6.59.3.10. tfinal

```
Double local_search.MultiCaseSimulatedAnnealing.tfinal [static]
```

6.59.3.11. **tinitial**

```
Double local_search.MultiCaseSimulatedAnnealing.tinitial [static]
```

6.59.3.12. **typeAcceptation**

```
AcceptType local_search.MultiCaseSimulatedAnnealing.typeAcceptation [private]
```

6.59.3.13. **typeCandidate**

```
CandidateType local_search.MultiCaseSimulatedAnnealing.typeCandidate [private]
```

6.59.3.14. **typeGenerator**

```
GeneratorType local_search.MultiCaseSimulatedAnnealing.typeGenerator [private]
```

6.59.3.15. **weight**

```
float local_search.MultiCaseSimulatedAnnealing.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/MultiCaseSimulatedAnnealing.java](#)

6.60. Referencia de la clase metaheuristics.generators.MultiCaseSimulatedAnnealing

Diagrama de herencia de metaheuristics.generators.MultiCaseSimulatedAnnealing

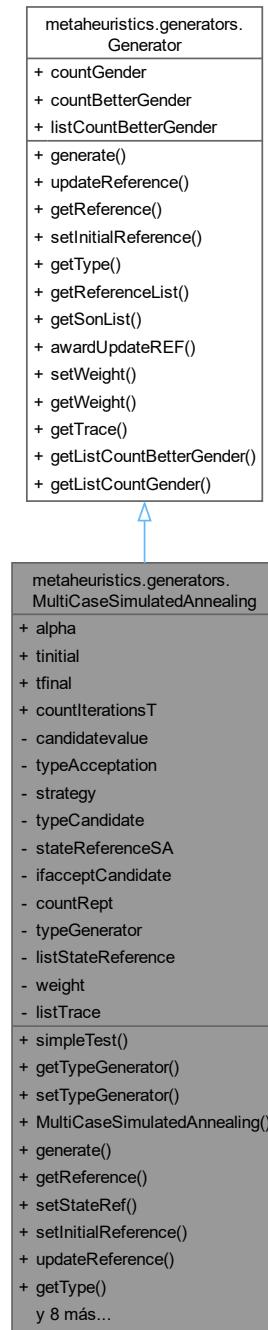
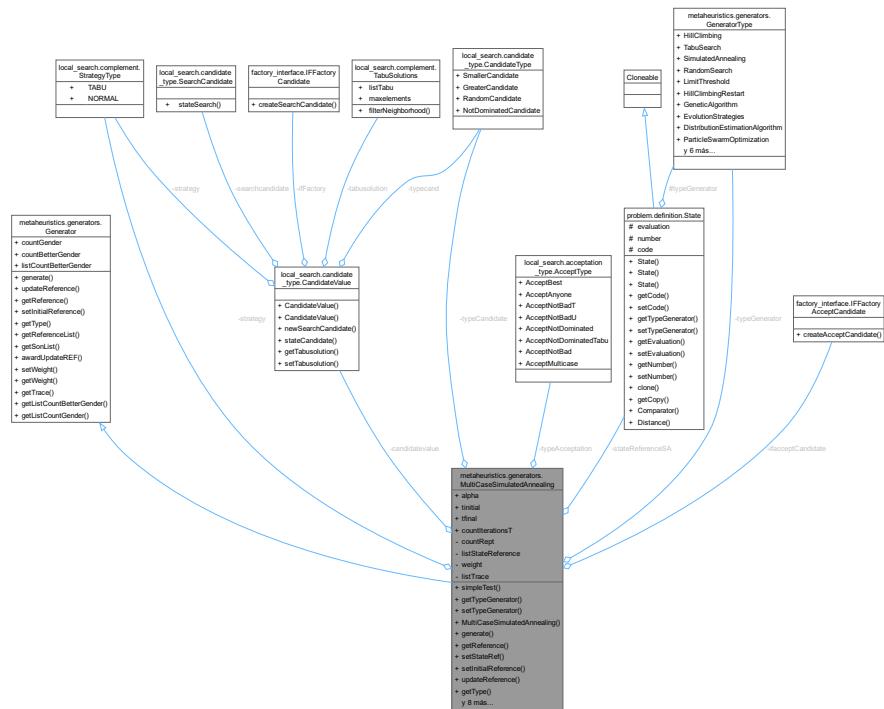


Diagrama de colaboración de `metaheuristics.generators.MultiCaseSimulatedAnnealing`:



Métodos públicos

- `int simpleTest ()`
- `GeneratorType getTypeGenerator ()`
- `void setTypeGenerator (GeneratorType typeGenerator)`
- `MultiCaseSimulatedAnnealing ()`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `State getReference ()`
- `void setStateRef (State stateRef)`
- `void setInitialReference (State statelInitialRef)`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `GeneratorType getType ()`
- `List< State > getReferenceList ()`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `int[] getListCountBetterGender ()`
- `int[] getListCountGender ()`
- `float[] getTrace ()`

Atributos públicos estáticos

- static Double `alpha`
- static Double `tinitial`
- static Double `tfinal`
- static int `countIterationsT`

Atributos privados

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceSA`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- int `countRept`
- `GeneratorType typeGenerator`
- List< `State` > `listStateReference` = new ArrayList<`State`>()
- float `weight`
- List< `Float` > `listTrace` = new ArrayList<`Float`>()

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.60.1. Documentación de constructores y destructores

6.60.1.1. MultiCaseSimulatedAnnealing()

```
metaheuristics.generators.MultiCaseSimulatedAnnealing.MultiCaseSimulatedAnnealing ()
```

6.60.2. Documentación de funciones miembro

6.60.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.MultiCaseSimulatedAnnealing.awardUpdateREF (
    State stateCandidate)
```

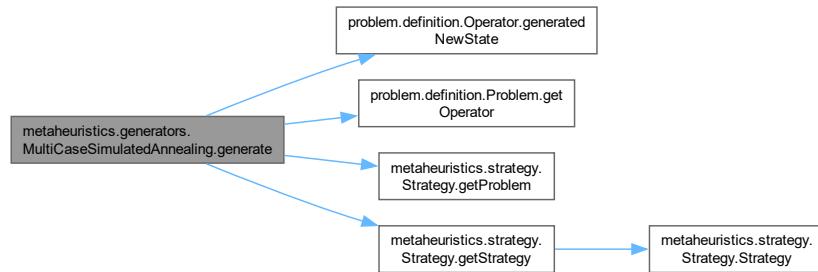
Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.2. generate()

```
State metaheuristics.generators.MultiCaseSimulatedAnnealing.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.60.2.3. getListCountBetterGender()

```
int[] metaheuristics.generators.MultiCaseSimulatedAnnealing.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.4. getListCountGender()

```
int[] metaheuristics.generators.MultiCaseSimulatedAnnealing.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.5. getReference()

```
State metaheuristics.generators.MultiCaseSimulatedAnnealing.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.6. `getReferenceList()`

```
List< State > metaheuristics.generators.MultiCaseSimulatedAnnealing.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.60.2.7. `getSonList()`

```
List< State > metaheuristics.generators.MultiCaseSimulatedAnnealing.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.8. `getTrace()`

```
float[] metaheuristics.generators.MultiCaseSimulatedAnnealing.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.9. `getType()`

```
GeneratorType metaheuristics.generators.MultiCaseSimulatedAnnealing.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.10. `getTypeGenerator()`

```
GeneratorType metaheuristics.generators.MultiCaseSimulatedAnnealing.getTypeGenerator ()
```

6.60.2.11. `getWeight()`

```
float metaheuristics.generators.MultiCaseSimulatedAnnealing.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.12. `setInitialReference()`

```
void metaheuristics.generators.MultiCaseSimulatedAnnealing.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.13. `setStateRef()`

```
void metaheuristics.generators.MultiCaseSimulatedAnnealing.setStateRef (
    State stateRef)
```

6.60.2.14. `setTypeGenerator()`

```
void metaheuristics.generators.MultiCaseSimulatedAnnealing.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.60.2.15. `setWeight()`

```
void metaheuristics.generators.MultiCaseSimulatedAnnealing.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.60.2.16. `simpleTest()`

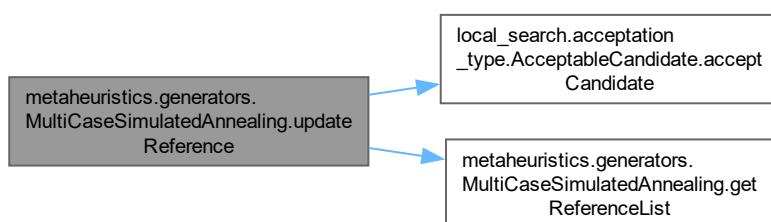
```
int metaheuristics.generators.MultiCaseSimulatedAnnealing.simpleTest ()
```

6.60.2.17. `updateReference()`

```
void metaheuristics.generators.MultiCaseSimulatedAnnealing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.60.3. Documentación de datos miembro

6.60.3.1. alpha

```
Double metaheuristics.generators.MultiCaseSimulatedAnnealing.alpha [static]
```

6.60.3.2. candidatevalue

```
CandidateValue metaheuristics.generators.MultiCaseSimulatedAnnealing.candidatevalue [private]
```

6.60.3.3. countIterationsT

```
int metaheuristics.generators.MultiCaseSimulatedAnnealing.countIterationsT [static]
```

6.60.3.4. countRept

```
int metaheuristics.generators.MultiCaseSimulatedAnnealing.countRept [private]
```

6.60.3.5. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.MultiCaseSimulatedAnnealing.ifacceptCandidate [private]
```

6.60.3.6. listStateReference

```
List<State> metaheuristics.generators.MultiCaseSimulatedAnnealing.listStateReference = new ArrayList<State>() [private]
```

6.60.3.7. listTrace

```
List<Float> metaheuristics.generators.MultiCaseSimulatedAnnealing.listTrace = new ArrayList<Float>() [private]
```

6.60.3.8. stateReferenceSA

```
State metaheuristics.generators.MultiCaseSimulatedAnnealing.stateReferenceSA [private]
```

6.60.3.9. strategy

```
StrategyType metaheuristics.generators.MultiCaseSimulatedAnnealing.strategy [private]
```

6.60.3.10. **tfinal**

```
Double metaheuristics.generators.MultiCaseSimulatedAnnealing.tfinal [static]
```

6.60.3.11. **tinitial**

```
Double metaheuristics.generators.MultiCaseSimulatedAnnealing.tinitial [static]
```

6.60.3.12. **typeAcceptation**

```
AcceptType metaheuristics.generators.MultiCaseSimulatedAnnealing.typeAcceptation [private]
```

6.60.3.13. **typeCandidate**

```
CandidateType metaheuristics.generators.MultiCaseSimulatedAnnealing.typeCandidate [private]
```

6.60.3.14. **typeGenerator**

```
GeneratorType metaheuristics.generators.MultiCaseSimulatedAnnealing.typeGenerator [private]
```

6.60.3.15. **weight**

```
float metaheuristics.generators.MultiCaseSimulatedAnnealing.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/MultiCaseSimulatedAnnealing.java](#)

6.61. Referencia de la clase metaheuristics.generators.MultiGenerator

Diagrama de herencia de metaheuristics.generators.MultiGenerator

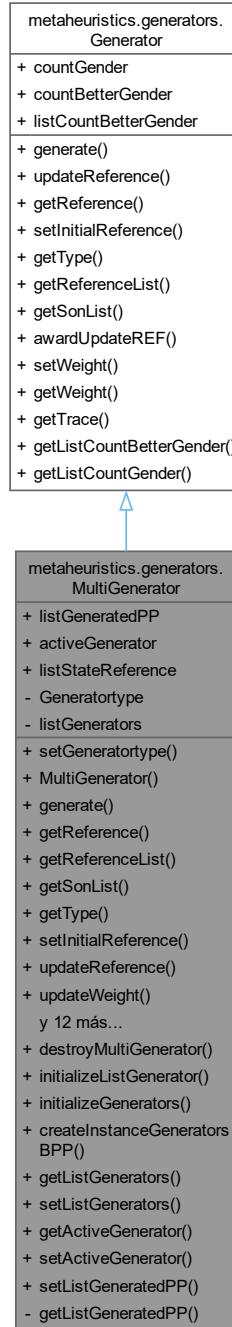
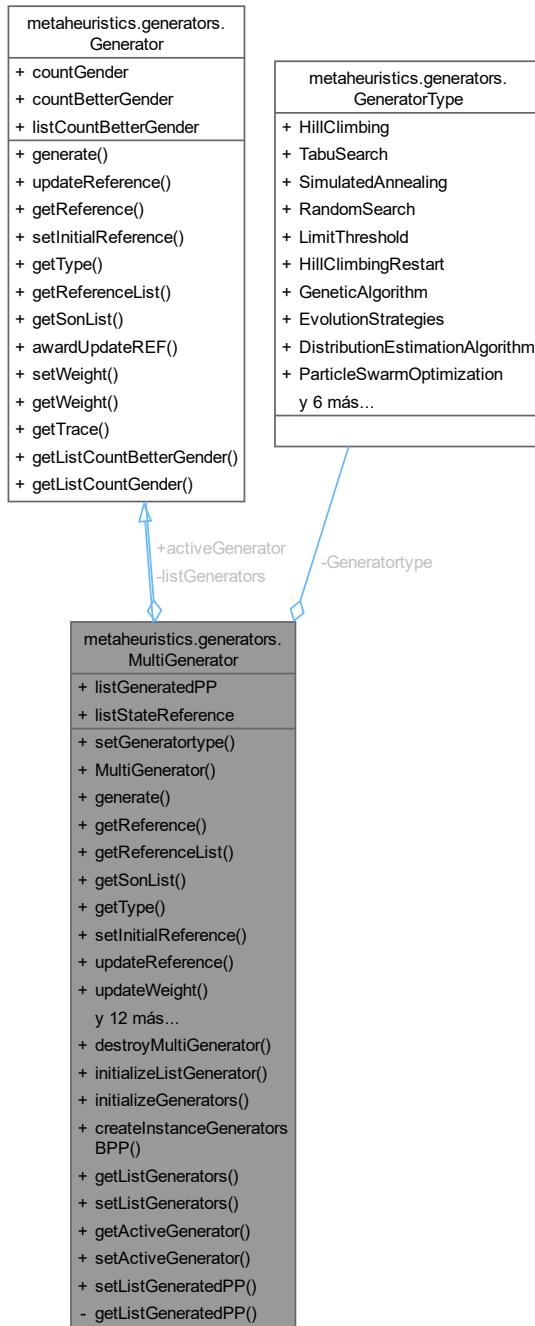


Diagrama de colaboración de metaheuristics.generators.MultiGenerator:



Métodos públicos

- void **setGeneratortype** (GeneratorType generatortype)
- **MultiGenerator** ()
- State **generate** (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

- `State getReference ()`
- `List< State > getReferenceList ()`
- `List< State > getSonList ()`
- `GeneratorType getType ()`
- `void setInitialReference (State stateInitialRef)`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `void updateWeight (State stateCandidate)`
- `boolean searchState (State stateCandidate)`
- `float getWeight ()`
- `Generator roulette ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `void updateAwardSC ()`
- `void updateAwardImp ()`
- `void setWeight (float weight)`
- `float[] getTrace ()`
- `void tournament (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `Object clone ()`
- `int[] getListCountBetterGender ()`
- `int[] getListCountGender ()`

Métodos públicos estáticos

- `static void destroyMultiGenerator ()`
- `static void initializeListGenerator () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `static void initializeGenerators () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `static void createInstanceGeneratorsBPP ()`
- `static Generator[] getListGenerators ()`
- `static void setListGenerators (Generator[] listGenerators)`
- `static Generator getActiveGenerator ()`
- `static void setActiveGenerator (Generator activeGenerator)`
- `static void setListGeneratedPP (List< State > listGeneratedPP)`

Atributos públicos estáticos

- `static List< State > listGeneratedPP = new ArrayList<State> ()`
- `static Generator activeGenerator`
- `static List< State > listStateReference = new ArrayList<State>()`

Métodos privados estáticos

- `static ArrayList< State > getListGeneratedPP ()`

Atributos privados

- `GeneratorType Generatortype`

Atributos estáticos privados

- static Generator[] listGenerators = new Generator[GeneratorType.values().length]

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

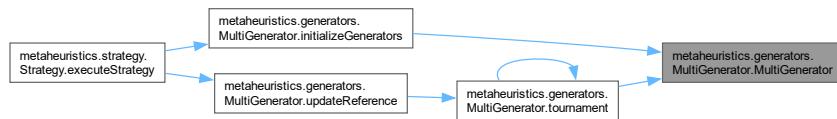
- int countGender
- int countBetterGender
- int[] listCountBetterGender

6.61.1. Documentación de constructores y destructores

6.61.1.1. MultiGenerator()

```
metaheuristics.generators.MultiGenerator.MultiGenerator ()
```

Gráfico de llamadas a esta función:



6.61.2. Documentación de funciones miembro

6.61.2.1. awardUpdateREF()

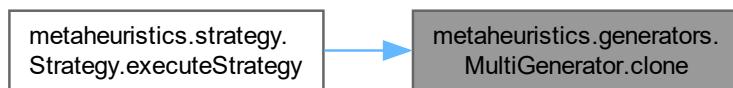
```
boolean metaheuristics.generators.MultiGenerator.awardUpdateREF (
    State stateCandidate)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.61.2.2. clone()

```
Object metaheuristics.generators.MultiGenerator.clone ()
```

Gráfico de llamadas a esta función:



6.61.2.3. `createInstanceGeneratorsBPP()`

```
void metaheuristics.generators.MultiGenerator.createInstanceGeneratorsBPP () [static]
```

Gráfico de llamadas de esta función:

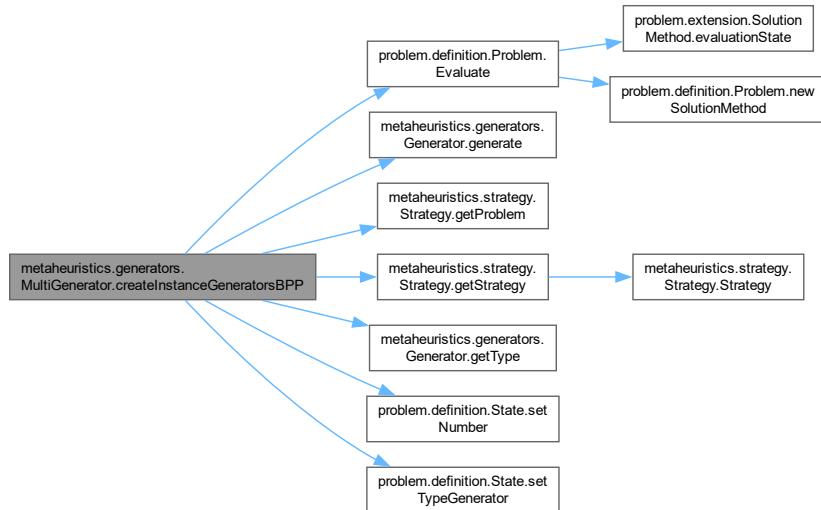


Gráfico de llamadas a esta función:



6.61.2.4. `destroyMultiGenerator()`

```
void metaheuristics.generators.MultiGenerator.destroyMultiGenerator () [static]
```

6.61.2.5. `generate()`

```
State metaheuristics.generators.MultiGenerator.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:

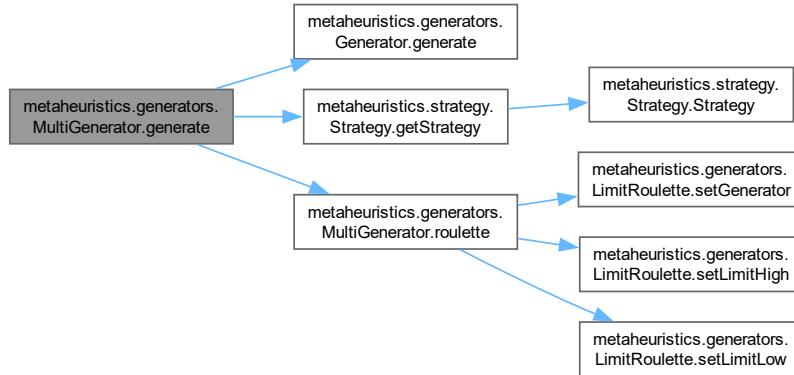
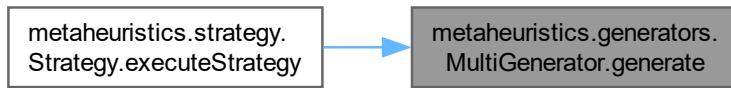


Gráfico de llamadas a esta función:



6.61.2.6. `getActiveGenerator()`

```
Generator metaheuristics.generators.MultiGenerator.getActiveGenerator () [static]
```

6.61.2.7. `getListCountBetterGender()`

```
int[] metaheuristics.generators.MultiGenerator.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.61.2.8. `getListCountGender()`

```
int[] metaheuristics.generators.MultiGenerator.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

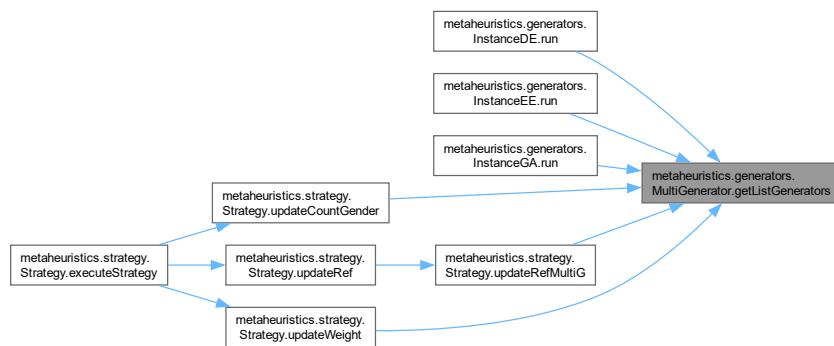
6.61.2.9. getListGeneratedPP()

```
ArrayList< State > metaheuristics.generators.MultiGenerator.getListGeneratedPP () [static],  
[private]
```

6.61.2.10. getListGenerators()

```
Generator[ ] metaheuristics.generators.MultiGenerator.getListGenerators () [static]
```

Gráfico de llamadas a esta función:



6.61.2.11. getReference()

```
State metaheuristics.generators.MultiGenerator.getReference ()
```

Reimplementado de `metaheuristics.generators.Generator`.

6.61.2.12. getReferenceList()

```
List< State > metaheuristics.generators.MultiGenerator.getReferenceList ()
```

Reimplementado de `metaheuristics.generators.Generator`.

Gráfico de llamadas a esta función:



6.61.2.13. `getSonList()`

```
List< State > metaheuristics.generators.MultiGenerator.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.61.2.14. `getTrace()`

```
float[] metaheuristics.generators.MultiGenerator.getTrace ()
```

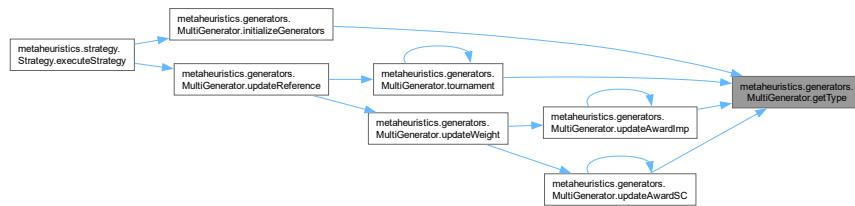
Reimplementado de [metaheuristics.generators.Generator](#).

6.61.2.15. `getType()`

```
GeneratorType metaheuristics.generators.MultiGenerator.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.61.2.16. `getWeight()`

```
float metaheuristics.generators.MultiGenerator.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.61.2.17. initializeGenerators()

```
void metaheuristics.generators.MultiGenerator.initializeGenerators () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException [static]
```

Gráfico de llamadas de esta función:

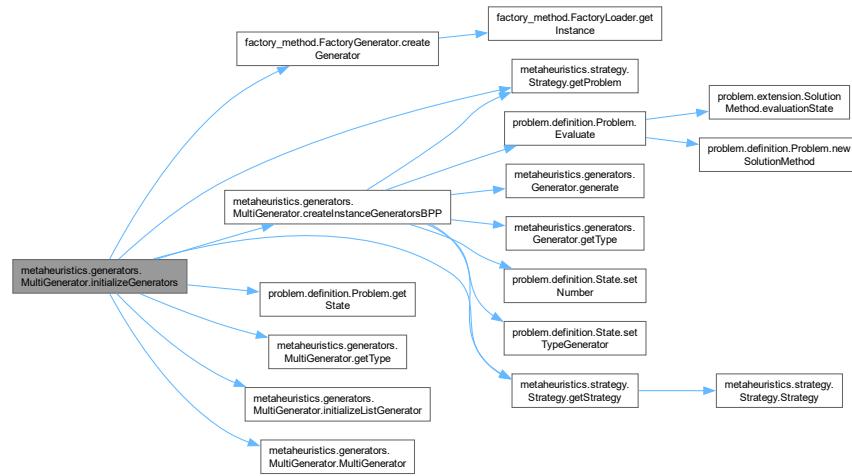


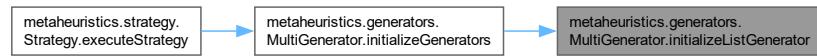
Gráfico de llamadas a esta función:



6.61.2.18. initializeListGenerator()

```
void metaheuristics.generators.MultiGenerator.initializeListGenerator () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException [static]
```

Gráfico de llamadas a esta función:



6.61.2.19. roulette()

```
Generator metaheuristics.generators.MultiGenerator.roulette ()
```

Gráfico de llamadas de esta función:

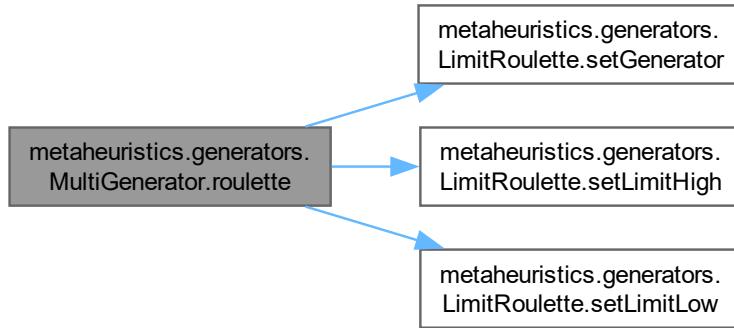
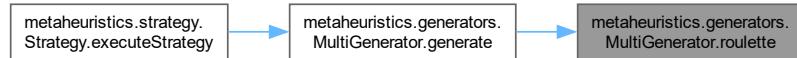


Gráfico de llamadas a esta función:



6.61.2.20. searchState()

```
boolean metaheuristics.generators.MultiGenerator.searchState (
    State stateCandidate)
```

Gráfico de llamadas de esta función:

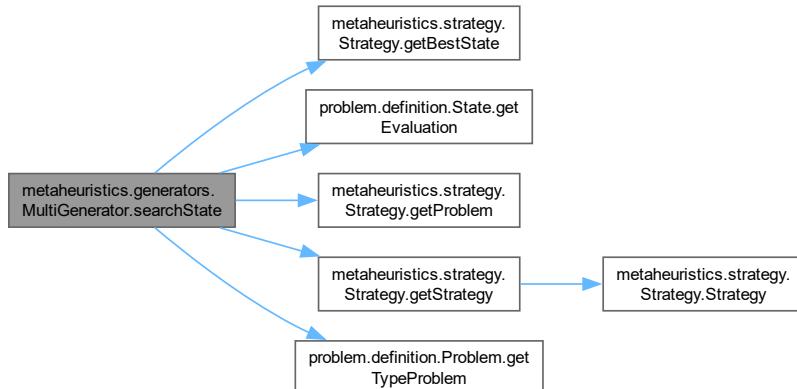


Gráfico de llamadas a esta función:



6.61.2.21. setActiveGenerator()

```
void metaheuristics.generators.MultiGenerator.setActiveGenerator (
    Generator activeGenerator) [static]
```

6.61.2.22. setGeneratortype()

```
void metaheuristics.generators.MultiGenerator.setGeneratortype (
    GeneratorType generatortype)
```

6.61.2.23. setInitialReference()

```
void metaheuristics.generators.MultiGenerator.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.61.2.24. setListGeneratedPP()

```
void metaheuristics.generators.MultiGenerator.setListGeneratedPP (
    List< State > listGeneratedPP) [static]
```

6.61.2.25. setListGenerators()

```
void metaheuristics.generators.MultiGenerator.setListGenerators (
    Generator[] listGenerators) [static]
```

6.61.2.26. setWeight()

```
void metaheuristics.generators.MultiGenerator.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.61.2.27. tournament()

```
void metaheuristics.generators.MultiGenerator.tournament (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Gráfico de llamadas de esta función:

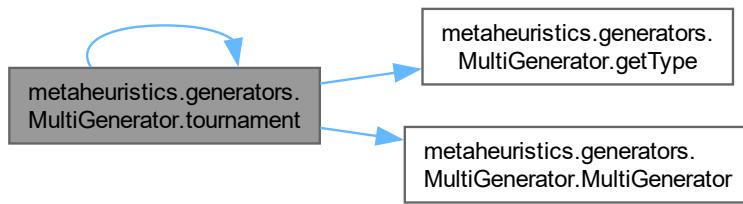
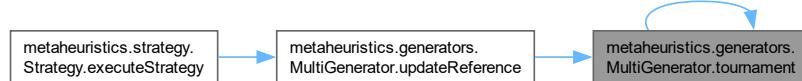


Gráfico de llamadas a esta función:



6.61.2.28. updateAwardImp()

```
void metaheuristics.generators.MultiGenerator.updateAwardImp ()
```

Gráfico de llamadas de esta función:

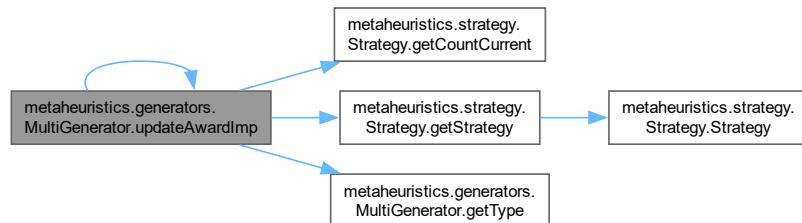


Gráfico de llamadas a esta función:



6.61.2.29. updateAwardSC()

```
void metaheuristics.generators.MultiGenerator.updateAwardSC ()
```

Gráfico de llamadas de esta función:

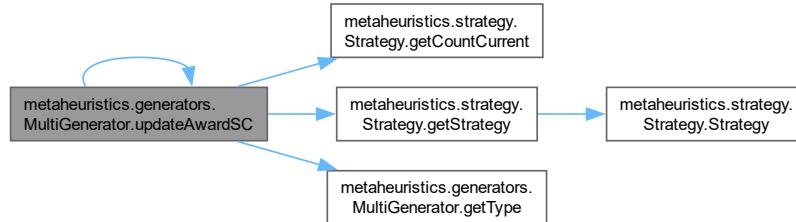


Gráfico de llamadas a esta función:



6.61.2.30. updateReference()

```
void metaheuristics.generators.MultiGenerator.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:

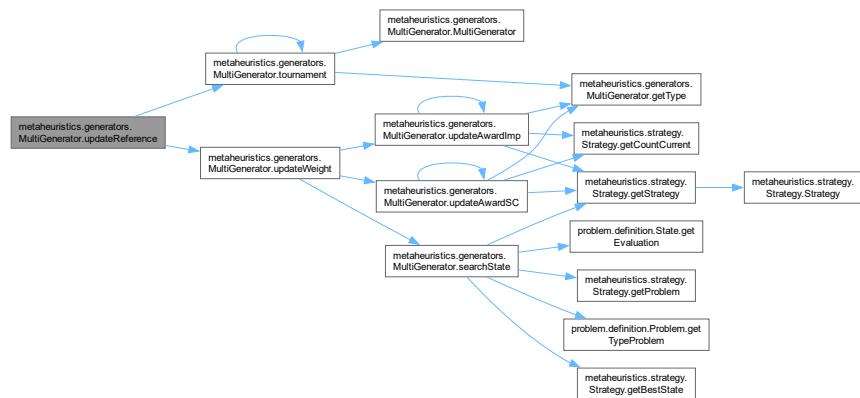


Gráfico de llamadas a esta función:



6.61.2.31. updateWeight()

```
void metaheuristics.generators.MultiGenerator.updateWeight (
    State stateCandidate)
```

Gráfico de llamadas de esta función:

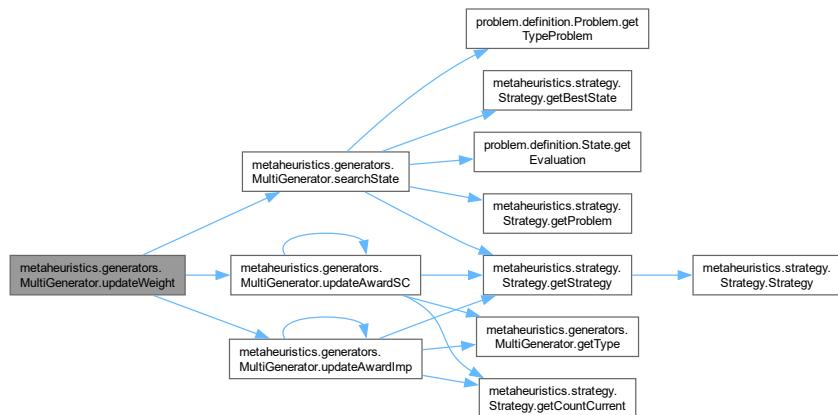
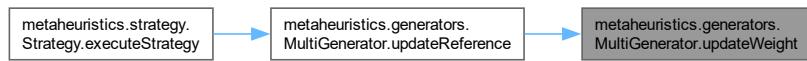


Gráfico de llamadas a esta función:



6.61.3. Documentación de datos miembro

6.61.3.1. activeGenerator

```
Generator metaheuristics.generators.MultiGenerator.activeGenerator [static]
```

6.61.3.2. Generatortype

```
GeneratorType metaheuristics.generators.MultiGenerator.Generatortype [private]
```

6.61.3.3. listGeneratedPP

```
List<State> metaheuristics.generators.MultiGenerator.listGeneratedPP = new ArrayList<State>()
() [static]
```

6.61.3.4. listGenerators

```
Generator [] metaheuristics.generators.MultiGenerator.listGenerators = new Generator[Generator<br>
Type.values().length] [static], [private]
```

6.61.3.5. listStateReference

```
List<State> metaheuristics.generators.MultiGenerator.listStateReference = new ArrayList<State>()
[static]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/MultiGenerator.java](#)

6.62. Referencia de la clase

metaheuristics.generators.MultiobjectiveHillClimbingDistance

Diagrama de herencia de metaheuristics.generators.MultiobjectiveHillClimbingDistance

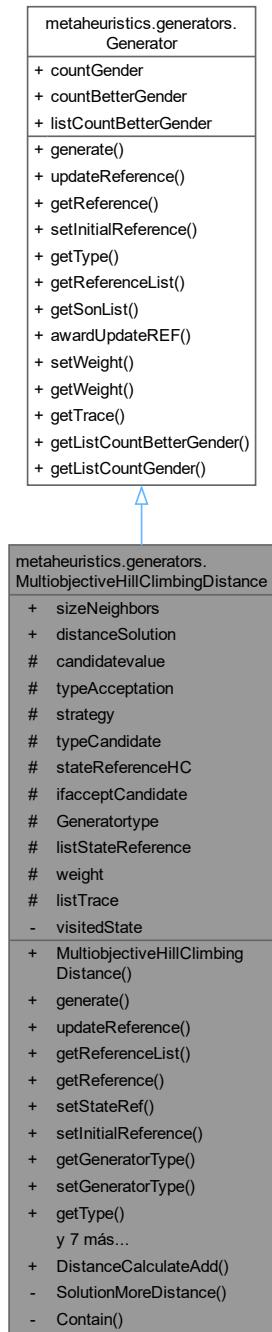
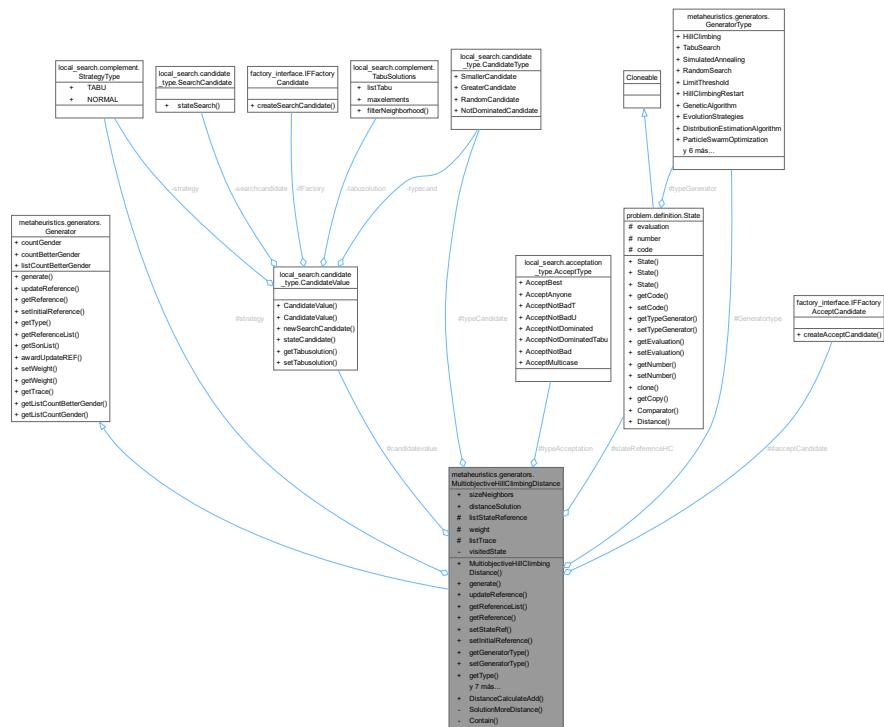


Diagrama de colaboración de metaheuristics.generators.MultiobjectiveHillClimbingDistance:



Métodos públicos

- `MultiobjectiveHillClimbingDistance ()`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `List< State > getReferenceList ()`
- `State getReference ()`
- `void setStateRef (State stateRef)`
- `void setInitialReference (State statelInitialRef)`
- `GeneratorType getGeneratorType ()`
- `void setGeneratorType (GeneratorType Generatortype)`
- `GeneratorType getType ()`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `int[] getCountBetterGender ()`
- `int[] getCountGender ()`
- `float[] getTrace ()`

Métodos públicos estáticos

- `static List< Double > DistanceCalculateAdd (List< State > solution)`

Atributos públicos estáticos

- static int `sizeNeighbors`
- static List< Double > `distanceSolution` = new ArrayList<Double>()

Atributos protegidos

- CandidateValue `candidatevalue`
- AcceptType `typeAcceptation`
- StrategyType `strategy`
- CandidateType `typeCandidate`
- State `stateReferenceHC`
- IFFactoryAcceptCandidate `ifacceptCandidate`
- GeneratorType `Generatortype`
- List< State > `listStateReference` = new ArrayList<State>()
- float `weight`
- List< Float > `listTrace` = new ArrayList<Float>()

Métodos privados

- State `SolutionMoreDistance` (List< State > state, List< Double > `distanceSolution`)
- boolean `Contain` (State state)

Atributos privados

- List< State > `visitedState` = new ArrayList<State>()

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.62.1. Documentación de constructores y destructores

6.62.1.1. [MultiobjectiveHillClimbingDistance\(\)](#)

```
metaheuristics.generators.MultiobjectiveHillClimbingDistance.MultiobjectiveHillClimbing←
Distance ()
```

6.62.2. Documentación de funciones miembro

6.62.2.1. [awardUpdateREF\(\)](#)

```
boolean metaheuristics.generators.MultiobjectiveHillClimbingDistance.awardUpdateREF (
    State stateCandidate)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.2. Contain()

```
boolean metaheuristics.generators.MultiobjectiveHillClimbingDistance.Contain (
    State state)  [private]
```

Gráfico de llamadas de esta función:

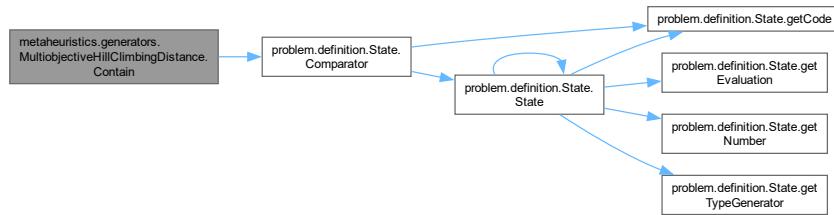


Gráfico de llamadas a esta función:



6.62.2.3. DistanceCalculateAdd()

```
List< Double > metaheuristics.generators.MultiobjectiveHillClimbingDistance.DistanceCalculate←
Add (
    List< State > solution)  [static]
```

Gráfico de llamadas de esta función:

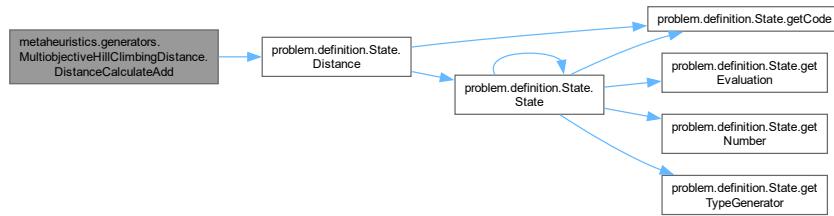
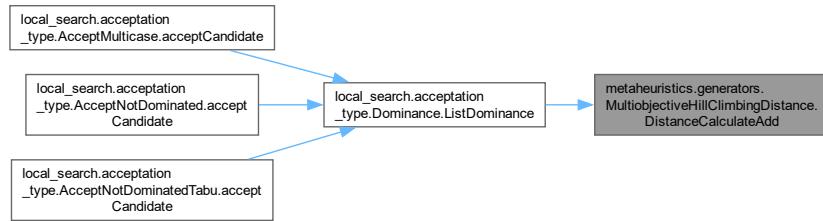


Gráfico de llamadas a esta función:

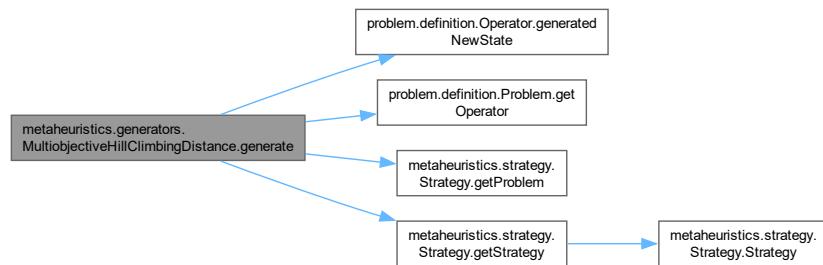


6.62.2.4. generate()

```
State metaheuristics.generators.MultiobjectiveHillClimbingDistance.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.62.2.5. getGeneratorType()

```
GeneratorType metaheuristics.generators.MultiobjectiveHillClimbingDistance.getGeneratorType ()
```

6.62.2.6. getListCountBetterGender()

```
int[] metaheuristics.generators.MultiobjectiveHillClimbingDistance.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.7. getListCountGender()

```
int[] metaheuristics.generators.MultiobjectiveHillClimbingDistance.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.8. getReference()

```
State metaheuristics.generators.MultiobjectiveHillClimbingDistance.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.9. getReferenceList()

```
List< State > metaheuristics.generators.MultiobjectiveHillClimbingDistance.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.62.2.10. getSonList()

```
List< State > metaheuristics.generators.MultiobjectiveHillClimbingDistance.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.11. getTrace()

```
float[] metaheuristics.generators.MultiobjectiveHillClimbingDistance.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.12. getType()

```
GeneratorType metaheuristics.generators.MultiobjectiveHillClimbingDistance.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.13. `getWeight()`

```
float metaheuristics.generators.MultiobjectiveHillClimbingDistance.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.14. `setGeneratorType()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingDistance.setGeneratorType (
    GeneratorType Generatortype)
```

6.62.2.15. `setInitialReference()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingDistance.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.16. `setStateRef()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingDistance.setStateRef (
    State stateRef)
```

6.62.2.17. `setWeight()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingDistance.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.62.2.18. `SolutionMoreDistance()`

```
State metaheuristics.generators.MultiobjectiveHillClimbingDistance.SolutionMoreDistance (
    List< State > state,
    List< Double > distanceSolution) [private]
```

Gráfico de llamadas a esta función:

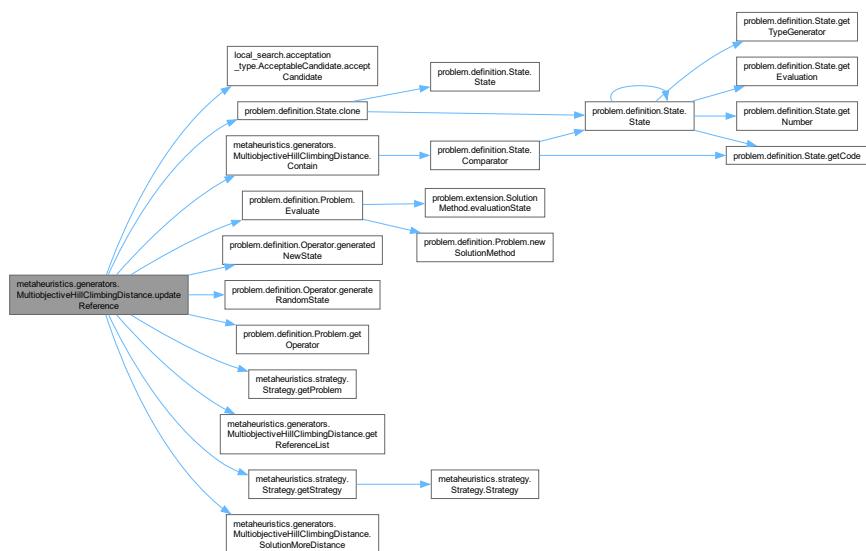


6.62.2.19. updateReference()

```
void metaheuristics.generators.MultiobjectiveHillClimbingDistance.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.62.3. Documentación de datos miembro

6.62.3.1. candidatevalue

[CandidateValue](#) metaheuristics.generators.MultiobjectiveHillClimbingDistance.candidatevalue
[protected]

6.62.3.2. distanceSolution

```
List<Double> metaheuristics.generators.MultiobjectiveHillClimbingDistance.distanceSolution =
new ArrayList<Double>() [static]
```

6.62.3.3. GeneratorType

[GeneratorType](#) metaheuristics.generators.MultiobjectiveHillClimbingDistance.GeneratorType [protected]

6.62.3.4. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.MultiobjectiveHillClimbingDistance.ifaccept←
Candidate [protected]
```

6.62.3.5. listStateReference

```
List<State> metaheuristics.generators.MultiobjectiveHillClimbingDistance.listStateReference =
new ArrayList<State>() [protected]
```

6.62.3.6. listTrace

```
List<Float> metaheuristics.generators.MultiobjectiveHillClimbingDistance.listTrace = new
ArrayList<Float>() [protected]
```

6.62.3.7. sizeNeighbors

```
int metaheuristics.generators.MultiobjectiveHillClimbingDistance.sizeNeighbors [static]
```

6.62.3.8. stateReferenceHC

```
State metaheuristics.generators.MultiobjectiveHillClimbingDistance.stateReferenceHC [protected]
```

6.62.3.9. strategy

```
StrategyType metaheuristics.generators.MultiobjectiveHillClimbingDistance.strategy [protected]
```

6.62.3.10. typeAcceptation

```
AcceptType metaheuristics.generators.MultiobjectiveHillClimbingDistance.typeAcceptation [protected]
```

6.62.3.11. typeCandidate

```
CandidateType metaheuristics.generators.MultiobjectiveHillClimbingDistance.typeCandidate [protected]
```

6.62.3.12. visitedState

```
List<State> metaheuristics.generators.MultiobjectiveHillClimbingDistance.visitedState = new
ArrayList<State>() [private]
```

6.62.3.13. weight

```
float metaheuristics.generators.MultiobjectiveHillClimbingDistance.weight [protected]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/MultiobjectiveHillClimbingDistance.java](#)

6.63. Referencia de la clase

metaheuristics.generators.MultiobjectiveHillClimbingRestart

Diagrama de herencia de metaheuristics.generators.MultiobjectiveHillClimbingRestart

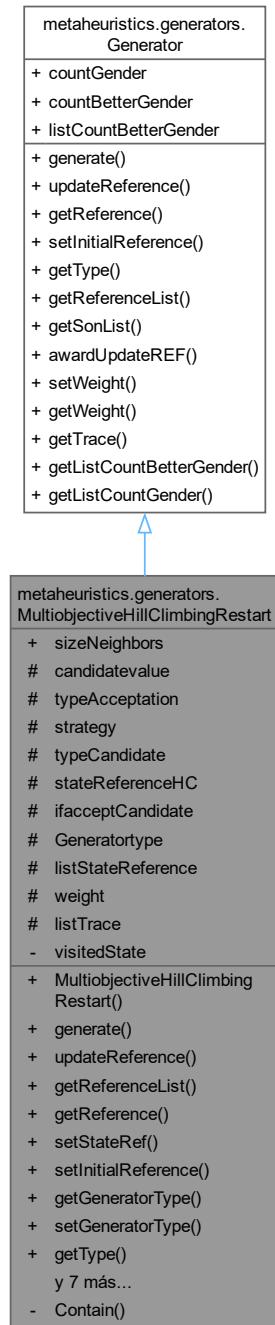
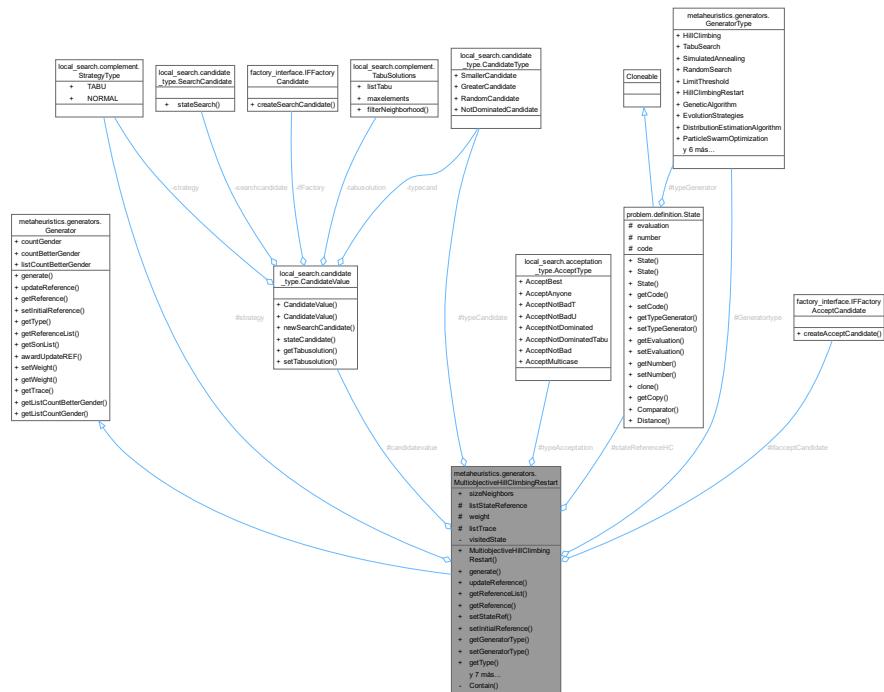


Diagrama de colaboración de metaheuristics.generators.MultiobjectiveHillClimbingRestart:



Métodos públicos

- [MultiobjectiveHillClimbingRestart \(\)](#)
- [State generate \(Integer operatornumber\)](#) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)
- [void updateReference \(State stateCandidate, Integer countIterationsCurrent\)](#) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)
- [List< State > getReferenceList \(\)](#)
- [State getReference \(\)](#)
- [void setStateRef \(State stateRef\)](#)
- [void setInitialReference \(State statelInitialRef\)](#)
- [GeneratorType getGeneratorType \(\)](#)
- [void setGeneratorType \(GeneratorType Generatortype\)](#)
- [GeneratorType getType \(\)](#)
- [List< State > getSonList \(\)](#)
- [boolean awardUpdateREF \(State stateCandidate\)](#)
- [float getWeight \(\)](#)
- [void setWeight \(float weight\)](#)
- [float\[\] getTrace \(\)](#)
- [int\[\] getCountBetterGender \(\)](#)
- [int\[\] getCountGender \(\)](#)

Atributos públicos estáticos

- static int [sizeNeighbors](#)

Atributos protegidos

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceHC`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- `GeneratorType Generatortype`
- `List< State > listStateReference = new ArrayList<State>()`
- `float weight`
- `List< Float > listTrace = new ArrayList<Float>()`

Métodos privados

- `boolean Contain (State state)`

Atributos privados

- `List< State > visitedState = new ArrayList<State>()`

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- `int countGender`
- `int countBetterGender`
- `int[] listCountBetterGender`

6.63.1. Documentación de constructores y destructores

6.63.1.1. [MultiobjectiveHillClimbingRestart\(\)](#)

```
metaheuristics.generators.MultiobjectiveHillClimbingRestart.MultiobjectiveHillClimbingRestart
()
```

6.63.2. Documentación de funciones miembro

6.63.2.1. [awardUpdateREF\(\)](#)

```
boolean metaheuristics.generators.MultiobjectiveHillClimbingRestart.awardUpdateREF (
    State stateCandidate)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.2. Contain()

```
boolean metaheuristics.generators.MultiobjectiveHillClimbingRestart.Contain (
    State state)  [private]
```

Gráfico de llamadas de esta función:

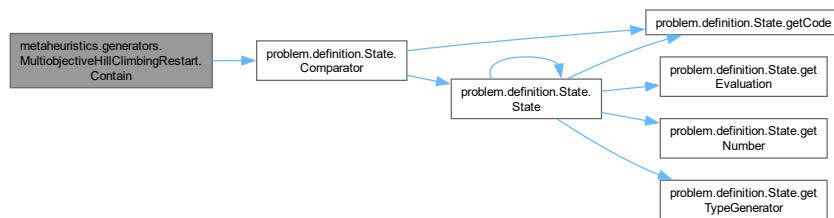


Gráfico de llamadas a esta función:

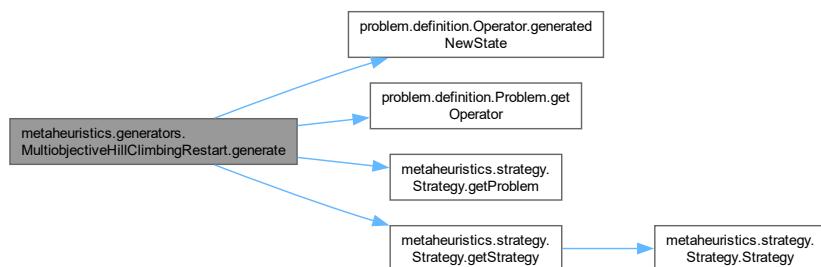


6.63.2.3. generate()

```
State metaheuristics.generators.MultiobjectiveHillClimbingRestart.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.63.2.4. `getGeneratorType()`

```
GeneratorType metaheuristics.generators.MultiobjectiveHillClimbingRestart.getGeneratorType ()
```

6.63.2.5. `getListCountBetterGender()`

```
int[] metaheuristics.generators.MultiobjectiveHillClimbingRestart.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.6. `getListCountGender()`

```
int[] metaheuristics.generators.MultiobjectiveHillClimbingRestart.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.7. `getReference()`

```
State metaheuristics.generators.MultiobjectiveHillClimbingRestart.getReference ()
```

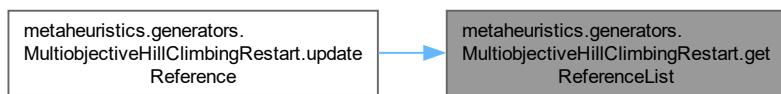
Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.8. `getReferenceList()`

```
List< State > metaheuristics.generators.MultiobjectiveHillClimbingRestart.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.63.2.9. `getSonList()`

```
List< State > metaheuristics.generators.MultiobjectiveHillClimbingRestart.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.10. `getTrace()`

```
float[ ] metaheuristics.generators.MultiobjectiveHillClimbingRestart.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.11. `getType()`

```
GeneratorType metaheuristics.generators.MultiobjectiveHillClimbingRestart.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.12. `getWeight()`

```
float metaheuristics.generators.MultiobjectiveHillClimbingRestart.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.13. `setGeneratorType()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingRestart.setGeneratorType (GeneratorType GeneratorType)
```

6.63.2.14. `setInitialReference()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingRestart.setInitialReference (State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.15. `setStateRef()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingRestart.setStateRef (State stateRef)
```

6.63.2.16. `setWeight()`

```
void metaheuristics.generators.MultiobjectiveHillClimbingRestart.setWeight (float weight)
```

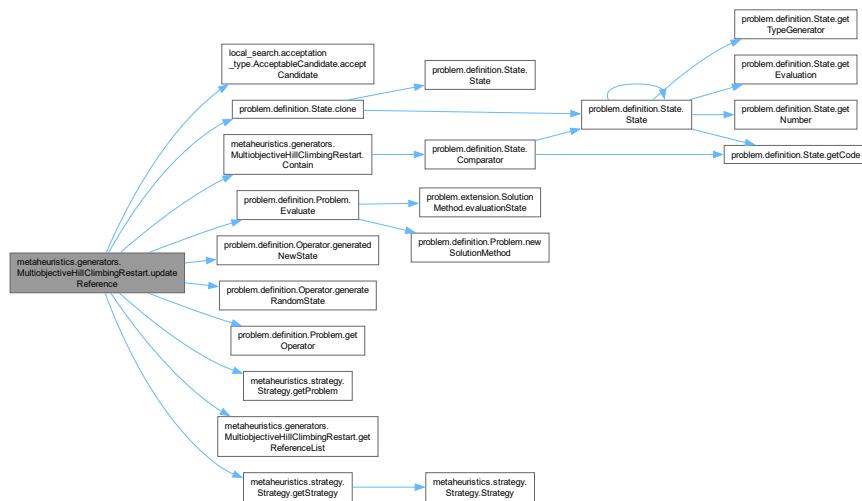
Reimplementado de [metaheuristics.generators.Generator](#).

6.63.2.17. updateReference()

```
void metaheuristics.generators.MultiobjectiveHillClimbingRestart.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.63.3. Documentación de datos miembro

6.63.3.1. candidatevalue

`CandidateValue` `metaheuristics.generators.MultiobjectiveHillClimbingRestart.candidatevalue` [protected]

6.63.3.2. GeneratorType

`GeneratorType` `metaheuristics.generators.MultiobjectiveHillClimbingRestart.GeneratorType` [protected]

6.63.3.3. ifacceptCandidate

`IFFactoryAcceptCandidate` `metaheuristics.generators.MultiobjectiveHillClimbingRestart.ifacceptCandidate` [protected]

6.63.3.4. listStateReference

```
List<State> metaheuristics.generators.MultiobjectiveHillClimbingRestart.listStateReference =  
new ArrayList<State>() [protected]
```

6.63.3.5. listTrace

```
List<Float> metaheuristics.generators.MultiobjectiveHillClimbingRestart.listTrace = new Array←  
List<Float>() [protected]
```

6.63.3.6. sizeNeighbors

```
int metaheuristics.generators.MultiobjectiveHillClimbingRestart.sizeNeighbors [static]
```

6.63.3.7. stateReferenceHC

```
State metaheuristics.generators.MultiobjectiveHillClimbingRestart.stateReferenceHC [protected]
```

6.63.3.8. strategy

```
StrategyType metaheuristics.generators.MultiobjectiveHillClimbingRestart.strategy [protected]
```

6.63.3.9. typeAcceptation

```
AcceptType metaheuristics.generators.MultiobjectiveHillClimbingRestart.typeAcceptation [protected]
```

6.63.3.10. typeCandidate

```
CandidateType metaheuristics.generators.MultiobjectiveHillClimbingRestart.typeCandidate [protected]
```

6.63.3.11. visitedState

```
List<State> metaheuristics.generators.MultiobjectiveHillClimbingRestart.visitedState = new  
ArrayList<State>() [private]
```

6.63.3.12. weight

```
float metaheuristics.generators.MultiobjectiveHillClimbingRestart.weight [protected]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/MultiobjectiveHillClimbingRestart.java](#)

6.64. Referencia de la clase

metaheuristics.generators.MultiobjectiveStochasticHillClimbing

Diagrama de herencia de metaheuristics.generators.MultiobjectiveStochasticHillClimbing

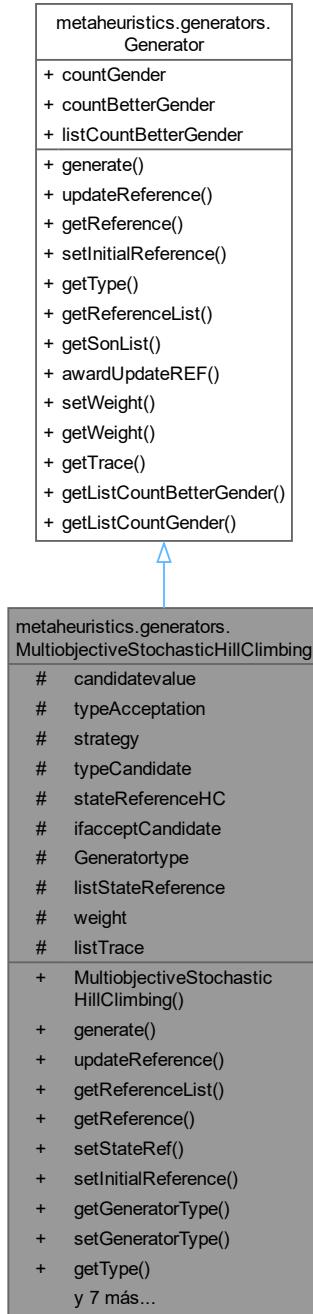
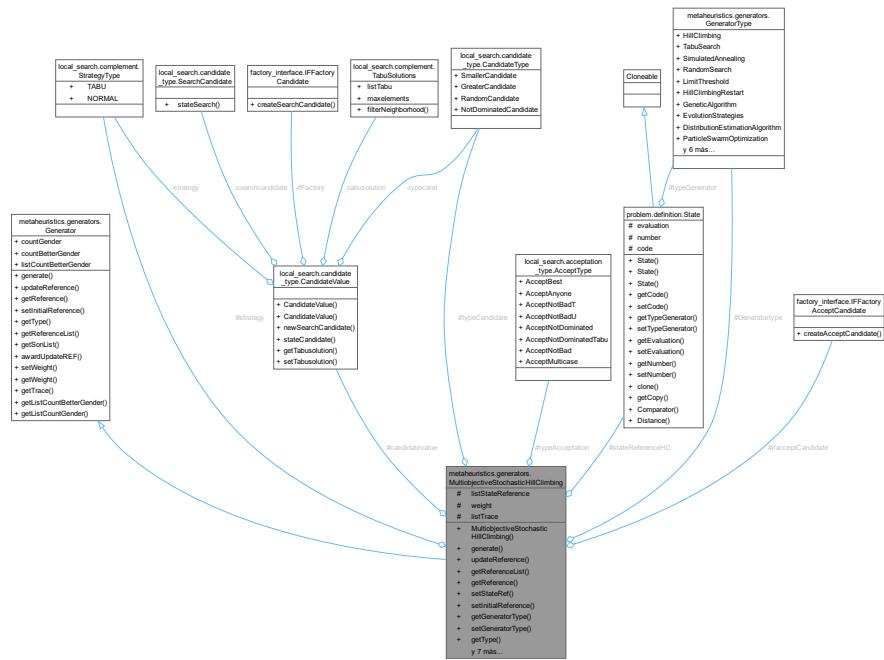


Diagrama de colaboración de metaheuristics.generators.MultiobjectiveStochasticHillClimbing:



Métodos públicos

- `MultiobjectiveStochasticHillClimbing()`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `List< State > getReferenceList ()`
- `State getReference ()`
- `void setStateRef (State stateRef)`
- `void setInitialReference (State stateInitialRef)`
- `GeneratorType getGeneratorType ()`
- `void setGeneratorType (GeneratorType Generatortype)`
- `GeneratorType getType ()`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `float[] getTrace ()`
- `int[] getCountBetterGender ()`
- `int[] getCountGender ()`

Atributos protegidos

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`

- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceHC`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- `GeneratorType Generatortype`
- `List< State > listStateReference = new ArrayList<State>()`
- `float weight`
- `List< Float > listTrace = new ArrayList<Float>()`

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- `int countGender`
- `int countBetterGender`
- `int[] listCountBetterGender`

6.64.1. Documentación de constructores y destructores

6.64.1.1. `MultiobjectiveStochasticHillClimbing()`

```
metaheuristics.generators.MultiobjectiveStochasticHillClimbing.MultiobjectiveStochasticHill←
Climbing ()
```

6.64.2. Documentación de funciones miembro

6.64.2.1. `awardUpdateREF()`

```
boolean metaheuristics.generators.MultiobjectiveStochasticHillClimbing.awardUpdateREF (
    State stateCandidate)
```

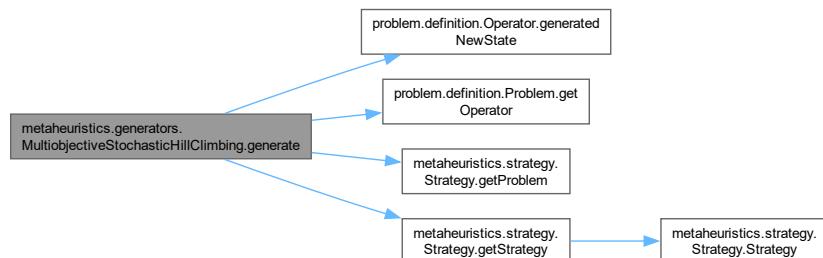
Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.2. `generate()`

```
State metaheuristics.generators.MultiobjectiveStochasticHillClimbing.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.64.2.3. getGeneratorType()

```
GeneratorType metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getGeneratorType()
()
```

6.64.2.4. getListCountBetterGender()

```
int[] metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getListCountBetterGender()
()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.5. getListCountGender()

```
int[] metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getListCountGender()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.6. getReference()

```
State metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getReference()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.7. getReferenceList()

```
List< State > metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getReferenceList()
()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.64.2.8. getSonList()

```
List< State > metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getSonList()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.9. `getTrace()`

```
float[ ] metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.10. `getType()`

```
GeneratorType metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.11. `getWeight()`

```
float metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.12. `setGeneratorType()`

```
void metaheuristics.generators.MultiobjectiveStochasticHillClimbing.setGeneratorType (
    GeneratorType Generatortype)
```

6.64.2.13. `setInitialReference()`

```
void metaheuristics.generators.MultiobjectiveStochasticHillClimbing.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.14. `setStateRef()`

```
void metaheuristics.generators.MultiobjectiveStochasticHillClimbing.setStateRef (
    State stateRef)
```

6.64.2.15. `setWeight()`

```
void metaheuristics.generators.MultiobjectiveStochasticHillClimbing.setWeight (
    float weight)
```

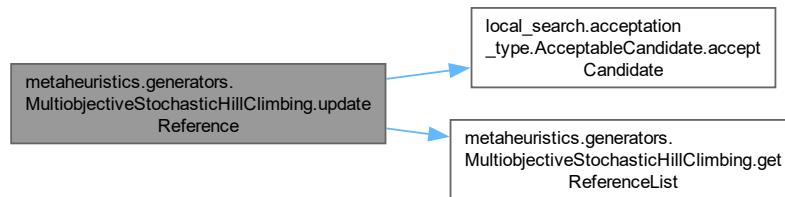
Reimplementado de [metaheuristics.generators.Generator](#).

6.64.2.16. updateReference()

```
void metaheuristics.generators.MultiobjectiveStochasticHillClimbing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.64.3. Documentación de datos miembro

6.64.3.1. candidatevalue

```
CandidateValue metaheuristics.generators.MultiobjectiveStochasticHillClimbing.candidatevalue
[protected]
```

6.64.3.2. GeneratorType

```
GeneratorType metaheuristics.generators.MultiobjectiveStochasticHillClimbing.GeneratorType
[protected]
```

6.64.3.3. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.MultiobjectiveStochasticHillClimbing.ifacceptCandidate
[protected]
```

6.64.3.4. listStateReference

```
List<State> metaheuristics.generators.MultiobjectiveStochasticHillClimbing.listStateReference
= new ArrayList<State>() [protected]
```

6.64.3.5. **listTrace**

```
List<Float> metaheuristics.generators.MultiobjectiveStochasticHillClimbing.listTrace = new  
ArrayList<Float>() [protected]
```

6.64.3.6. **stateReferenceHC**

```
State metaheuristics.generators.MultiobjectiveStochasticHillClimbing.stateReferenceHC [protected]
```

6.64.3.7. **strategy**

```
StrategyType metaheuristics.generators.MultiobjectiveStochasticHillClimbing.strategy [protected]
```

6.64.3.8. **typeAcceptation**

```
AcceptType metaheuristics.generators.MultiobjectiveStochasticHillClimbing.typeAcceptation  
[protected]
```

6.64.3.9. **typeCandidate**

```
CandidateType metaheuristics.generators.MultiobjectiveStochasticHillClimbing.typeCandidate  
[protected]
```

6.64.3.10. **weight**

```
float metaheuristics.generators.MultiobjectiveStochasticHillClimbing.weight [protected]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/MultiobjectiveStochasticHillClimbing.java](#)

6.65. Referencia de la clase metaheuristics.generators.MultiobjectiveTabuSearch

Diagrama de herencia de metaheuristics.generators.MultiobjectiveTabuSearch

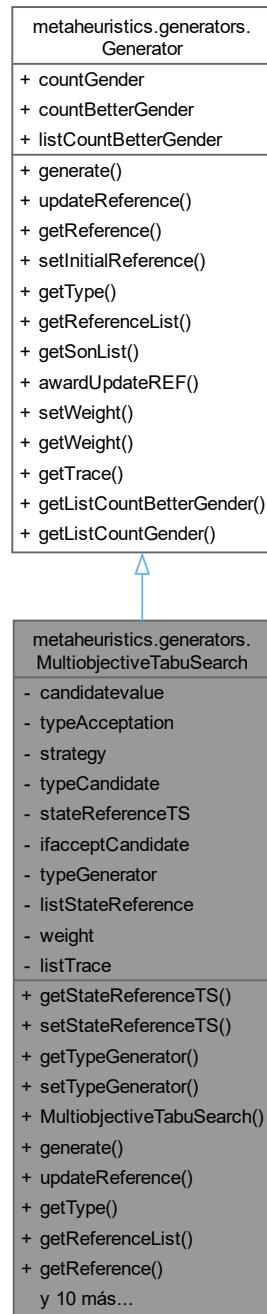
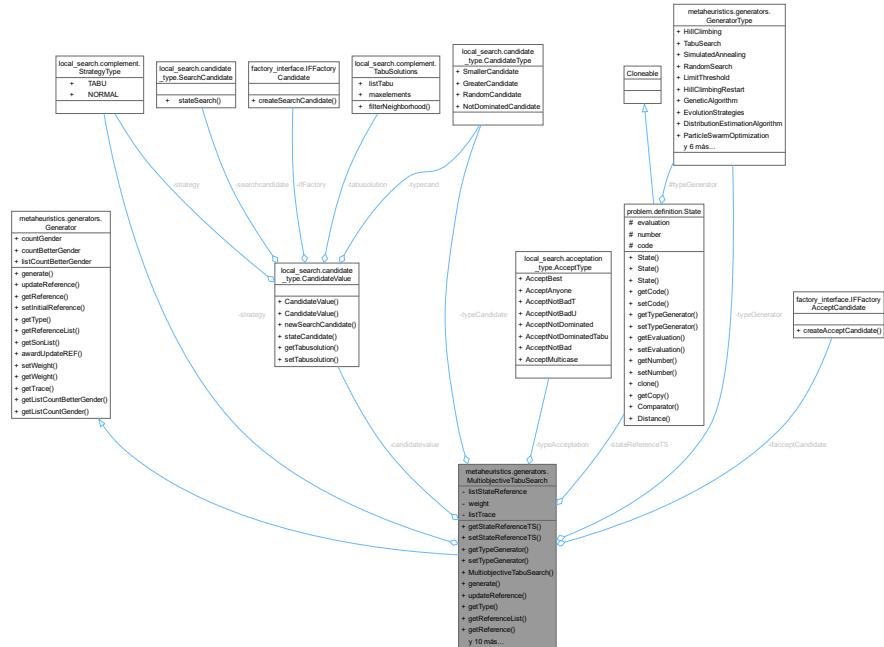


Diagrama de colaboración de metaheuristics.generators.MultiobjectiveTabuSearch:



Métodos públicos

- **State getStateReferenceTS ()**
- **void setStateReferenceTS (State stateReferenceTS)**
- **GeneratorType getTypeGenerator ()**
- **void setTypeGenerator (GeneratorType typeGenerator)**
- **MultiobjectiveTabuSearch ()**
- **State generate (Integer operatornumber)** throws **IllegalArgumentException**, **SecurityException**, **ClassNotFoundException**, **InstantiationException**, **IllegalAccessException**, **InvocationTargetException**, **NoSuchMethodException**
- **void updateReference (State stateCandidate, Integer countIterationsCurrent)** throws **IllegalArgumentException**, **SecurityException**, **ClassNotFoundException**, **InstantiationException**, **IllegalAccessException**, **InvocationTargetException**, **NoSuchMethodException**
- **GeneratorType getType ()**
- **List< State > getReferenceList ()**
- **State getReference ()**
- **void setInitialReference (State statelInitialRef)**
- **void setStateRef (State stateRef)**
- **List< State > getSonList ()**
- **void setTypeCandidate (CandidateType typeCandidate)**
- **boolean awardUpdateREF (State stateCandidate)**
- **float getWeight ()**
- **void setWeight (float weight)**
- **int[] getCountBetterGender ()**
- **int[] getCountGender ()**
- **float[] getTrace ()**

Atributos privados

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceTS`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- `GeneratorType typeGenerator`
- `List< State > listStateReference = new ArrayList<State>()`
- `float weight`
- `List< Float > listTrace = new ArrayList<Float>()`

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

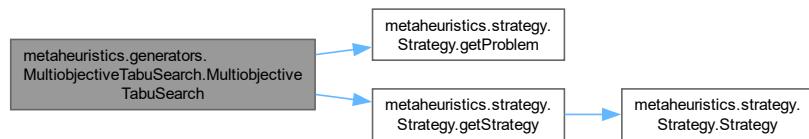
- `int countGender`
- `int countBetterGender`
- `int[] listCountBetterGender`

6.65.1. Documentación de constructores y destructores

6.65.1.1. MultiobjectiveTabuSearch()

```
metaheuristics.generators.MultiobjectiveTabuSearch.MultiobjectiveTabuSearch ()
```

Gráfico de llamadas de esta función:



6.65.2. Documentación de funciones miembro

6.65.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.MultiobjectiveTabuSearch.awardUpdateREF (\n    State stateCandidate)
```

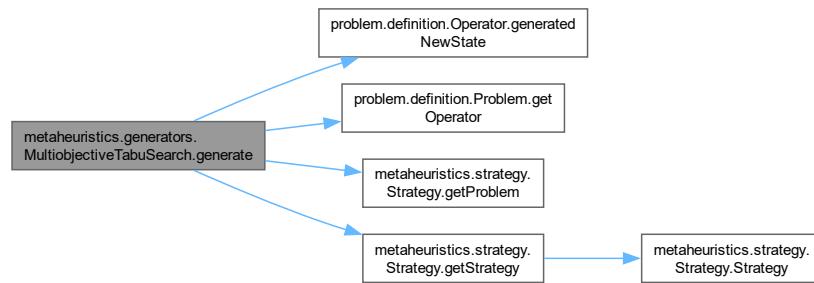
Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.2. generate()

```
State metaheuristics.generators.MultiobjectiveTabuSearch.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.65.2.3. getListCountBetterGender()

```
int[] metaheuristics.generators.MultiobjectiveTabuSearch.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.4. getListCountGender()

```
int[] metaheuristics.generators.MultiobjectiveTabuSearch.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.5. getReference()

```
State metaheuristics.generators.MultiobjectiveTabuSearch.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.6. `getReferenceList()`

```
List< State > metaheuristics.generators.MultiobjectiveTabuSearch.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.65.2.7. `getSonList()`

```
List< State > metaheuristics.generators.MultiobjectiveTabuSearch.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.8. `getStateReferenceTS()`

```
State metaheuristics.generators.MultiobjectiveTabuSearch.getStateReferenceTS ()
```

6.65.2.9. `getTrace()`

```
float[] metaheuristics.generators.MultiobjectiveTabuSearch.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.10. `getType()`

```
GeneratorType metaheuristics.generators.MultiobjectiveTabuSearch.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.11. `getTypeGenerator()`

```
GeneratorType metaheuristics.generators.MultiobjectiveTabuSearch.getTypeGenerator ()
```

6.65.2.12. `getWeight()`

```
float metaheuristics.generators.MultiobjectiveTabuSearch.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.13. `setInitialReference()`

```
void metaheuristics.generators.MultiobjectiveTabuSearch.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.14. `setStateRef()`

```
void metaheuristics.generators.MultiobjectiveTabuSearch.setStateRef (
    State stateRef)
```

6.65.2.15. `setStateReferenceTS()`

```
void metaheuristics.generators.MultiobjectiveTabuSearch.setStateReferenceTS (
    State stateReferenceTS)
```

6.65.2.16. `setTypeCandidate()`

```
void metaheuristics.generators.MultiobjectiveTabuSearch.setTypeCandidate (
    CandidateType typeCandidate)
```

6.65.2.17. `setTypeGenerator()`

```
void metaheuristics.generators.MultiobjectiveTabuSearch.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.65.2.18. `setWeight()`

```
void metaheuristics.generators.MultiobjectiveTabuSearch.setWeight (
    float weight)
```

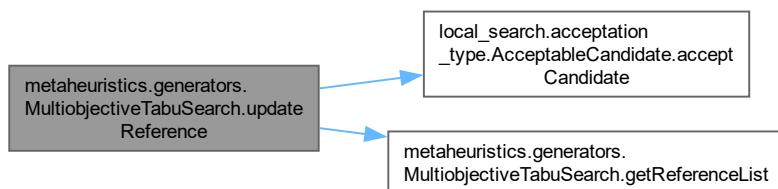
Reimplementado de [metaheuristics.generators.Generator](#).

6.65.2.19. `updateReference()`

```
void metaheuristics.generators.MultiobjectiveTabuSearch.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.65.3. Documentación de datos miembro

6.65.3.1. candidatevalue

```
CandidateValue metaheuristics.generators.MultiobjectiveTabuSearch.candidatevalue [private]
```

6.65.3.2. ifacceptCandidate

```
IFFFactoryAcceptCandidate metaheuristics.generators.MultiobjectiveTabuSearch.ifacceptCandidate [private]
```

6.65.3.3. listStateReference

```
List<State> metaheuristics.generators.MultiobjectiveTabuSearch.listStateReference = new ArrayList<State>()  
[private]
```

6.65.3.4. listTrace

```
List<Float> metaheuristics.generators.MultiobjectiveTabuSearch.listTrace = new ArrayList<Float>()  
[private]
```

6.65.3.5. stateReferenceTS

```
State metaheuristics.generators.MultiobjectiveTabuSearch.stateReferenceTS [private]
```

6.65.3.6. strategy

```
StrategyType metaheuristics.generators.MultiobjectiveTabuSearch.strategy [private]
```

6.65.3.7. typeAcceptation

```
AcceptType metaheuristics.generators.MultiobjectiveTabuSearch.typeAcceptation [private]
```

6.65.3.8. typeCandidate

```
CandidateType metaheuristics.generators.MultiobjectiveTabuSearch.typeCandidate [private]
```

6.65.3.9. typeGenerator

```
GeneratorType metaheuristics.generators.MultiobjectiveTabuSearch.typeGenerator [private]
```

6.65.3.10. weight

```
float metaheuristics.generators.MultiobjectiveTabuSearch.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/MultiobjectiveTabuSearch.java](#)

6.66. Referencia de la clase problem.extension.MultiObjetivoPuro

Diagrama de herencia de problem.extension.MultiObjetivoPuro

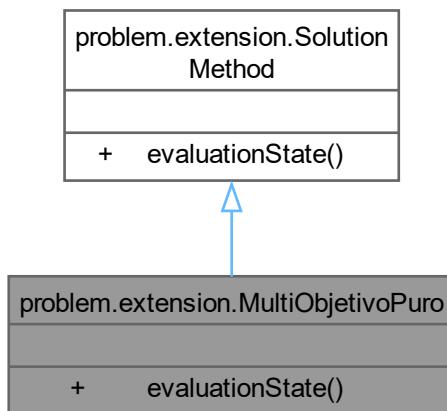
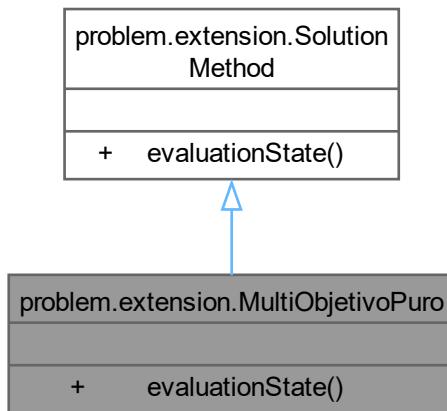


Diagrama de colaboración de problem.extension.MultiObjetivoPuro:



Métodos públicos

- void [evaluationState \(State state\)](#)

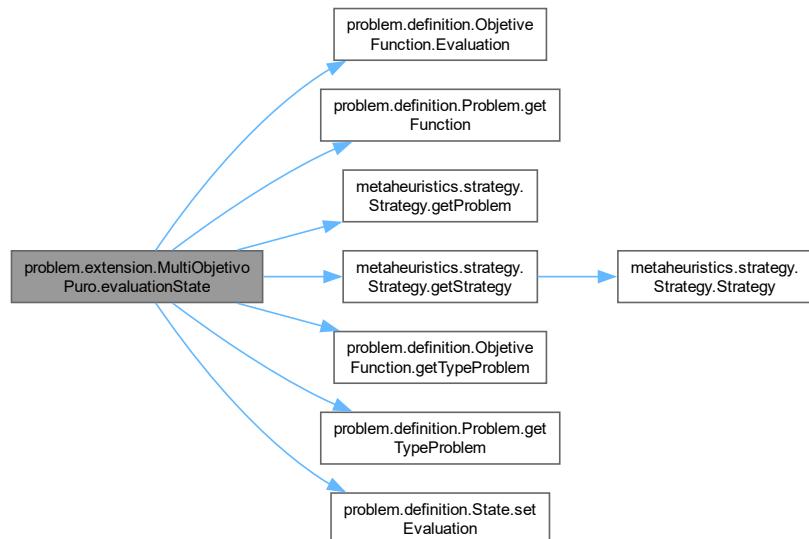
6.66.1. Documentación de funciones miembro

6.66.1.1. [evaluationState\(\)](#)

```
void problem.extension.MultiObjetivoPuro.evaluationState (
    State state)
```

Reimplementado de [problem.extension.SolutionMethod](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/extension/MultiObjetivoPuro.java](#)

6.67. Referencia de la clase

evolutionary_algorithms.complement.Mutation

Diagrama de herencia de `evolutionary_algorithms.complement.Mutation`

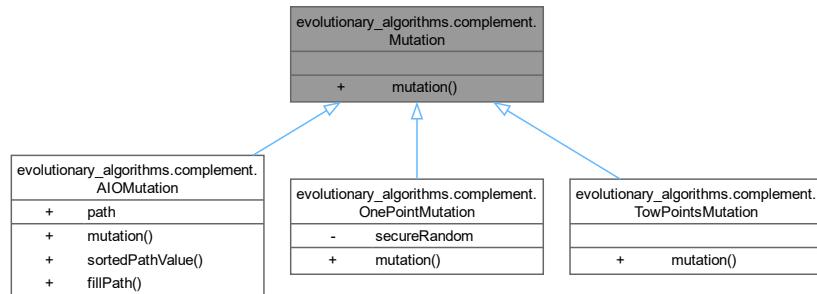
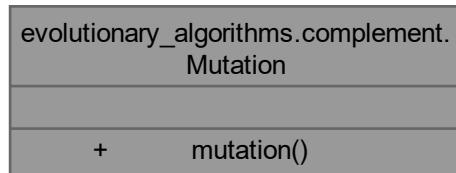


Diagrama de colaboración de `evolutionary_algorithms.complement.Mutation`:



Métodos públicos

- abstract `State mutation (State state, double PM)`

6.67.1. Documentación de funciones miembro

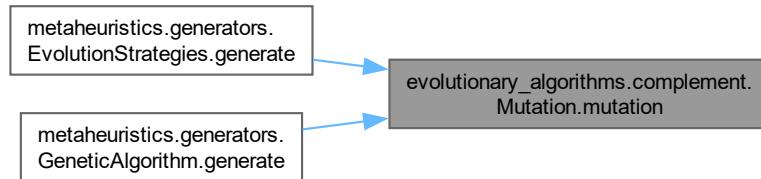
6.67.1.1. `mutation()`

```

abstract State evolutionary_algorithms.complement.Mutation.mutation (
    State state,
    double PM) [abstract]
  
```

Reimplementado en `evolutionary_algorithms.complement.AIOMutation`, `evolutionary_algorithms.complement.OnePointMutation` y `evolutionary_algorithms.complement.TowPointsMutation`.

Gráfico de llamadas a esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/evolutionary_algorithms/complement/Mutation.java

6.68. Referencia de la clase problem_operators.MutationOperator

Diagrama de herencia de problem_operators.MutationOperator

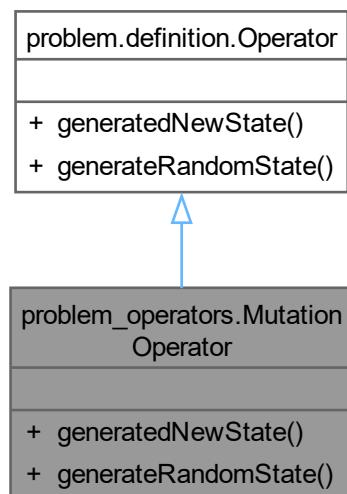
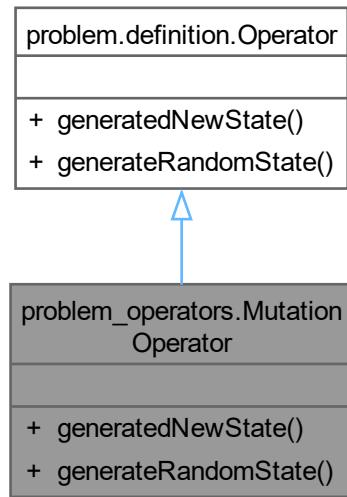


Diagrama de colaboración de problem_operators.MutationOperator:



Métodos públicos

- List< [State](#) > [generatedNewState](#) ([State](#) stateCurrent, Integer operatornumber)
- List< [State](#) > [generateRandomState](#) (Integer operatornumber)

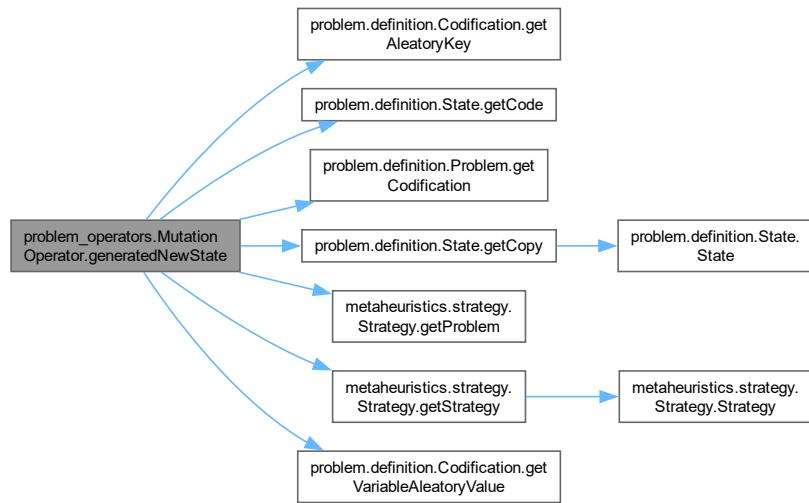
6.68.1. Documentación de funciones miembro

6.68.1.1. [generatedNewState\(\)](#)

```
List< State > problem_operators.MutationOperator.generatedNewState (
    State stateCurrent,
    Integer operatornumber)
```

Reimplementado de [problem.definition.Operator](#).

Gráfico de llamadas de esta función:



6.68.1.2. generateRandomState()

```
List< State > problem_operators.MutationOperator.generateRandomState (
    Integer operatornumber)
```

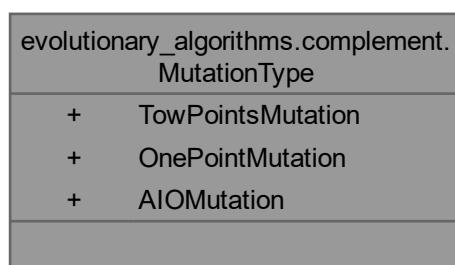
Reimplementado de `problem.definition.Operator`.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem_operators/MutationOperator.java](#)

6.69. Referencia de la enumeración evolutionary_algorithms.complement.MutationType

Diagrama de colaboración de `evolutionary_algorithms.complement.MutationType`:



Atributos públicos

- [TowPointsMutation](#)
- [OnePointMutation](#)
- [AIOMutation](#)

6.69.1. Documentación de datos miembro

6.69.1.1. AIOMutation

```
evolutionary_algorithms.complement.MutationType.AIOMutation
```

6.69.1.2. OnePointMutation

```
evolutionary_algorithms.complement.MutationType.OnePointMutation
```

6.69.1.3. TowPointsMutation

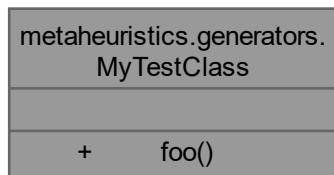
```
evolutionary_algorithms.complement.MutationType.TowPointsMutation
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/MutationType.java](#)

6.70. Referencia de la clase metaheuristics.generators.MyTestClass

Diagrama de colaboración de `metaheuristics.generators.MyTestClass`:



Métodos públicos

- [int foo \(\)](#)

6.70.1. Documentación de funciones miembro

6.70.1.1. foo()

```
int metaheuristics.generators.MyTestClass.foo ()
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/MyTestClass.java](#)

6.71. Referencia de la clase

local_search.candidate_type.NotDominatedCandidate

Diagrama de herencia de local_search.candidate_type.NotDominatedCandidate

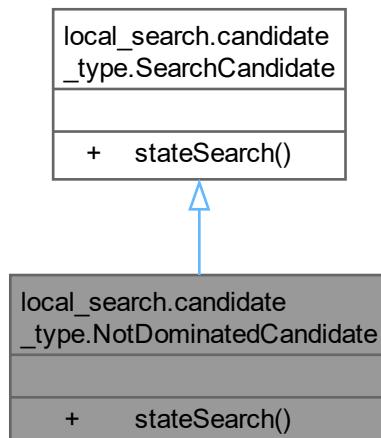
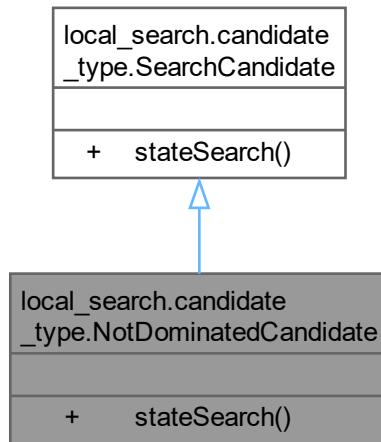


Diagrama de colaboración de local_search.candidate_type.NotDominatedCandidate:



Métodos públicos

- **State stateSearch (List< State > listNeighborhood)** throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.71.1. Documentación de funciones miembro

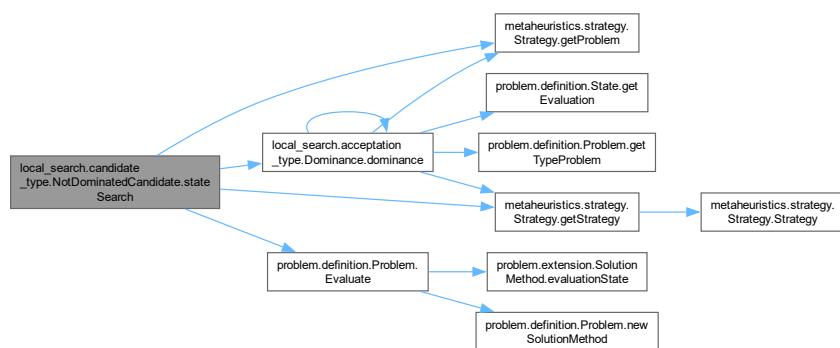
6.71.1.1. stateSearch()

```

State local_search.candidate_type.NotDominatedCandidate.stateSearch (
    List< State > listNeighborhood) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
  
```

Reimplementado de [local_search.candidate_type.SearchCandidate](#).

Gráfico de llamadas de esta función:

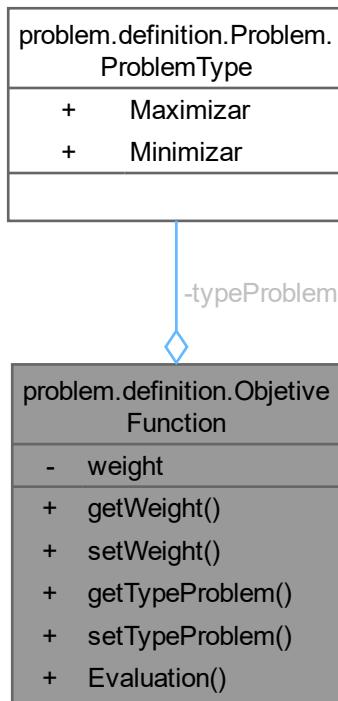


La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/candidate_type/NotDominatedCandidate.java](#)

6.72. Referencia de la clase problem.definition.ObjectiveFunction

Diagrama de colaboración de problem.definition.ObjectiveFunction:



Métodos públicos

- `float getWeight ()`
- `void setWeight (float weight)`
- `ProblemType getTypeProblem ()`
- `void setTypeProblem (ProblemType typeProblem)`
- abstract Double `Evaluation (State state)`

Atributos privados

- `ProblemType typeProblem`
- `float weight`

6.72.1. Documentación de funciones miembro

6.72.1.1. Evaluation()

```
abstract Double problem.definition.ObjectiveFunction.Evaluation (
    State state) [abstract]
```

Gráfico de llamadas a esta función:



6.72.1.2. getTypeProblem()

```
ProblemType problem.definition.ObjectiveFunction.getTypeProblem ()
```

Gráfico de llamadas a esta función:



6.72.1.3. getWeight()

```
float problem.definition.ObjectiveFunction.getWeight ()
```

6.72.1.4. setTypeProblem()

```
void problem.definition.ObjectiveFunction.setTypeProblem (
    ProblemType typeProblem)
```

6.72.1.5. setWeight()

```
void problem.definition.ObjectiveFunction.setWeight (
    float weight)
```

6.72.2. Documentación de datos miembro

6.72.2.1. typeProblem

```
ProblemType problem.definition.ObjectiveFunction.typeProblem [private]
```

6.72.2.2. weight

```
float problem.definition.ObjectiveFunction.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/definition/ObjectiveFunction.java](#)

6.73. Referencia de la clase evolutionary_algorithms.complement.OnePointCrossover

Diagrama de herencia de evolutionary_algorithms.complement.OnePointCrossover

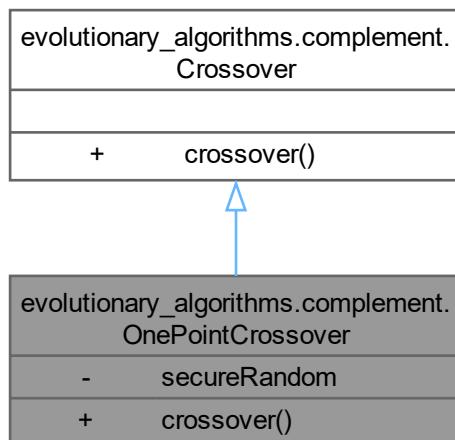
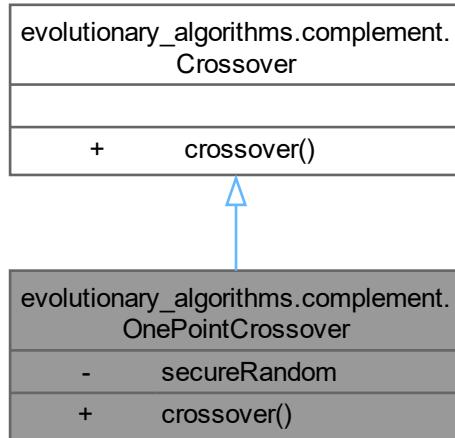


Diagrama de colaboración de `evolutionary_algorithms.complement.OnePointCrossover`:



Métodos públicos

- `State crossover (State father1, State father2, double PC)`

Atributos estáticos privados

- `static final SecureRandom secureRandom = new SecureRandom()`

6.73.1. Documentación de funciones miembro

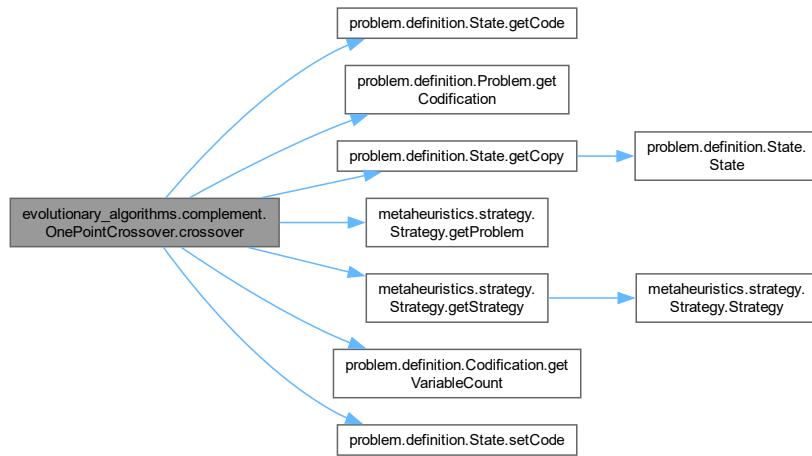
6.73.1.1. `crossover()`

```

State evolutionary_algorithms.complement.OnePointCrossover.crossover (
    State father1,
    State father2,
    double PC)
  
```

Reimplementado de `evolutionary_algorithms.complement.Crossover`.

Gráfico de llamadas de esta función:



6.73.2. Documentación de datos miembro

6.73.2.1. `secureRandom`

```
final SecureRandom evolutionary_algorithms.complement.OnePointCrossover.secureRandom = new  
SecureRandom() [static], [private]
```

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/evolutionary_algorithms/complement/[OnePointCrossover.java](#)

6.74. Referencia de la clase

evolutionary_algorithms.complement.OnePointMutation

Diagrama de herencia de evolutionary_algorithms.complement.OnePointMutation

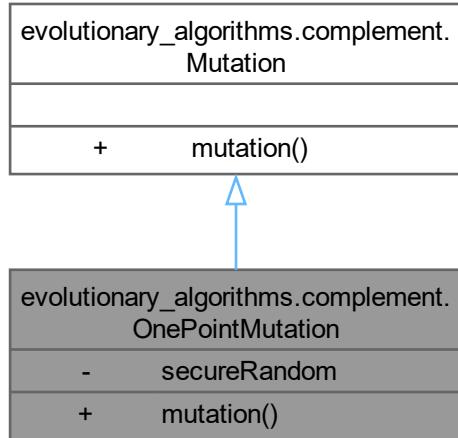
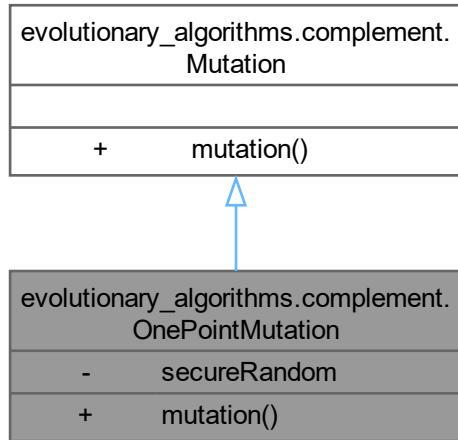


Diagrama de colaboración de `evolutionary_algorithms.complement.OnePointMutation`:



Métodos públicos

- [State mutation \(State state, double PM\)](#)

Atributos estáticos privados

- static final SecureRandom `secureRandom` = new SecureRandom()

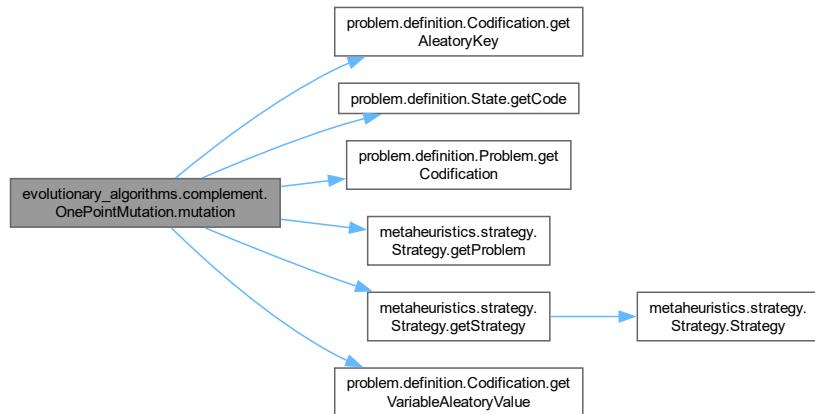
6.74.1. Documentación de funciones miembro

6.74.1.1. mutation()

```
State evolutionary_algorithms.complement.OnePointMutation.mutation (
    State state,
    double PM)
```

Reimplementado de [evolutionary_algorithms.complement.Mutation](#).

Gráfico de llamadas de esta función:



6.74.2. Documentación de datos miembro

6.74.2.1. secureRandom

```
final SecureRandom evolutionary_algorithms.complement.OnePointMutation.secureRandom = new
SecureRandom() [static], [private]
```

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/evolutionary_algorithms/complement/OnePointMutation.java`

6.75. Referencia de la clase problem.definition.Operator

Diagrama de herencia de problem.definition.Operator

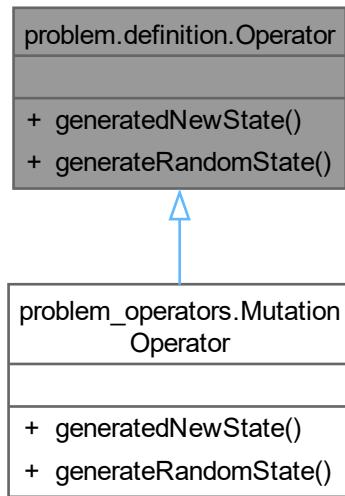
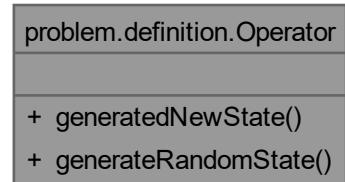


Diagrama de colaboración de problem.definition.Operator:



Métodos públicos

- abstract List<[State](#)> [generatedNewState](#) ([State stateCurrent](#), [Integer operatornumber](#))
- abstract List<[State](#)> [generateRandomState](#) ([Integer operatornumber](#))

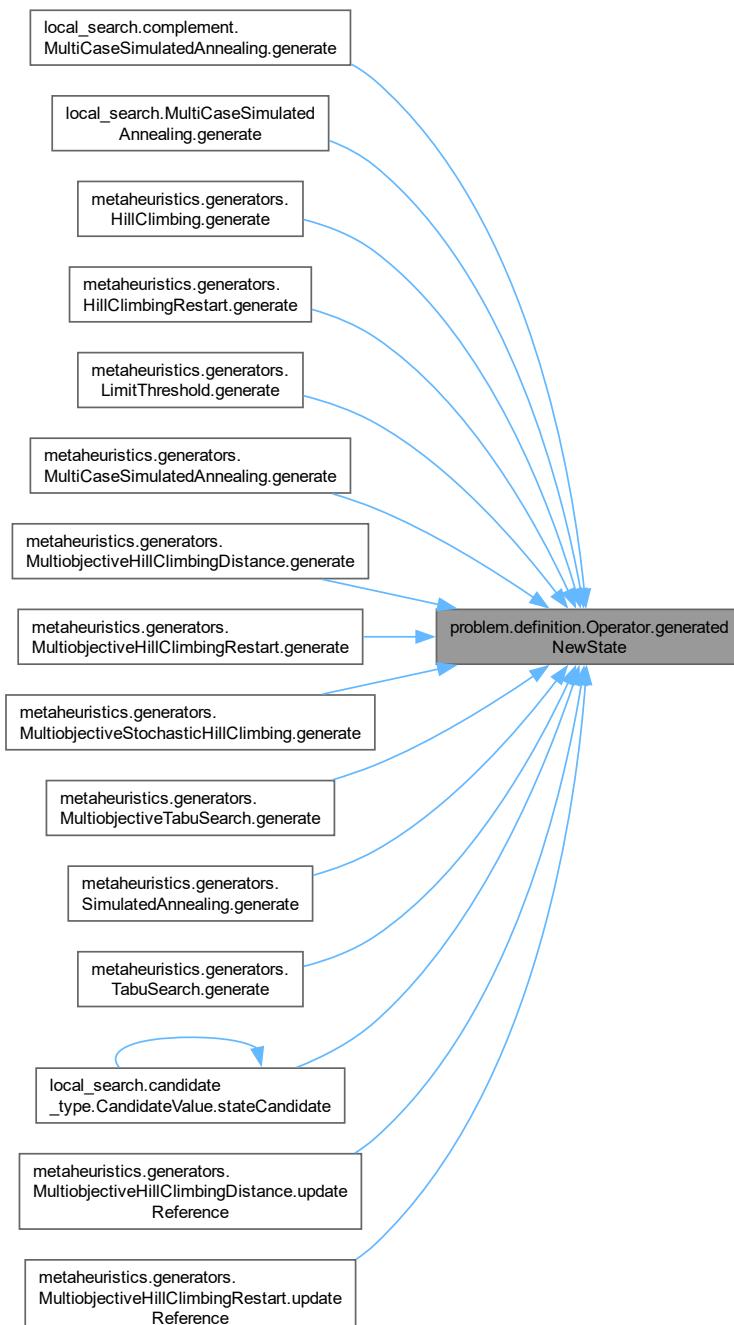
6.75.1. Documentación de funciones miembro

6.75.1.1. generatedNewState()

```
abstract List< State > problem.definition.Operator.generatedNewState (
    State stateCurrent,
    Integer operatornumber) [abstract]
```

Reimplementado en [problem_operators.MutationOperator](#).

Gráfico de llamadas a esta función:

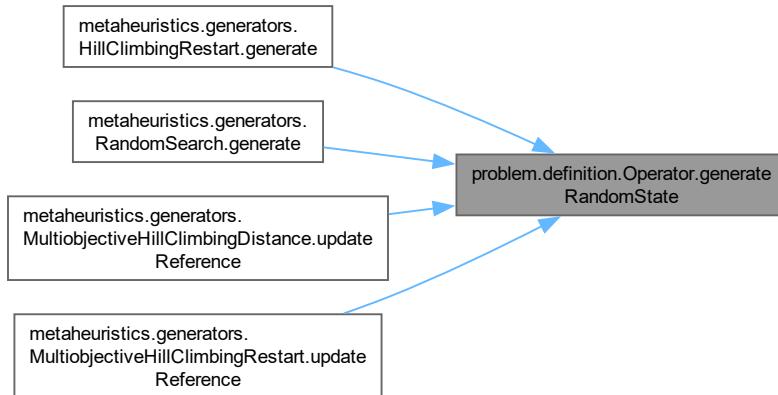


6.75.1.2. generateRandomState()

```
abstract List< State > problem.definition.Operator.generateRandomState (
    Integer operatornumber) [abstract]
```

Reimplementado en [problem_operators.MutationOperator](#).

Gráfico de llamadas a esta función:



La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/definition/Operator.java](#)

6.76. Referencia de la clase metaheuristics.generators.Particle

Representa una partícula en el algoritmo de Optimización por Enjambre de Partículas (PSO).

Diagrama de herencia de metaheuristics.generators.Particle

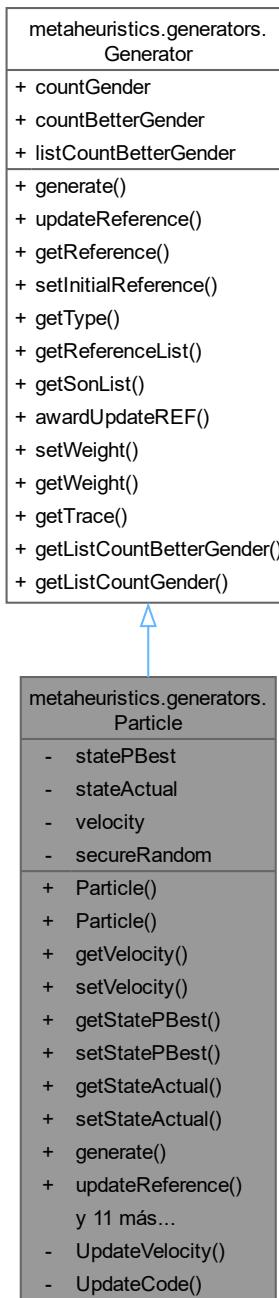
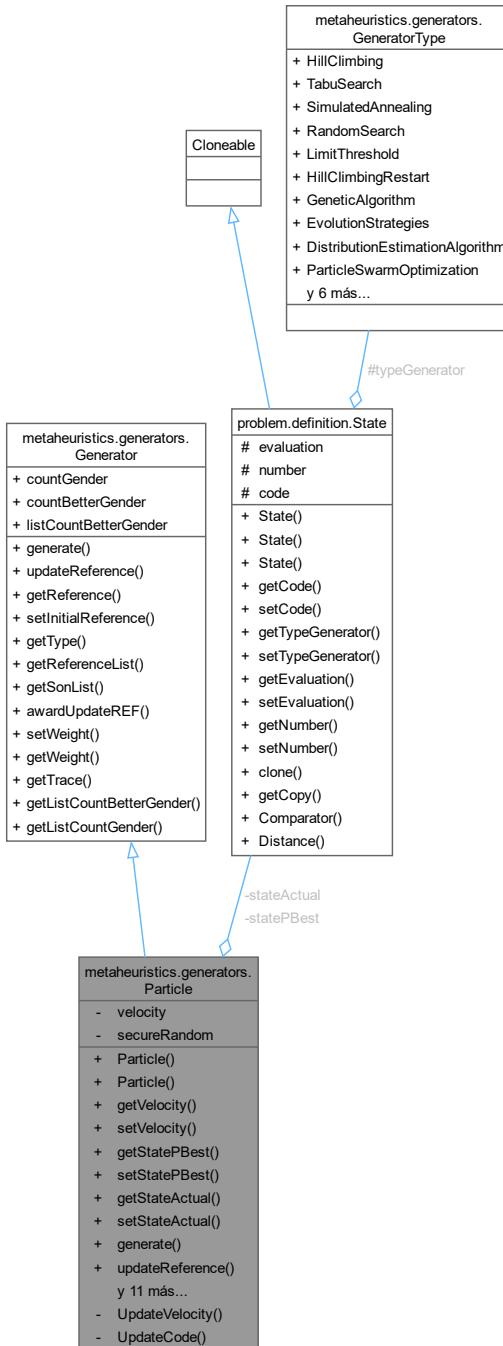


Diagrama de colaboración de metaheuristics.generators.Particle:



Métodos públicos

- **Particle ()**
Constructor por defecto.
- **Particle (State statePBest, State stateActual, ArrayList< Object > velocity)**
Constructor parametrizado.
- **ArrayList< Object > getVelocity ()**

- void [setVelocity](#) (ArrayList< Object > *velocity*)
- [State getStatePBest](#) ()
- void [setStatePBest](#) ([State statePBest](#))
- [State getStateActual](#) ()
- void [setStateActual](#) ([State stateActual](#))
- [State generate](#) (Integer *operatornumber*) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)
Genera un nuevo estado para la partícula.
- void [updateReference](#) ([State stateCandidate](#), Integer *countIterationsCurrent*) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)
- [State getReference](#) ()
- void [setInitialReference](#) ([State statelInitialRef](#))
- [GeneratorType getType](#) ()
- List< [State](#) > [getReferenceList](#) ()
- List< [State](#) > [getSonList](#) ()
- boolean [awardUpdateREF](#) ([State stateCandidate](#))
- void [setWeight](#) (float *weight*)
- float [getWeight](#) ()
- float[] [getTrace](#) ()
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()

Métodos privados

- ArrayList< Object > [UpdateVelocity](#) ()
Actualiza el vector de velocidad de la partícula.
- ArrayList< Object > [UpdateCode](#) (ArrayList< Object > *actualVelocity*)

Atributos privados

- [State statePBest](#)
Mejor estado alcanzado por esta partícula (pBest).
- [State stateActual](#)
Estado actual de la partícula.
- ArrayList< Object > [velocity](#)
Vector de velocidad actual de la partícula.

Atributos estáticos privados

- static final SecureRandom [secureRandom](#) = new SecureRandom()
Generador de números aleatorios seguro.

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.76.1. Descripción detallada

Representa una partícula en el algoritmo de Optimización por Enjambre de Partículas (PSO).

Cada partícula mantiene su estado actual, su mejor estado histórico (pBest) y su velocidad actual. Extiende de la clase [Generator](#) para integrarse en el framework de metaheurísticas.

6.76.2. Documentación de constructores y destructores

6.76.2.1. Particle() [1/2]

```
metaheuristics.generators.Particle.Particle ()
```

Constructor por defecto.

Inicializa el estado actual, pBest y velocidad vacíos.

6.76.2.2. Particle() [2/2]

```
metaheuristics.generators.Particle.Particle (
    State statePBest,
    State stateActual,
    ArrayList< Object > velocity)
```

Constructor parametrizado.

Parámetros

<code>statePBest</code>	Mejor estado histórico inicial.
<code>stateActual</code>	Estado actual inicial.
<code>velocity</code>	Velocidad inicial.

6.76.3. Documentación de funciones miembro

6.76.3.1. awardUpdateREF()

```
boolean metaheuristics.generators.Particle.awardUpdateREF (
    State stateCandidate)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.2. generate()

```
State metaheuristics.generators.Particle.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Genera un nuevo estado para la partícula.

Actualiza la velocidad y la posición (código) de la partícula basándose en las ecuaciones de PSO.

Parámetros

<code>operatornumber</code>	Identificador del operador (no utilizado en esta implementación específica).
-----------------------------	--

Devuelve

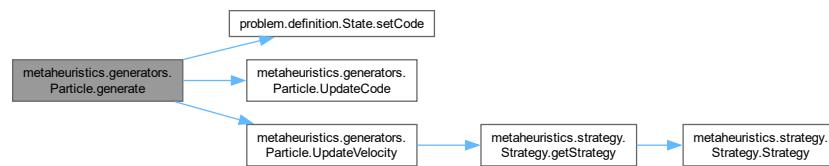
`null` (El estado se actualiza internamente en `stateActual`).

Excepciones

<code>IllegalArgumentException</code>	Si ocurre un error en la generación.
<code>SecurityException</code>	Si hay restricciones de seguridad.
<code>ClassNotFoundException</code>	Si no se encuentra una clase necesaria.
<code>InstantiationException</code>	Si falla la instancia.
<code>IllegalAccessException</code>	Si hay acceso ilegal.
<code>InvocationTargetException</code>	Si falla la invocación de un método.
<code>NoSuchMethodException</code>	Si no se encuentra un método.

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:

**6.76.3.3. getListCountBetterGender()**

```
int[] metaheuristics.generators.Particle.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.4. getListCountGender()

```
int[] metaheuristics.generators.Particle.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.5. getReference()

```
State metaheuristics.generators.Particle.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.6. `getReferenceList()`

```
List< State > metaheuristics.generators.Particle.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.7. `getSonList()`

```
List< State > metaheuristics.generators.Particle.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.8. `getStateActual()`

```
State metaheuristics.generators.Particle.getStateActual ()
```

6.76.3.9. `getStatePBest()`

```
State metaheuristics.generators.Particle.getStatePBest ()
```

Gráfico de llamadas a esta función:



6.76.3.10. `getTrace()`

```
float[] metaheuristics.generators.Particle.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.11. `getType()`

```
GeneratorType metaheuristics.generators.Particle.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.12. `getVelocity()`

```
ArrayList< Object > metaheuristics.generators.Particle.getVelocity ()
```

6.76.3.13. `getWeight()`

```
float metaheuristics.generators.Particle.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.14. `setInitialReference()`

```
void metaheuristics.generators.Particle.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.15. `setStateActual()`

```
void metaheuristics.generators.Particle.setStateActual (
    State stateActual)
```

6.76.3.16. `setStatePBest()`

```
void metaheuristics.generators.Particle.setStatePBest (
    State statePBest)
```

6.76.3.17. `setVelocity()`

```
void metaheuristics.generators.Particle.setVelocity (
    ArrayList< Object > velocity)
```

6.76.3.18. `setWeight()`

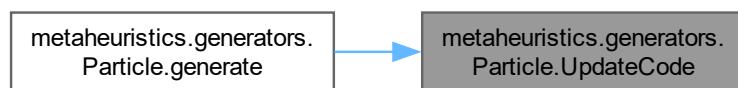
```
void metaheuristics.generators.Particle.setWeight (
    float weight)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.76.3.19. `UpdateCode()`

```
ArrayList< Object > metaheuristics.generators.Particle.UpdateCode (
    ArrayList< Object > actualVelocity) [private]
```

Gráfico de llamadas a esta función:



6.76.3.20. updateReference()

```
void metaheuristics.generators.Particle.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:

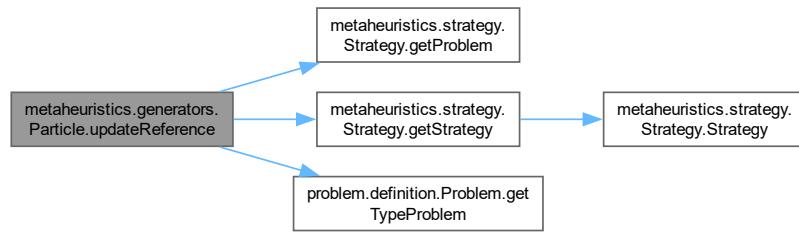
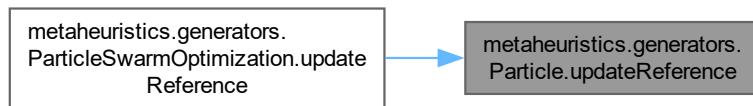


Gráfico de llamadas a esta función:



6.76.3.21. UpdateVelocity()

```
ArrayList< Object > metaheuristics.generators.Particle.UpdateVelocity () [private]
```

Actualiza el vector de velocidad de la partícula.

Calcula la nueva velocidad basándose en la inercia, el componente cognitivo (`pBest`) y el componente social (`lBest`). Aplica un factor de restricción para controlar la convergencia.

Devuelve

ArrayList<Object> Nueva velocidad calculada.

Gráfico de llamadas de esta función:

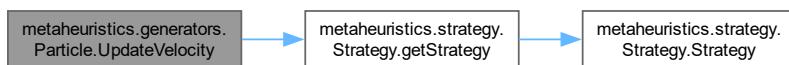
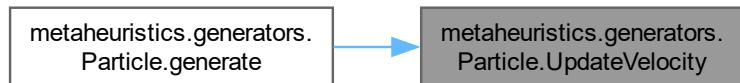


Gráfico de llamadas a esta función:



6.76.4. Documentación de datos miembro

6.76.4.1. secureRandom

```
final SecureRandom metaheuristics.generators.Particle.secureRandom = new SecureRandom() [static],  
[private]
```

Generador de números aleatorios seguro.

6.76.4.2. stateActual

```
State metaheuristics.generators.Particle.stateActual [private]
```

Estado actual de la partícula.

6.76.4.3. statePBest

```
State metaheuristics.generators.Particle.statePBest [private]
```

Mejor estado alcanzado por esta partícula (pBest).

6.76.4.4. velocity

```
ArrayList<Object> metaheuristics.generators.Particle.velocity [private]
```

Vector de velocidad actual de la partícula.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/Particle.java](#)

6.77. Referencia de la clase

metaheuristics.generators.ParticleSwarmOptimization

Implementación del algoritmo de Optimización por Enjambre de Partículas (PSO).

Diagrama de herencia de metaheuristics.generators.ParticleSwarmOptimization

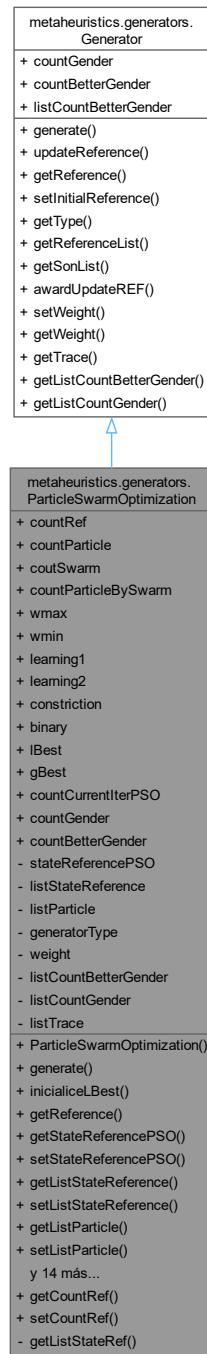
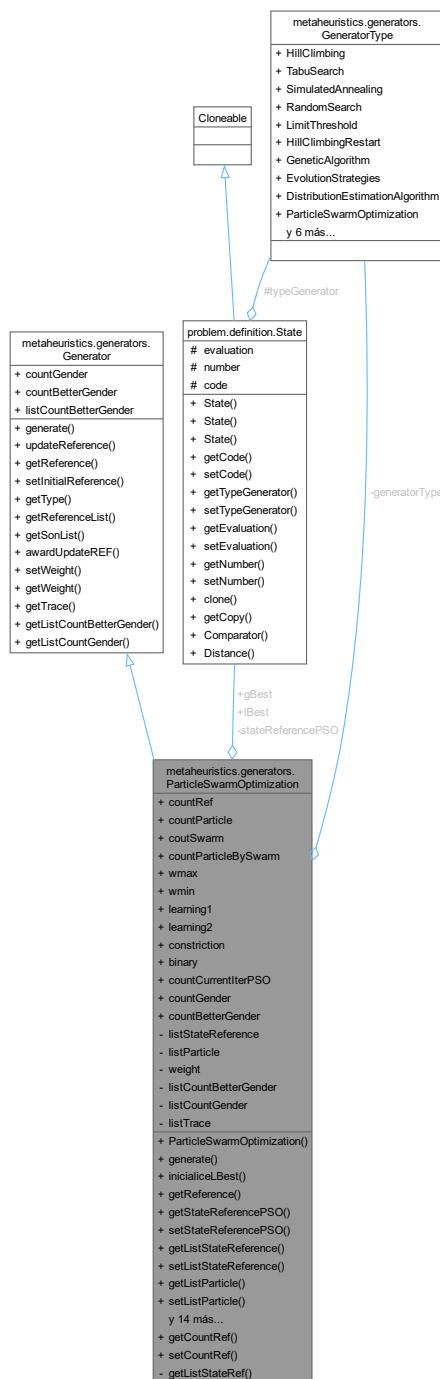


Diagrama de colaboración de metaheuristics.generators.ParticleSwarmOptimization:



Métodos públicos

■ `ParticleSwarmOptimization ()`

Constructor por defecto.

- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

- void **inicialiceLBest** ()
- State **getReference** ()
- State **getStateReferencePSO** ()
- void **setStateReferencePSO** (State stateReferencePSO)
- List< Particle > **getListStateReference** ()
- void **setListStateReference** (List< State > listStateReference)
- List< Particle > **getListParticle** ()
- List< Particle > **setListParticle** (List< Particle > listParticle)
- GeneratorType **getGeneratorType** ()
- void **setGeneratorType** (GeneratorType generatorType)
- void **updateReference** (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- State **gBestInicial** ()
- void **setInitialReference** (State stateInitialRef)
- GeneratorType **getType** ()
- List< State > **getReferenceList** ()
- List< State > **getSonList** ()
- boolean **awardUpdateREF** (State stateCandidate)
- void **setWeight** (float weight)
- float **getWeight** ()
- int[] **getListCountBetterGender** ()
- int[] **getListCountGender** ()
- float[] **getTrace** ()

Métodos públicos estáticos

- static int **getCountRef** ()
- static void **setCountRef** (int countRef)

Atributos públicos estáticos

- static int **countRef** = 0

*Cantidad total de partículas (coutSwarm * countParticleSwarm).*
- static int **countParticle** = 0

Cantidad de partículas que se han movido en el cúmulo actual.
- static int **coutSwarm** = 0

Cantidad de cúmulos (swarms).
- static int **countParticleBySwarm** = 0

Cantidad de partículas por cúmulo.
- static double **wmax** = 0.9

Peso de inercia máximo.
- static double **wmin** = 0.2

Peso de inercia mínimo.
- static int **learning1** = 2

Factores de aprendizaje cognitivo y social.
- static int **learning2** = 2
- static double **constriction**

Factor de restricción calculado dinámicamente.
- static boolean **binary** = false

Indica si el problema es binario.
- static State[] **lBest**

Array de mejores estados locales por cúmulo.

- static State gBest

Mejor estado global encontrado.

- static int countCurrentIterPSO

Contador de iteraciones actual del PSO.

- static int countGender = 0

- static int countBetterGender = 0

Métodos privados

- List< Particle > getListStateRef ()

Atributos privados

- State stateReferencePSO
- List< State > listStateReference = new ArrayList<State>()
- List< Particle > listParticle = new ArrayList<Particle> ()
- GeneratorType generatorType
- float weight = 50
- int[] listCountBetterGender = new int[10]
- int[] listCountGender = new int[10]
- float[] listTrace = new float[1200000]

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int countGender
- int countBetterGender
- int[] listCountBetterGender

6.77.1. Descripción detallada

Implementación del algoritmo de Optimización por Enjambre de Partículas (PSO).

Esta clase gestiona el enjambre de partículas, sus movimientos, y la actualización de los mejores estados locales (lBest) y globales (gBest).

6.77.2. Documentación de constructores y destructores

6.77.2.1. ParticleSwarmOptimization()

```
metaheuristics.generators.ParticleSwarmOptimization.ParticleSwarmOptimization ()
```

Constructor por defecto.

Inicializa el enjambre y configura los parámetros iniciales. Gráfico de llamadas de esta función:

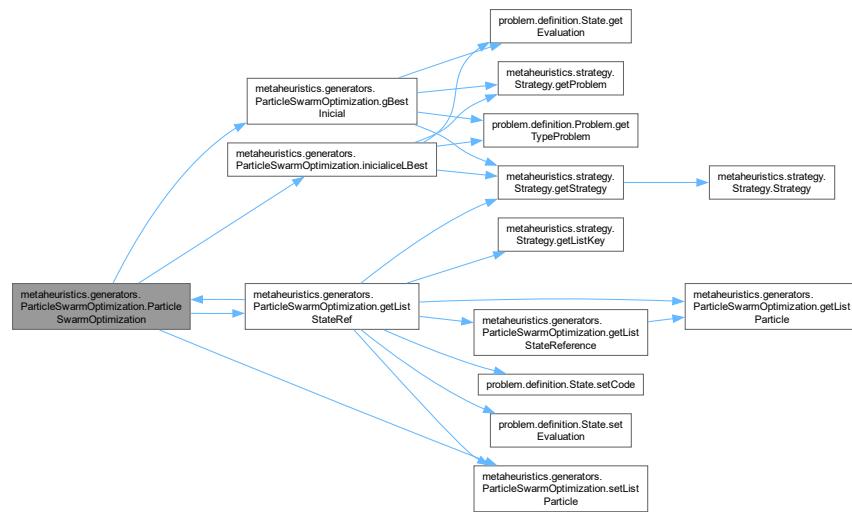
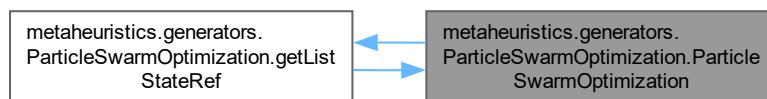


Gráfico de llamadas a esta función:



6.77.3. Documentación de funciones miembro

6.77.3.1. awardUpdateREF()

```
boolean metaheuristics.generators.ParticleSwarmOptimization.awardUpdateREF (
    State stateCandidate)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.2. gBestIncial()

```
State metaheuristics.generators.ParticleSwarmOptimization.gBestIncial ()
```

Gráfico de llamadas de esta función:

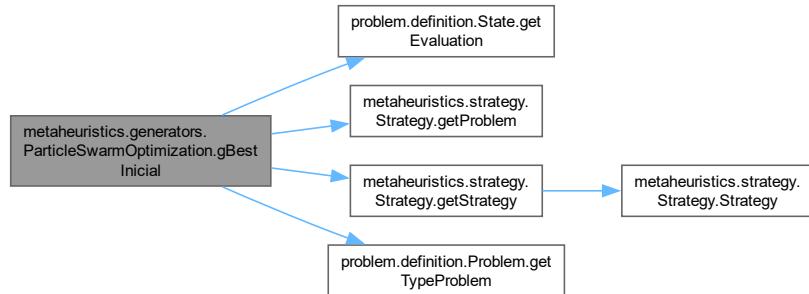
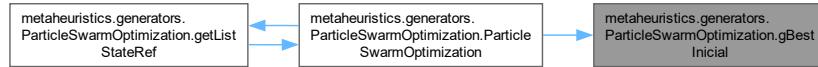


Gráfico de llamadas a esta función:



6.77.3.3. generate()

```
State metaheuristics.generators.ParticleSwarmOptimization.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class<→
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de `metaheuristics.generators.Generator`.

6.77.3.4. getCountRef()

```
int metaheuristics.generators.ParticleSwarmOptimization.getCountRef () [static]
```

6.77.3.5. getGeneratorType()

```
GeneratorType metaheuristics.generators.ParticleSwarmOptimization.getGeneratorType ()
```

6.77.3.6. getListCountBetterGender()

```
int[] metaheuristics.generators.ParticleSwarmOptimization.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.7. getListCountGender()

```
int[] metaheuristics.generators.ParticleSwarmOptimization.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.8. getListParticle()

```
List< Particle > metaheuristics.generators.ParticleSwarmOptimization.getListParticle ()
```

Gráfico de llamadas a esta función:



6.77.3.9. getListStateRef()

```
List< Particle > metaheuristics.generators.ParticleSwarmOptimization.getListStateRef () [private]
```

Gráfico de llamadas de esta función:

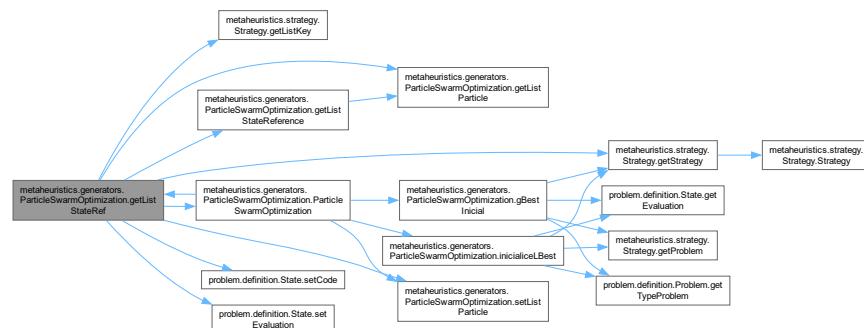


Gráfico de llamadas a esta función:



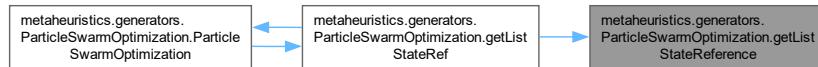
6.77.3.10. `getListStateReference()`

```
List< Particle > metaheuristics.generators.ParticleSwarmOptimization.getListStateReference ()
```

Gráfico de llamadas de esta función:



Gráfico de llamadas a esta función:



6.77.3.11. `getReference()`

```
State metaheuristics.generators.ParticleSwarmOptimization.getReference ()
```

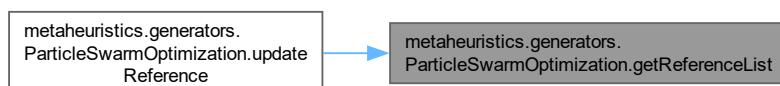
Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.12. `getReferenceList()`

```
List< State > metaheuristics.generators.ParticleSwarmOptimization.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas a esta función:



6.77.3.13. `getSonList()`

```
List< State > metaheuristics.generators.ParticleSwarmOptimization.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.14. `getStateReferencePSO()`

```
State metaheuristics.generators.ParticleSwarmOptimization.getStateReferencePSO ()
```

6.77.3.15. `getTrace()`

```
float[ ] metaheuristics.generators.ParticleSwarmOptimization.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.16. `getType()`

```
GeneratorType metaheuristics.generators.ParticleSwarmOptimization.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.17. `getWeight()`

```
float metaheuristics.generators.ParticleSwarmOptimization.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.18. `inicialiceLBest()`

```
void metaheuristics.generators.ParticleSwarmOptimization.inicialiceLBest ()
```

Gráfico de llamadas de esta función:

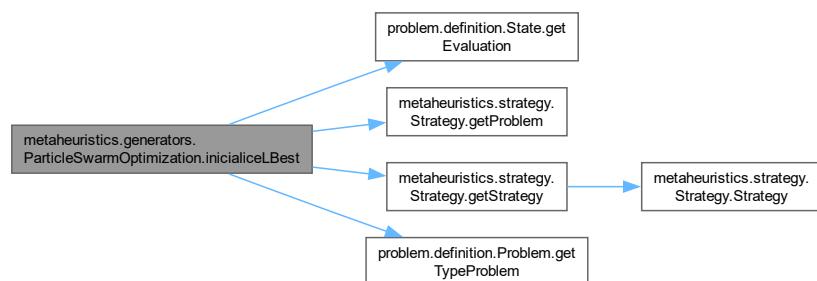
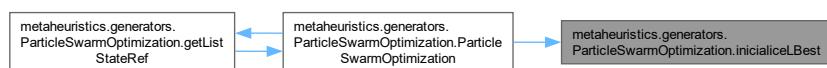


Gráfico de llamadas a esta función:



6.77.3.19. `setCountRef()`

```
void metaheuristics.generators.ParticleSwarmOptimization.setCountRef (
    int countRef) [static]
```

6.77.3.20. `setGeneratorType()`

```
void metaheuristics.generators.ParticleSwarmOptimization.setGeneratorType (
    GeneratorType generatorType)
```

6.77.3.21. `setInitialReference()`

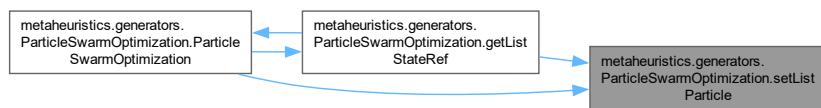
```
void metaheuristics.generators.ParticleSwarmOptimization.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.22. `setListParticle()`

```
List< Particle > metaheuristics.generators.ParticleSwarmOptimization.setListParticle (
    List< Particle > listParticle)
```

Gráfico de llamadas a esta función:



6.77.3.23. `setListStateReference()`

```
void metaheuristics.generators.ParticleSwarmOptimization.setListStateReference (
    List< State > listStateReference)
```

6.77.3.24. `setStateReferencePSO()`

```
void metaheuristics.generators.ParticleSwarmOptimization.setStateReferencePSO (
    State stateReferencePSO)
```

6.77.3.25. `setWeight()`

```
void metaheuristics.generators.ParticleSwarmOptimization.setWeight (
    float weight)
```

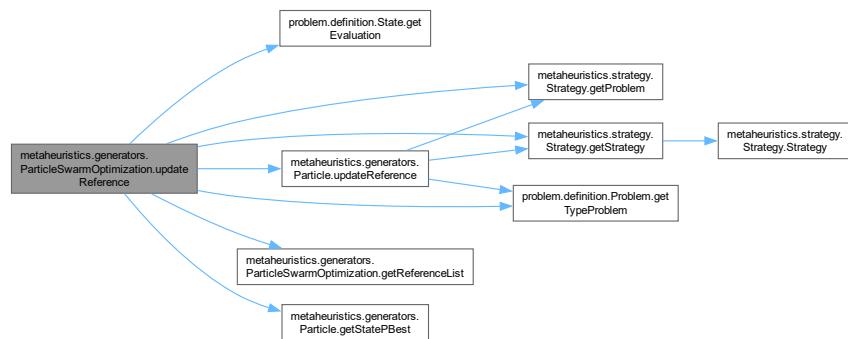
Reimplementado de [metaheuristics.generators.Generator](#).

6.77.3.26. updateReference()

```
void metaheuristics.generators.ParticleSwarmOptimization.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.77.4. Documentación de datos miembro

6.77.4.1. binary

```
boolean metaheuristics.generators.ParticleSwarmOptimization.binary = false [static]
```

Indica si el problema es binario.

6.77.4.2. constriction

```
double metaheuristics.generators.ParticleSwarmOptimization.constriction [static]
```

Factor de restricción calculado dinámicamente.

6.77.4.3. countBetterGender

```
int metaheuristics.generators.ParticleSwarmOptimization.countBetterGender = 0 [static]
```

6.77.4.4. countCurrentIterPSO

```
int metaheuristics.generators.ParticleSwarmOptimization.countCurrentIterPSO [static]
```

Contador de iteraciones actual del PSO.

6.77.4.5. countGender

```
int metaheuristics.generators.ParticleSwarmOptimization.countGender = 0 [static]
```

6.77.4.6. countParticle

```
int metaheuristics.generators.ParticleSwarmOptimization.countParticle = 0 [static]
```

Cantidad de partículas que se han movido en el cúmulo actual.

6.77.4.7. countParticleBySwarm

```
int metaheuristics.generators.ParticleSwarmOptimization.countParticleBySwarm = 0 [static]
```

Cantidad de partículas por cúmulo.

6.77.4.8. countRef

```
int metaheuristics.generators.ParticleSwarmOptimization.countRef = 0 [static]
```

Cantidad total de partículas (coutSwarm * countParticleSwarm).

6.77.4.9. coutSwarm

```
int metaheuristics.generators.ParticleSwarmOptimization.coutSwarm = 0 [static]
```

Cantidad de cúmulos (swarms).

6.77.4.10. gBest

```
State metaheuristics.generators.ParticleSwarmOptimization.gBest [static]
```

Mejor estado global encontrado.

6.77.4.11. generatorType

```
GeneratorType metaheuristics.generators.ParticleSwarmOptimization.generatorType [private]
```

6.77.4.12. lBest

```
State [ ] metaheuristics.generators.ParticleSwarmOptimization.lBest [static]
```

Array de mejores estados locales por cúmulo.

6.77.4.13. learning1

```
int metaheuristics.generators.ParticleSwarmOptimization.learning1 = 2 [static]
```

Factores de aprendizaje cognitivo y social.

6.77.4.14. learning2

```
int metaheuristics.generators.ParticleSwarmOptimization.learning2 = 2 [static]
```

6.77.4.15. listCountBetterGender

```
int [] metaheuristics.generators.ParticleSwarmOptimization.listCountBetterGender = new int[10] [private]
```

6.77.4.16. listCountGender

```
int [] metaheuristics.generators.ParticleSwarmOptimization.listCountGender = new int[10] [private]
```

6.77.4.17. listParticle

```
List<Particle> metaheuristics.generators.ParticleSwarmOptimization.listParticle = new Array←  
List<Particle> () [private]
```

6.77.4.18. listStateReference

```
List<State> metaheuristics.generators.ParticleSwarmOptimization.listStateReference = new  
ArrayList<State>() [private]
```

6.77.4.19. listTrace

```
float [] metaheuristics.generators.ParticleSwarmOptimization.listTrace = new float[1200000]  
[private]
```

6.77.4.20. stateReferencePSO

```
State metaheuristics.generators.ParticleSwarmOptimization.stateReferencePSO [private]
```

6.77.4.21. weight

```
float metaheuristics.generators.ParticleSwarmOptimization.weight = 50 [private]
```

6.77.4.22. wmax

```
double metaheuristics.generators.ParticleSwarmOptimization.wmax = 0.9 [static]
```

Peso de inercia máximo.

6.77.4.23. wmin

```
double metaheuristics.generators.ParticleSwarmOptimization.wmin = 0.2 [static]
```

Peso de inercia mínimo.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/ParticleSwarmOptimization.java](#)

6.78. Referencia de la clase **evolutionary_algorithms.complement.ProbabilisticSampling**

Diagrama de herencia de evolutionary_algorithms.complement.ProbabilisticSampling

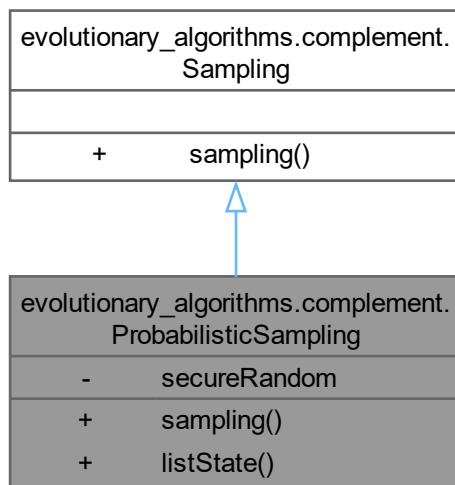
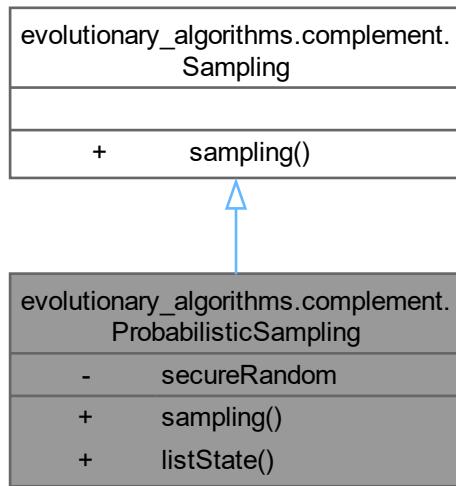


Diagrama de colaboración de evolutionary_algorithms.complement.ProbabilisticSampling:



Métodos públicos

- List< State > `sampling` (List< State > fathers, int countInd)
- List< State > `listState` (int countInd)

Atributos estáticos privados

- static final SecureRandom `secureRandom` = new SecureRandom()

6.78.1. Documentación de funciones miembro

6.78.1.1. `listState()`

```
List< State > evolutionary_algorithms.complement.ProbabilisticSampling.listState (
    int countInd)
```

Gráfico de llamadas de esta función:

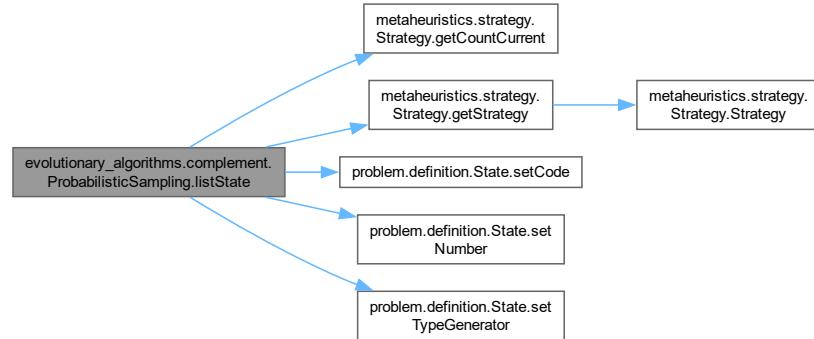


Gráfico de llamadas a esta función:

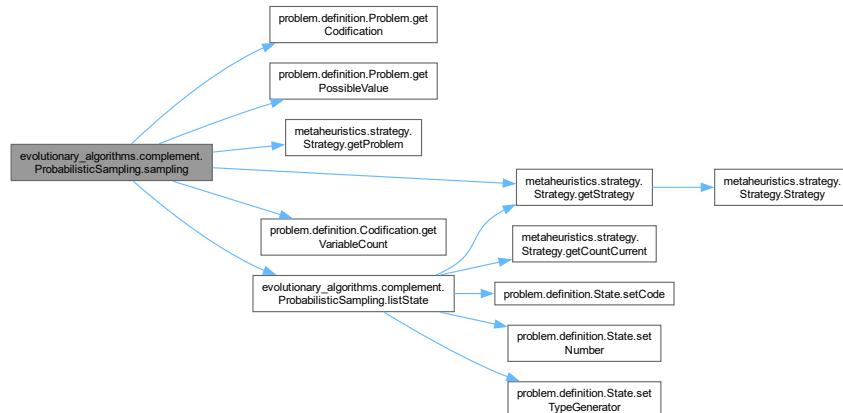


6.78.1.2. sampling()

```
List< State > evolutionary_algorithms.complement.ProbabilisticSampling.sampling (
    List< State > fathers,
    int countInd)
```

Reimplementado de [evolutionary_algorithms.complement.Sampling](#).

Gráfico de llamadas de esta función:



6.78.2. Documentación de datos miembro

6.78.2.1. secureRandom

```
final SecureRandom evolutionary_algorithms.complement.ProbabilisticSampling.secureRandom = new  
SecureRandom() [static], [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/ProbabilisticSampling.java](#)

6.79. Referencia de la clase evolutionary_algorithms.complement.Probability

Diagrama de colaboración de `evolutionary_algorithms.complement.Probability`:

evolutionary_algorithms.complement. Probability
- key
- value
- probability
+ Probability()
+ Probability()
+ getProbability()
+ setProbability()
+ getKey()
+ setKey()
+ getValue()
+ setValue()

Métodos públicos

- [Probability \(\)](#)
- [Probability \(Probability other\)](#)
- [float getProbability \(\)](#)
- [void setProbability \(float probability\)](#)
- [Object getKey \(\)](#)
- [void setKey \(Object key\)](#)
- [Object getValue \(\)](#)
- [void setValue \(Object value\)](#)

Atributos privados

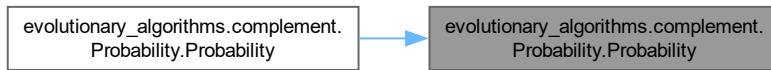
- Object `key`
- Object `value`
- float `probability`

6.79.1. Documentación de constructores y destructores

6.79.1.1. Probability() [1/2]

```
evolutionary_algorithms.complement.Probability.Probability ()
```

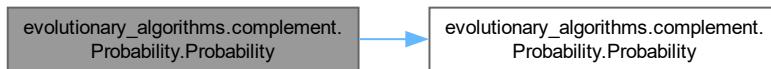
Gráfico de llamadas a esta función:



6.79.1.2. Probability() [2/2]

```
evolutionary_algorithms.complement.Probability.Probability (\n    Probability other)
```

Gráfico de llamadas de esta función:



6.79.2. Documentación de funciones miembro

6.79.2.1. getKey()

```
Object evolutionary_algorithms.complement.Probability.getKey ()
```

6.79.2.2. getProbability()

```
float evolutionary_algorithms.complement.Probability.getProbability ()
```

6.79.2.3. `getValue()`

```
Object evolutionary_algorithms.complement.Probability.getValue ()
```

6.79.2.4. `setKey()`

```
void evolutionary_algorithms.complement.Probability.setKey (
    Object key)
```

Gráfico de llamadas a esta función:



6.79.2.5. `setProbability()`

```
void evolutionary_algorithms.complement.Probability.setProbability (
    float probability)
```

Gráfico de llamadas a esta función:



6.79.2.6. `setValue()`

```
void evolutionary_algorithms.complement.Probability.setValue (
    Object value)
```

Gráfico de llamadas a esta función:



6.79.3. Documentación de datos miembro

6.79.3.1. key

```
Object evolutionary_algorithms.complement.Probability.key [private]
```

6.79.3.2. probability

```
float evolutionary_algorithms.complement.Probability.probability [private]
```

6.79.3.3. value

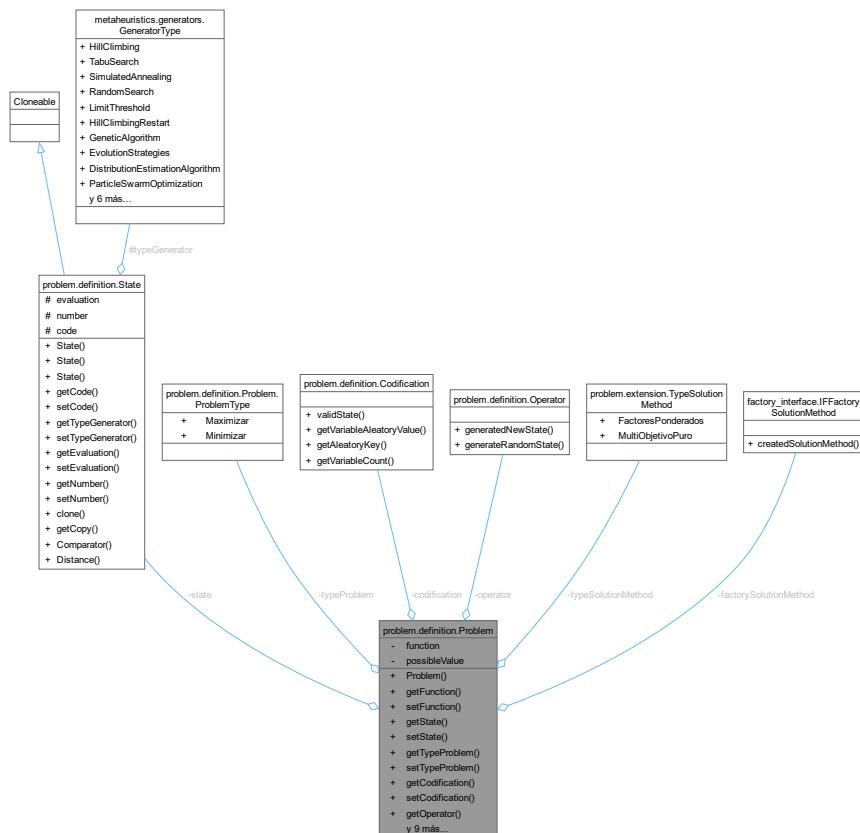
```
Object evolutionary_algorithms.complement.Probability.value [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/Probability.java](#)

6.80. Referencia de la clase problem.definition.Problem

Diagrama de colaboración de problem.definition.Problem:



Clases

- enum ProblemType

Métodos públicos

- Problem ()
- ArrayList< ObjetiveFunction > getFunction ()
- void setFunction (ArrayList< ObjetiveFunction > function)
- State getState ()
- void setState (State state)
- ProblemType getTypeProblem ()
- void setTypeProblem (ProblemType typeProblem)
- Codification getCodification ()
- void setCodification (Codification codification)
- Operator getOperator ()
- void setOperator (Operator operator)
- int getPossibleValue ()
- void setPossibleValue (int possibleValue)
- void Evaluate (State state) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- TypeSolutionMethod getTypeSolutionMethod ()
- void setTypeSolutionMethod (TypeSolutionMethod typeSolutionMethod)
- IFFactorySolutionMethod getFactorySolutionMethod ()
- void setFactorySolutionMethod (IFFactorySolutionMethod factorySolutionMethod)
- SolutionMethod newSolutionMethod (TypeSolutionMethod typeSolutionMethod) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

Atributos privados

- ArrayList< ObjetiveFunction > function
- State state
- ProblemType typeProblem
- Codification codification
- Operator operator
- int possibleValue
- TypeSolutionMethod typeSolutionMethod
- IFFactorySolutionMethod factorySolutionMethod

6.80.1. Documentación de constructores y destructores

6.80.1.1. Problem()

```
problem.definition.Problem.Problem ()
```

6.80.2. Documentación de funciones miembro

6.80.2.1. Evaluate()

```
void problem.definition.Problem.Evaluate (
    State state) throws IllegalArgumentException, SecurityException, ClassNotFoundException,
InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Gráfico de llamadas de esta función:

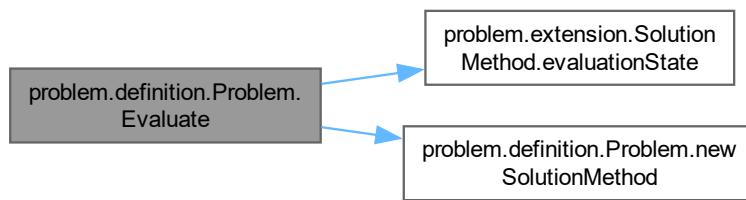
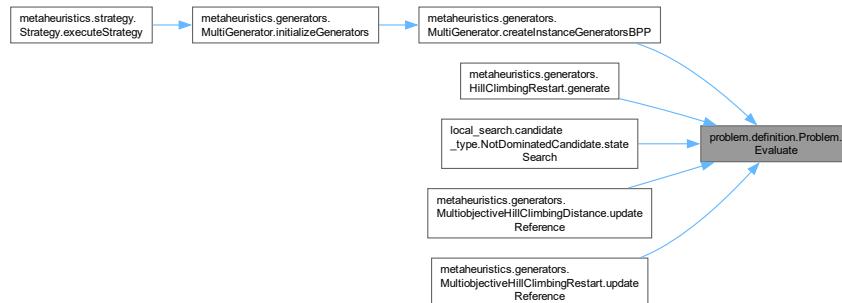


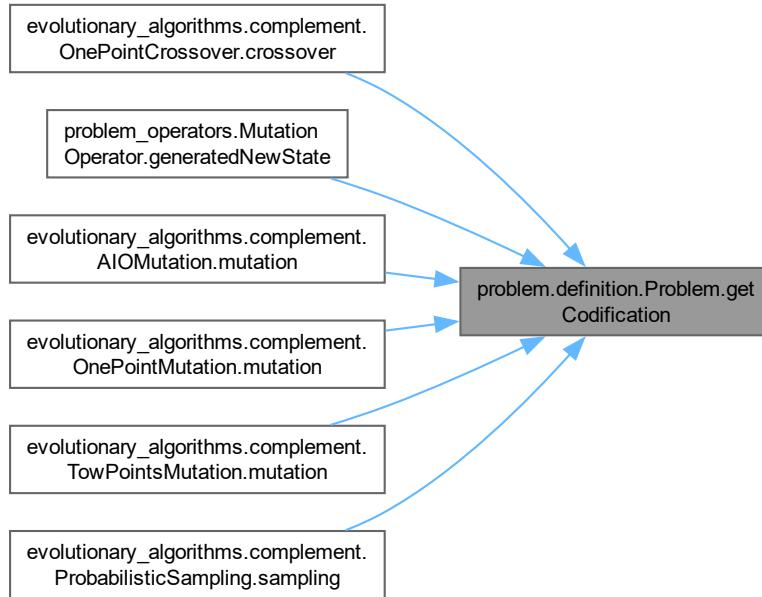
Gráfico de llamadas a esta función:



6.80.2.2. getCodification()

```
Codification problem.definition.Problem.getCodification ()
```

Gráfico de llamadas a esta función:



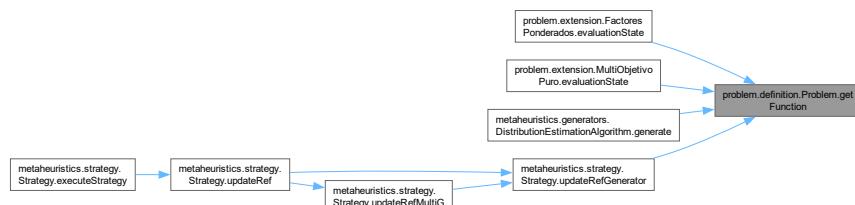
6.80.2.3. `getFactorySolutionMethod()`

```
IFFactorySolutionMethod problem.definition.Problem.getFactorySolutionMethod ()
```

6.80.2.4. `getFunction()`

```
ArrayList< ObjectiveFunction > problem.definition.Problem.getFunction ()
```

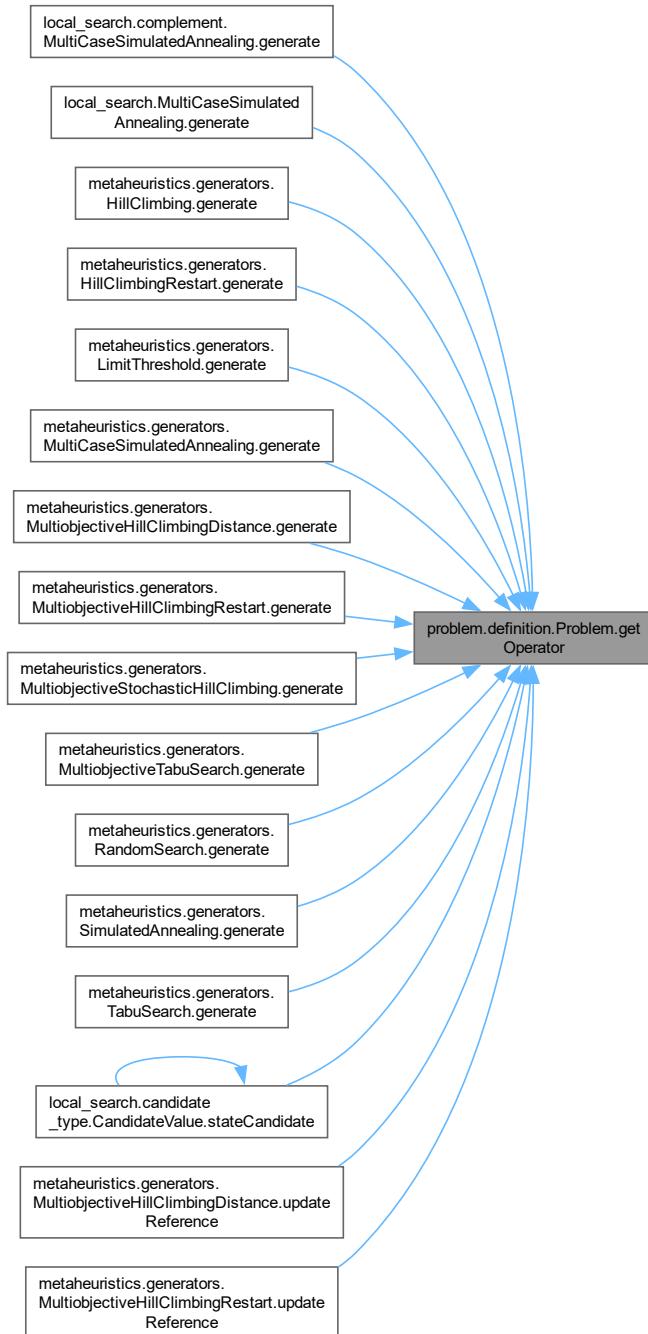
Gráfico de llamadas a esta función:



6.80.2.5. getOperator()

```
Operator problem.definition.Problem.getOperator ()
```

Gráfico de llamadas a esta función:



6.80.2.6. getPossibleValue()

```
int problem.definition.Problem.getPossibleValue ()
```

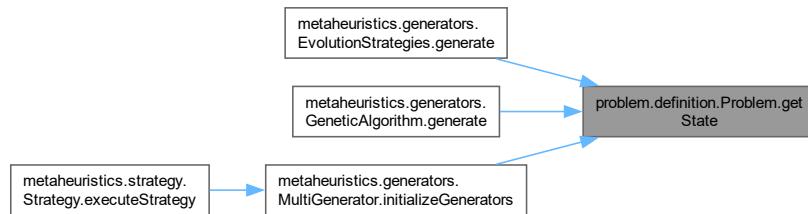
Gráfico de llamadas a esta función:



6.80.2.7. `getState()`

```
State problem.definition.Problem.getState ()
```

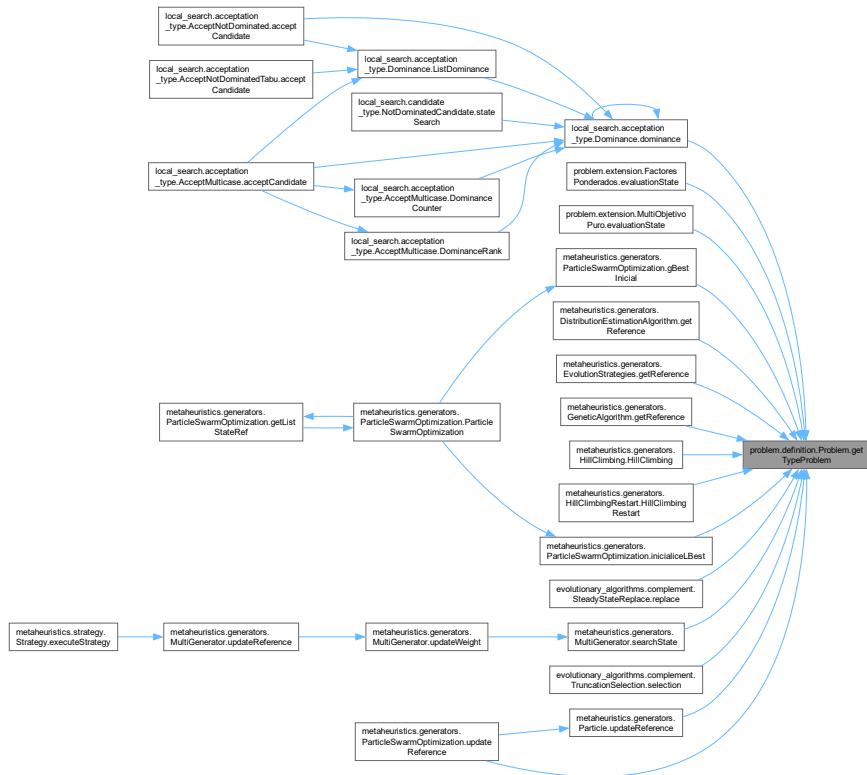
Gráfico de llamadas a esta función:



6.80.2.8. `getTypeProblem()`

```
ProblemType problem.definition.Problem.getTypeProblem ()
```

Gráfico de llamadas a esta función:



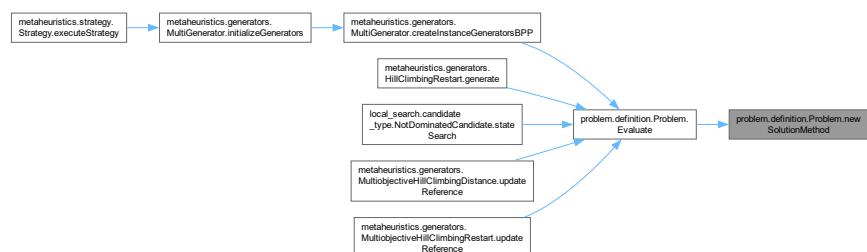
6.80.2.9. getTypeSolutionMethod()

```
TypeSolutionMethod problem.definition.Problem.getTypeSolutionMethod ()
```

6.80.2.10. newSolutionMethod()

```
SolutionMethod problem.definition.Problem.newSolutionMethod (
    TypeSolutionMethod typeSolutionMethod) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Gráfico de llamadas a esta función:



6.80.2.11. setCodification()

```
void problem.definition.Problem.setCodification (
    Codification codification)
```

6.80.2.12. setFactorySolutionMethod()

```
void problem.definition.Problem.setFactorySolutionMethod (
    IFFactorySolutionMethod factorySolutionMethod)
```

6.80.2.13. setFunction()

```
void problem.definition.Problem.setFunction (
    ArrayList< ObjetiveFunction > function)
```

6.80.2.14. setOperator()

```
void problem.definition.Problem.setOperator (
    Operator operator)
```

6.80.2.15. setPossibleValue()

```
void problem.definition.Problem.setPossibleValue (
    int possibleValue)
```

6.80.2.16. setState()

```
void problem.definition.Problem.setState (
    State state)
```

Gráfico de llamadas a esta función:

**6.80.2.17. setTypeProblem()**

```
void problem.definition.Problem.setTypeProblem (
    ProblemType typeProblem)
```

6.80.2.18. `setTypeSolutionMethod()`

```
void problem.definition.Problem.setTypeSolutionMethod (
    TypeSolutionMethod typeSolutionMethod)
```

6.80.3. Documentación de datos miembro

6.80.3.1. `codification`

```
Codification problem.definition.Problem.codification [private]
```

6.80.3.2. `factorySolutionMethod`

```
IFFactorySolutionMethod problem.definition.Problem.factorySolutionMethod [private]
```

6.80.3.3. `function`

```
ArrayList<ObjectiveFunction> problem.definition.Problem.function [private]
```

6.80.3.4. `operator`

```
Operator problem.definition.Problem.operator [private]
```

6.80.3.5. `possibleValue`

```
int problem.definition.Problem.possibleValue [private]
```

6.80.3.6. `state`

```
State problem.definition.Problem.state [private]
```

6.80.3.7. `typeProblem`

```
ProblemType problem.definition.Problem.typeProblem [private]
```

6.80.3.8. `typeSolutionMethod`

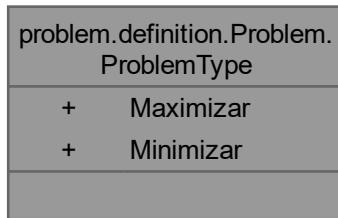
```
TypeSolutionMethod problem.definition.Problem.typeSolutionMethod [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/definition/Problem.java](#)

6.81. Referencia de la enumeración problem.definition.Problem.ProblemType

Diagrama de colaboración de problem.definition.Problem.ProblemType:



Atributos públicos

- [Maximizar](#)
- [Minimizar](#)

6.81.1. Documentación de datos miembro

6.81.1.1. Maximizar

`problem.definition.Problem. ProblemType.Maximizar`

6.81.1.2. Minimizar

`problem.definition.Problem. ProblemType.Minimizar`

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/problem/definition/Problem.java](#)

6.82. Referencia de la clase

local_search.candidate_type.RandomCandidate

Diagrama de herencia de local_search.candidate_type.RandomCandidate

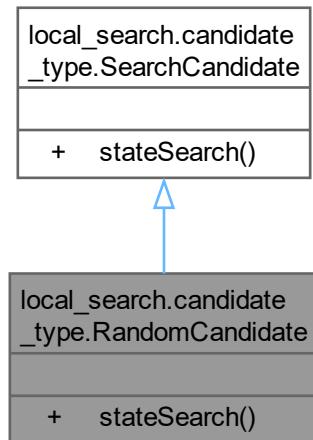
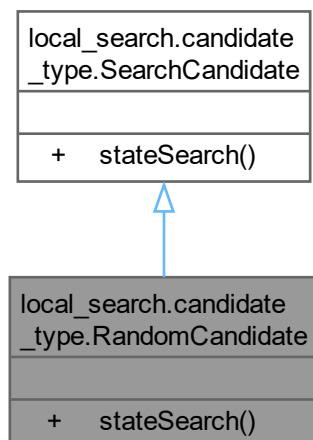


Diagrama de colaboración de local_search.candidate_type.RandomCandidate:



Métodos públicos

- [State stateSearch \(List< State > listNeighborhood\)](#)

6.82.1. Documentación de funciones miembro

6.82.1.1. stateSearch()

```
State local_search.candidate_type.RandomCandidate.stateSearch (
    List< State > listNeighborhood)
```

Reimplementado de [local_search.candidate_type.SearchCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/candidate_type/RandomCandidate.java](#)

6.83. Referencia de la clase metaheuristics.generators.RandomSearch

Diagrama de herencia de metaheuristics.generators.RandomSearch

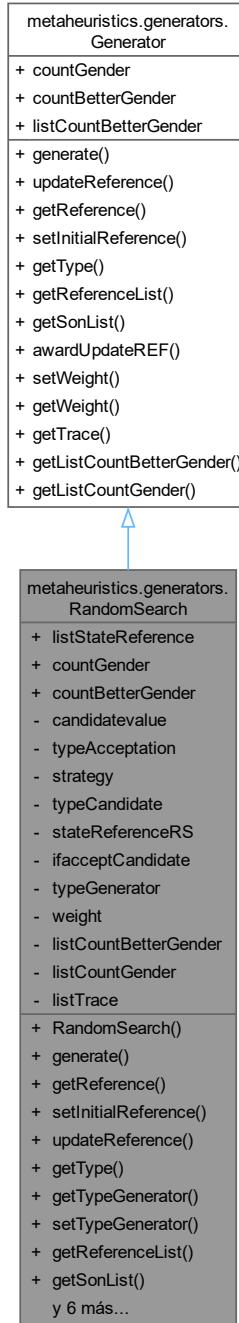
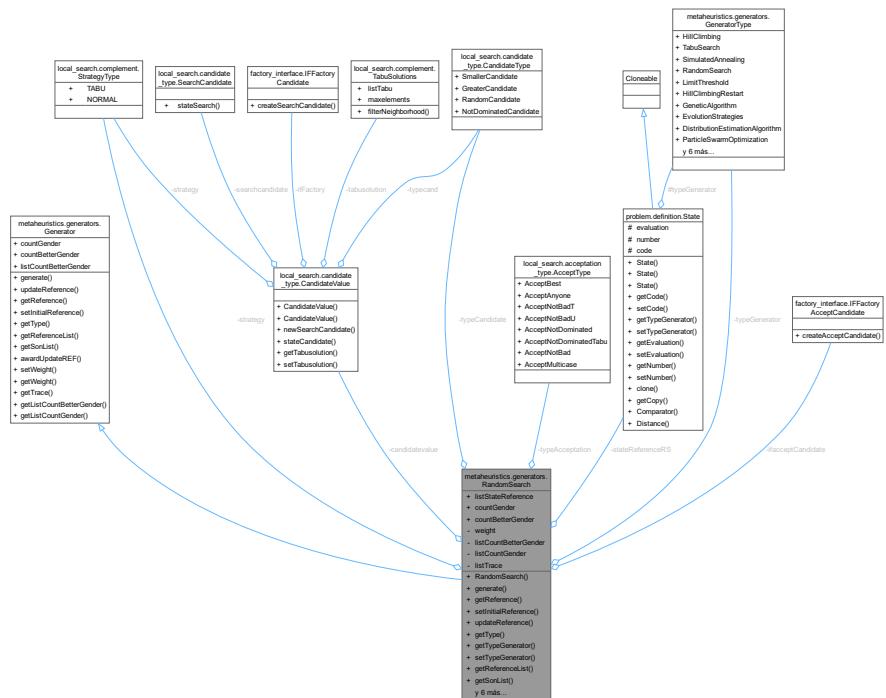


Diagrama de colaboración de metaheuristics.generators.RandomSearch:



Métodos públicos

- **RandomSearch ()**
 - **State generate (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**
 - **State getReference ()**
 - **void setInitialReference (State stateInitialRef)**
 - **void updateReference (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**
 - **GeneratorType getType ()**
 - **GeneratorType getTypeGenerator ()**
 - **void setTypeGenerator (GeneratorType typeGenerator)**
 - **List< State > getReferenceList ()**
 - **List< State > getSonList ()**
 - **boolean awardUpdateREF (State stateCandidate)**
 - **float getWeight ()**
 - **void setWeight (float weight)**
 - **int[] getListCountBetterGender ()**
 - **int[] getListCountGender ()**
 - **float[] getTrace ()**

Atributos públicos estáticos

- static List< State > listStateReference = new ArrayList<State>()
 - static int countGender = 0
 - static int countBetterGender = 0

Atributos privados

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceRS`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- `GeneratorType typeGenerator`
- float `weight`
- int[] `listCountBetterGender = new int[10]`
- int[] `listCountGender = new int[10]`
- float[] `listTrace = new float[1200000]`

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.83.1. Documentación de constructores y destructores

6.83.1.1. [RandomSearch\(\)](#)

```
metaheuristics.generators.RandomSearch.RandomSearch ()
```

6.83.2. Documentación de funciones miembro

6.83.2.1. [awardUpdateREF\(\)](#)

```
boolean metaheuristics.generators.RandomSearch.awardUpdateREF (
    State stateCandidate)
```

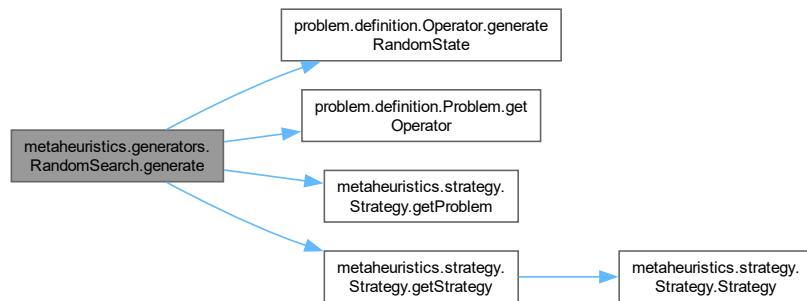
Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.2. generate()

```
State metaheuristics.generators.RandomSearch.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.83.2.3. getListCountBetterGender()

```
int[] metaheuristics.generators.RandomSearch.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.4. getListCountGender()

```
int[] metaheuristics.generators.RandomSearch.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.5. getReference()

```
State metaheuristics.generators.RandomSearch.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.6. getReferenceList()

```
List< State > metaheuristics.generators.RandomSearch.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.7. `getSonList()`

```
List< State > metaheuristics.generators.RandomSearch.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.8. `getTrace()`

```
float[ ] metaheuristics.generators.RandomSearch.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.9. `getType()`

```
GeneratorType metaheuristics.generators.RandomSearch.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.10. `getTypeGenerator()`

```
GeneratorType metaheuristics.generators.RandomSearch.getTypeGenerator ()
```

6.83.2.11. `getWeight()`

```
float metaheuristics.generators.RandomSearch.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.12. `setInitialReference()`

```
void metaheuristics.generators.RandomSearch.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.13. `setTypeGenerator()`

```
void metaheuristics.generators.RandomSearch.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.83.2.14. `setWeight()`

```
void metaheuristics.generators.RandomSearch.setWeight (
    float weight)
```

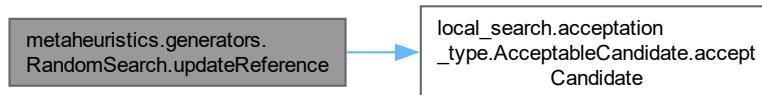
Reimplementado de [metaheuristics.generators.Generator](#).

6.83.2.15. updateReference()

```
void metaheuristics.generators.RandomSearch.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.83.3. Documentación de datos miembro

6.83.3.1. candidatevalue

```
CandidateValue metaheuristics.generators.RandomSearch.candidatevalue [private]
```

6.83.3.2. countBetterGender

```
int metaheuristics.generators.RandomSearch.countBetterGender = 0 [static]
```

6.83.3.3. countGender

```
int metaheuristics.generators.RandomSearch.countGender = 0 [static]
```

6.83.3.4. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.RandomSearch.ifacceptCandidate [private]
```

6.83.3.5. listCountBetterGender

```
int [] metaheuristics.generators.RandomSearch.listCountBetterGender = new int[10] [private]
```

6.83.3.6. **listCountGender**

```
int [] metaheuristics.generators.RandomSearch.listCountGender = new int[10] [private]
```

6.83.3.7. **listStateReference**

```
List<State> metaheuristics.generators.RandomSearch.listStateReference = new ArrayList<State>()  
[static]
```

6.83.3.8. **listTrace**

```
float [] metaheuristics.generators.RandomSearch.listTrace = new float[1200000] [private]
```

6.83.3.9. **stateReferenceRS**

```
State metaheuristics.generators.RandomSearch.stateReferenceRS [private]
```

6.83.3.10. **strategy**

```
StrategyType metaheuristics.generators.RandomSearch.strategy [private]
```

6.83.3.11. **typeAcceptation**

```
AcceptType metaheuristics.generators.RandomSearch.typeAcceptation [private]
```

6.83.3.12. **typeCandidate**

```
CandidateType metaheuristics.generators.RandomSearch.typeCandidate [private]
```

6.83.3.13. **typeGenerator**

```
GeneratorType metaheuristics.generators.RandomSearch.typeGenerator [private]
```

6.83.3.14. **weight**

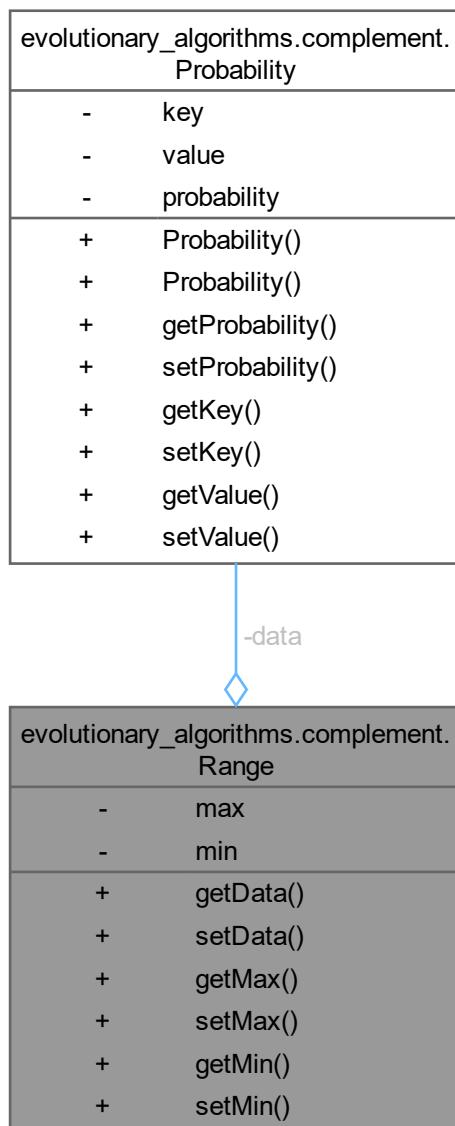
```
float metaheuristics.generators.RandomSearch.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/RandomSearch.java](#)

6.84. Referencia de la clase evolutionary_algorithms.complement.Range

Diagrama de colaboración de evolutionary_algorithms.complement.Range:



Métodos públicos

- `Probability getData ()`
- `void setData (Probability data)`
- `float getMax ()`
- `void setMax (float max)`
- `float getMin ()`
- `void setMin (float min)`

Atributos privados

- `Probability data`
- float `max`
- float `min`

6.84.1. Documentación de funciones miembro

6.84.1.1. `getData()`

```
Probability evolutionary_algorithms.complement.Range.getData ()
```

6.84.1.2. `getMax()`

```
float evolutionary_algorithms.complement.Range.getMax ()
```

6.84.1.3. `getMin()`

```
float evolutionary_algorithms.complement.Range.getMin ()
```

6.84.1.4. `setData()`

```
void evolutionary_algorithms.complement.Range.setData (
    Probability data)
```

6.84.1.5. `setMax()`

```
void evolutionary_algorithms.complement.Range.setMax (
    float max)
```

6.84.1.6. `setMin()`

```
void evolutionary_algorithms.complement.Range.setMin (
    float min)
```

6.84.2. Documentación de datos miembro

6.84.2.1. `data`

```
Probability evolutionary_algorithms.complement.Range.data [private]
```

6.84.2.2. max

```
float evolutionary_algorithms.complement.Range.max [private]
```

6.84.2.3. min

```
float evolutionary_algorithms.complement.Range.min [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/Range.java](#)

6.85. Referencia de la clase evolutionary_algorithms.complement.Replace

Diagrama de herencia de evolutionary_algorithms.complement.Replace

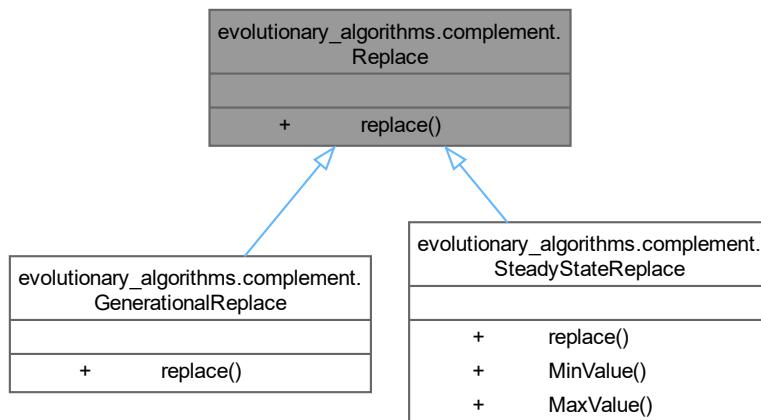
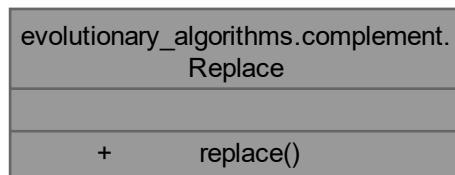


Diagrama de colaboración de evolutionary_algorithms.complement.Replace:



Métodos públicos

- abstract List< State > replace (State stateCandidate, List< State >listState) throws IllegalArgument←Exception, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

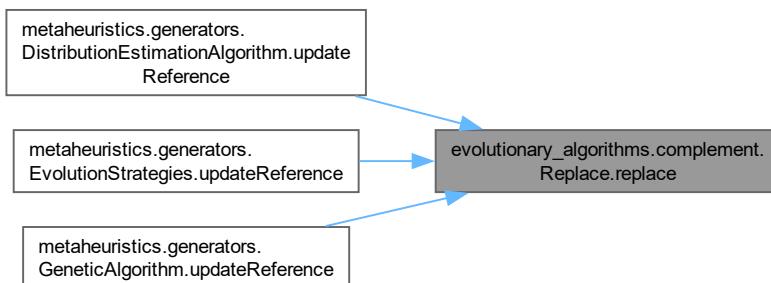
6.85.1. Documentación de funciones miembro

6.85.1.1. replace()

```
abstract List< State > evolutionary_algorithms.complement.Replace.replace (
    State stateCandidate,
    List< State > listState) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException [abstract]
```

Reimplementado en [evolutionary_algorithms.complement.GenerationalReplace](#) y [evolutionary_algorithms.complement.SteadyStateReplace](#)

Gráfico de llamadas a esta función:

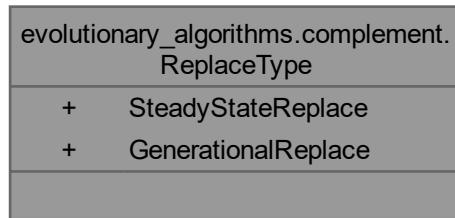


La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/Replace.java](#)

6.86. Referencia de la enumeración evolutionary_algorithms.complement.ReplaceType

Diagrama de colaboración de evolutionary_algorithms.complement.ReplaceType:



Atributos públicos

- [SteadyStateReplace](#)
- [GenerationalReplace](#)

6.86.1. Documentación de datos miembro

6.86.1.1. GenerationalReplace

`evolutionary_algorithms.complement.ReplaceType.GenerationalReplace`

6.86.1.2. SteadyStateReplace

`evolutionary_algorithms.complement.ReplaceType.SteadyStateReplace`

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/ReplaceType.java](#)

6.87. Referencia de la clase

evolutionary_algorithms.complement.RouletteSelection

Diagrama de herencia de evolutionary_algorithms.complement.RouletteSelection

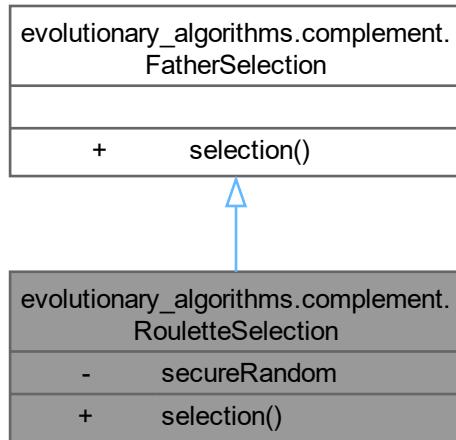
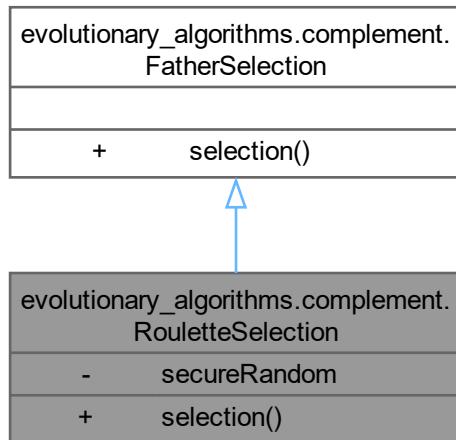


Diagrama de colaboración de evolutionary_algorithms.complement.RouletteSelection:



Métodos públicos

- List< State > `selection` (List< State > listState, int truncation)

Atributos estáticos privados

- static final SecureRandom `secureRandom` = new SecureRandom()

6.87.1. Documentación de funciones miembro

6.87.1.1. selection()

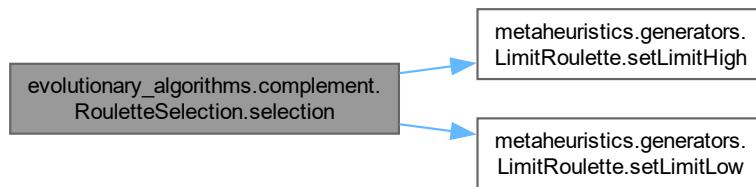
```
List< State > evolutionary_algorithms.complement.RouletteSelection.selection (
```

```
    List< State > listState,
```

```
    int truncation)
```

Reimplementado de [evolutionary_algorithms.complement.FatherSelection](#).

Gráfico de llamadas de esta función:



6.87.2. Documentación de datos miembro

6.87.2.1. secureRandom

```
final SecureRandom evolutionary_algorithms.complement.RouletteSelection.secureRandom = new  
SecureRandom() [static], [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/RouletteSelection.java](#)

6.88. Referencia de la clase

evolutionary_algorithms.complement.Sampling

Diagrama de herencia de evolutionary_algorithms.complement.Sampling

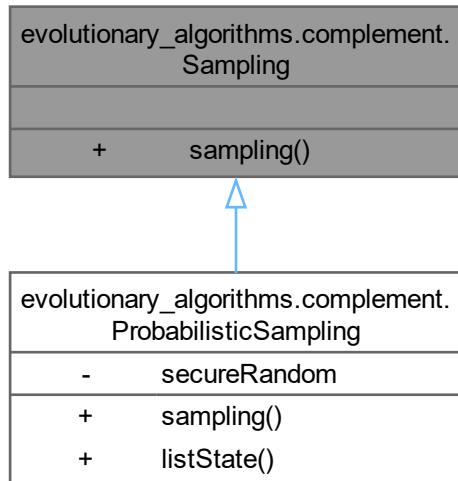
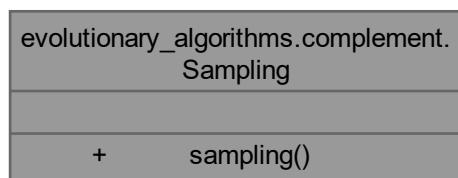


Diagrama de colaboración de `evolutionary_algorithms.complement.Sampling`:



Métodos públicos

- abstract List<[State](#)> `sampling` (List<[State](#)> fathers, int countInd)

6.88.1. Documentación de funciones miembro

6.88.1.1. sampling()

```
abstract List< State > evolutionary_algorithms.complement.Sampling.sampling (   
    List< State > fathers,   
    int countInd) [abstract]
```

Reimplementado en [evolutionary_algorithms.complement.ProbabilisticSampling](#).

Gráfico de llamadas a esta función:

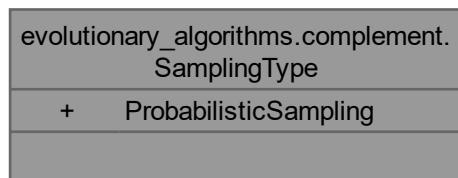


La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/Sampling.java](#)

6.89. Referencia de la enumeración evolutionary_algorithms.complement.SamplingType

Diagrama de colaboración de `evolutionary_algorithms.complement.SamplingType`:



Atributos públicos

- [ProbabilisticSampling](#)

6.89.1. Documentación de datos miembro

6.89.1.1. ProbabilisticSampling

`evolutionary_algorithms.complement.SamplingType.ProbabilisticSampling`

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/SamplingType.java](#)

6.90. Referencia de la clase

`local_search.candidate_type.SearchCandidate`

Diagrama de herencia de `local_search.candidate_type.SearchCandidate`

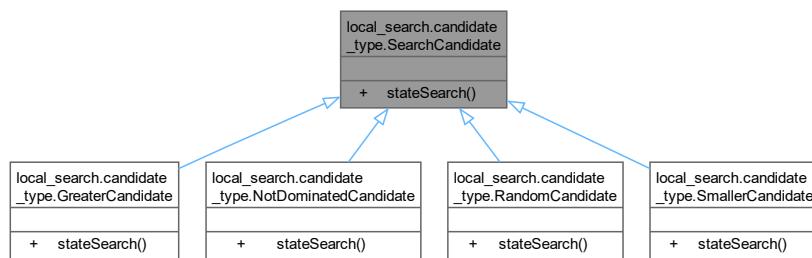


Diagrama de colaboración de `local_search.candidate_type.SearchCandidate`:



Métodos públicos

- abstract `State stateSearch (List< State > listNeighborhood)` throws `IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`

6.90.1. Documentación de funciones miembro

6.90.1.1. stateSearch()

```
abstract State local_search.candidate_type.SearchCandidate.stateSearch (
    List< State > listNeighborhood) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException [abstract]
```

Reimplementado en [local_search.candidate_type.GreaterCandidate](#), [local_search.candidate_type.NotDominatedCandidate](#), [local_search.candidate_type.RandomCandidate](#) y [local_search.candidate_type.SmallerCandidate](#).

Gráfico de llamadas a esta función:

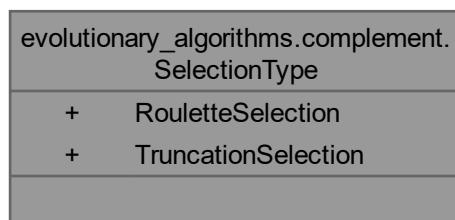


La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/candidate_type/SearchCandidate.java](#)

6.91. Referencia de la enumeración evolutionary_algorithms.complement.SelectionType

Diagrama de colaboración de `evolutionary_algorithms.complement.SelectionType`:



Atributos públicos

- [RouletteSelection](#)
- [TruncationSelection](#)

6.91.1. Documentación de datos miembro

6.91.1.1. RouletteSelection

```
evolutionary_algorithms.complement.SelectionType.RouletteSelection
```

6.91.1.2. TruncationSelection

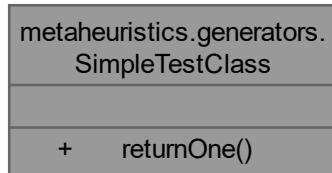
```
evolutionary_algorithms.complement.SelectionType.TruncationSelection
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/SelectionType.java](#)

6.92. Referencia de la clase metaheuristics.generators.SimpleTestClass

Diagrama de colaboración de metaheuristics.generators.SimpleTestClass:



Métodos públicos

- [int returnOne \(\)](#)

6.92.1. Documentación de funciones miembro

6.92.1.1. returnOne()

```
int metaheuristics.generators.SimpleTestClass.returnOne ()
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/SimpleTestClass.java](#)

6.93. Referencia de la clase metaheuristics.generators.SimulatedAnnealing

Diagrama de herencia de metaheuristics.generators.SimulatedAnnealing

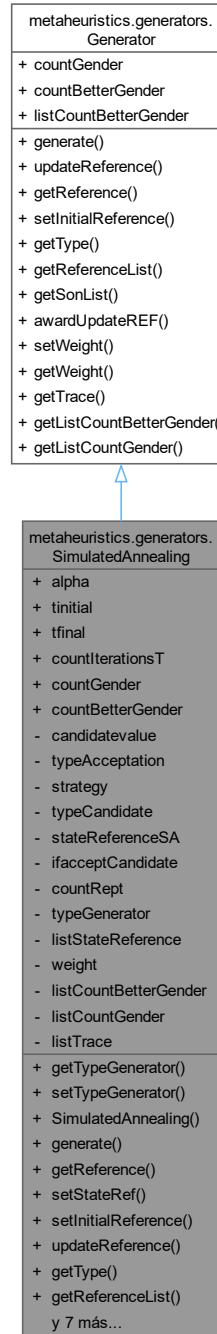
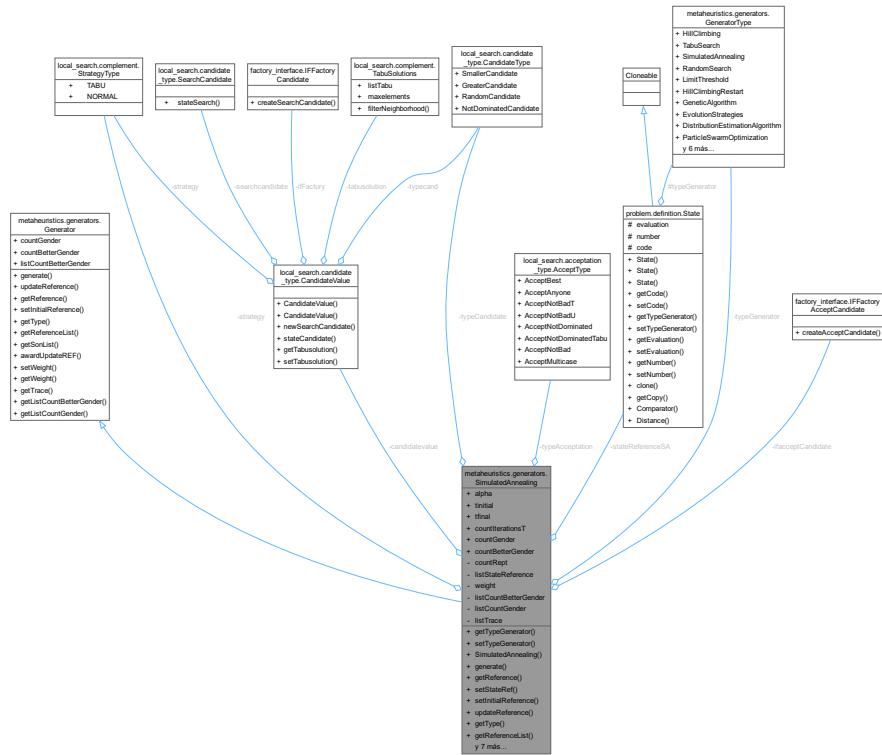


Diagrama de colaboración de metaheuristics.generators.SimulatedAnnealing:



Métodos públicos

- `GeneratorType getTypeGenerator ()`
- `void setTypeGenerator (GeneratorType typeGenerator)`
- `SimulatedAnnealing ()`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `State getReference ()`
- `void setStateRef (State stateRef)`
- `void setInitialReference (State stateInitialRef)`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `GeneratorType getType ()`
- `List< State > getReferenceList ()`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `int[] getCountBetterGender ()`
- `int[] getCountGender ()`
- `float[] getTrace ()`

Atributos públicos estáticos

- static Double `alpha`
- static Double `tinitial`
- static Double `tfinal`
- static int `countIterationsT`
- static int `countGender` = 0
- static int `countBetterGender` = 0

Atributos privados

- `CandidateValue candidatevalue`
- `AcceptType typeAcceptation`
- `StrategyType strategy`
- `CandidateType typeCandidate`
- `State stateReferenceSA`
- `IFFactoryAcceptCandidate ifacceptCandidate`
- int `countRept`
- `GeneratorType typeGenerator`
- List< `State` > `listStateReference` = new ArrayList<`State`>()
- float `weight`
- int[] `listCountBetterGender` = new int[10]
- int[] `listCountGender` = new int[10]
- float[] `listTrace` = new float[1200000]

Otros miembros heredados

Atributos públicos heredados de `metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.93.1. Documentación de constructores y destructores

6.93.1.1. `SimulatedAnnealing()`

```
metaheuristics.generators.SimulatedAnnealing.SimulatedAnnealing ()
```

6.93.2. Documentación de funciones miembro

6.93.2.1. `awardUpdateREF()`

```
boolean metaheuristics.generators.SimulatedAnnealing.awardUpdateREF (
    State stateCandidate)
```

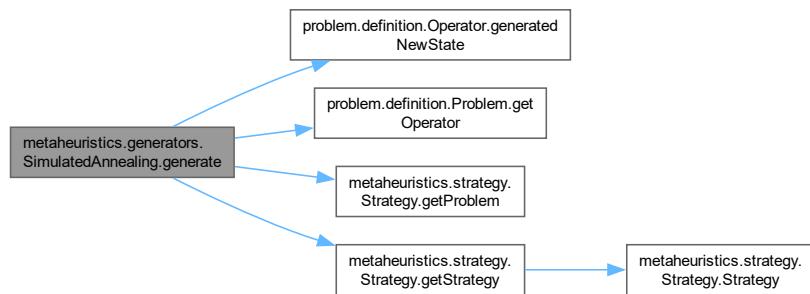
Reimplementado de `metaheuristics.generators.Generator`.

6.93.2.2. generate()

```
State metaheuristics.generators.SimulatedAnnealing.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class<-
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.93.2.3. getListCountBetterGender()

```
int[] metaheuristics.generators.SimulatedAnnealing.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.4. getListCountGender()

```
int[] metaheuristics.generators.SimulatedAnnealing.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.5. getReference()

```
State metaheuristics.generators.SimulatedAnnealing.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.6. getReferenceList()

```
List< State > metaheuristics.generators.SimulatedAnnealing.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.7. getSonList()

```
List< State > metaheuristics.generators.SimulatedAnnealing.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.8. getTrace()

```
float[ ] metaheuristics.generators.SimulatedAnnealing.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.9. getType()

```
GeneratorType metaheuristics.generators.SimulatedAnnealing.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.10. getTypeGenerator()

```
GeneratorType metaheuristics.generators.SimulatedAnnealing.getTypeGenerator ()
```

6.93.2.11. getWeight()

```
float metaheuristics.generators.SimulatedAnnealing.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.12. setInitialReference()

```
void metaheuristics.generators.SimulatedAnnealing.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.13. setStateRef()

```
void metaheuristics.generators.SimulatedAnnealing.setStateRef (
    State stateRef)
```

6.93.2.14. setTypeGenerator()

```
void metaheuristics.generators.SimulatedAnnealing.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.93.2.15. `setWeight()`

```
void metaheuristics.generators.SimulatedAnnealing.setWeight (
    float weight)
```

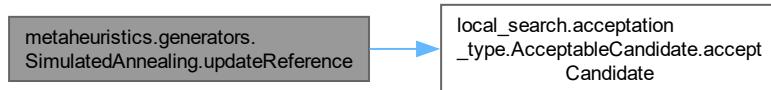
Reimplementado de [metaheuristics.generators.Generator](#).

6.93.2.16. `updateReference()`

```
void metaheuristics.generators.SimulatedAnnealing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.93.3. Documentación de datos miembro

6.93.3.1. `alpha`

```
Double metaheuristics.generators.SimulatedAnnealing.alpha [static]
```

6.93.3.2. `candidatevalue`

```
CandidateValue metaheuristics.generators.SimulatedAnnealing.candidatevalue [private]
```

6.93.3.3. `countBetterGender`

```
int metaheuristics.generators.SimulatedAnnealing.countBetterGender = 0 [static]
```

6.93.3.4. `countGender`

```
int metaheuristics.generators.SimulatedAnnealing.countGender = 0 [static]
```

6.93.3.5. countIterationsT

```
int metaheuristics.generators.SimulatedAnnealing.countIterationsT [static]
```

6.93.3.6. countRept

```
int metaheuristics.generators.SimulatedAnnealing.countRept [private]
```

6.93.3.7. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.SimulatedAnnealing.ifacceptCandidate [private]
```

6.93.3.8. listCountBetterGender

```
int [] metaheuristics.generators.SimulatedAnnealing.listCountBetterGender = new int[10] [private]
```

6.93.3.9. listCountGender

```
int [] metaheuristics.generators.SimulatedAnnealing.listCountGender = new int[10] [private]
```

6.93.3.10. listStateReference

```
List<State> metaheuristics.generators.SimulatedAnnealing.listStateReference = new Array←  
List<State>() [private]
```

6.93.3.11. listTrace

```
float [] metaheuristics.generators.SimulatedAnnealing.listTrace = new float[1200000] [private]
```

6.93.3.12. stateReferenceSA

```
State metaheuristics.generators.SimulatedAnnealing.stateReferenceSA [private]
```

6.93.3.13. strategy

```
StrategyType metaheuristics.generators.SimulatedAnnealing.strategy [private]
```

6.93.3.14. tfinal

```
Double metaheuristics.generators.SimulatedAnnealing.tfinal [static]
```

6.93.3.15. `tinitial`

```
Double metaheuristics.generators.SimulatedAnnealing.tinitial [static]
```

6.93.3.16. `typeAcceptation`

```
AcceptType metaheuristics.generators.SimulatedAnnealing.typeAcceptation [private]
```

6.93.3.17. `typeCandidate`

```
CandidateType metaheuristics.generators.SimulatedAnnealing.typeCandidate [private]
```

6.93.3.18. `typeGenerator`

```
GeneratorType metaheuristics.generators.SimulatedAnnealing.typeGenerator [private]
```

6.93.3.19. `weight`

```
float metaheuristics.generators.SimulatedAnnealing.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/SimulatedAnnealing.java](#)

6.94. Referencia de la clase

`local_search.candidate_type.SmallerCandidate`

Diagrama de herencia de `local_search.candidate_type.SmallerCandidate`

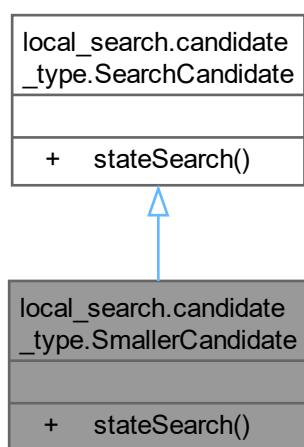
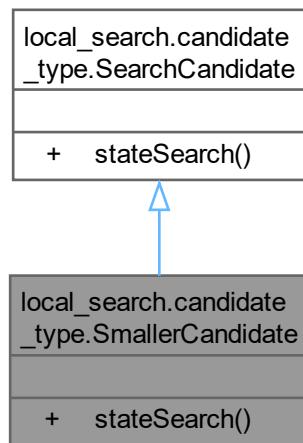


Diagrama de colaboración de local_search.candidate_type.SmallerCandidate:



Métodos públicos

- **State stateSearch (List< State > listNeighborhood) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException**

6.94.1. Documentación de funciones miembro

6.94.1.1. stateSearch()

```
State local_search.candidate_type.SmallerCandidate.stateSearch (
    List< State > listNeighborhood) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [local_search.candidate_type.SearchCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/candidate_type/SmallerCandidate.java](#)

6.95. Referencia de la clase problem.extension.SolutionMethod

Diagrama de herencia de problem.extension.SolutionMethod

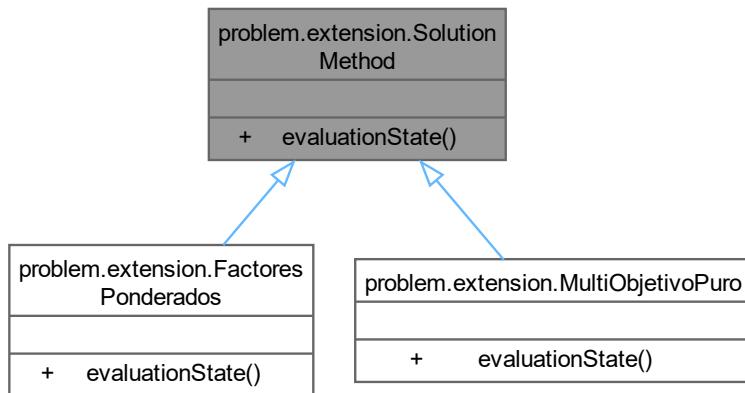
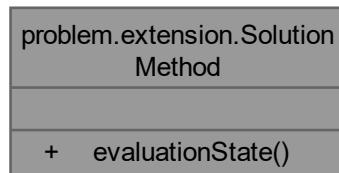


Diagrama de colaboración de problem.extension.SolutionMethod:



Métodos públicos

- abstract void `evaluationState (State state)`

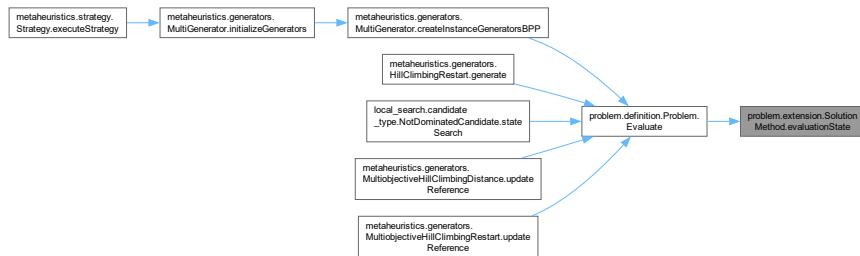
6.95.1. Documentación de funciones miembro

6.95.1.1. `evaluationState()`

```
abstract void problem.extension.SolutionMethod.evaluationState (
    State state) [abstract]
```

Reimplementado en [problem.extension.FactoresPonderados](#) y [problem.extension.MultiObjetivoPuro](#).

Gráfico de llamadas a esta función:



La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/extension/SolutionMethod.java](#)

6.96. Referencia de la clase problem.definition.State

Diagrama de herencia de problem.definition.State

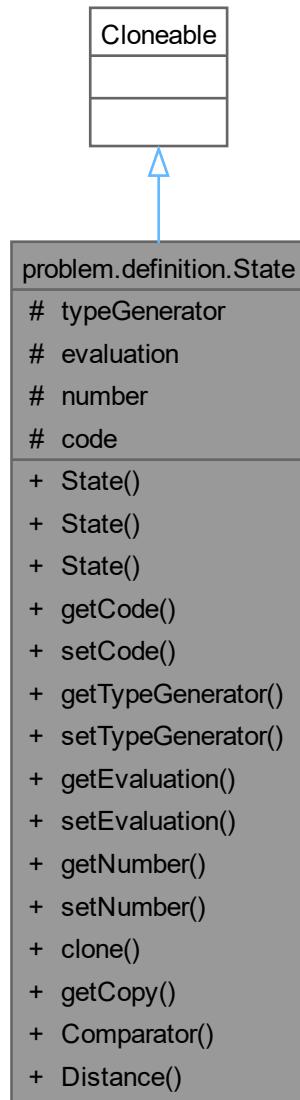
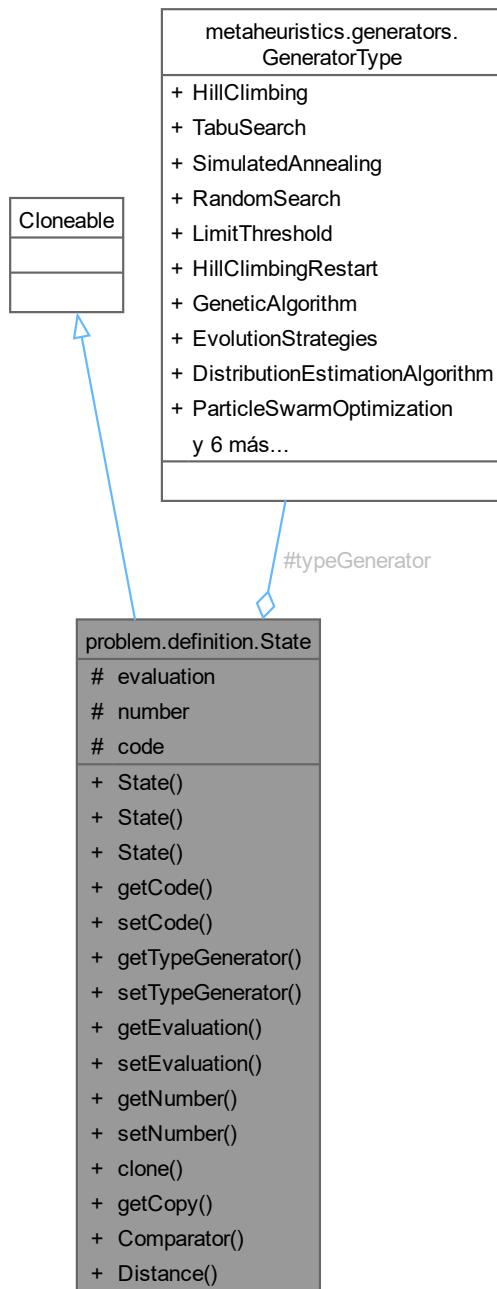


Diagrama de colaboración de problem.definition.State:



Métodos públicos

- `State (State ps)`
- `State (ArrayList< Object > code)`
- `State ()`
- `ArrayList< Object > getCode ()`
- `void setCode (ArrayList< Object > listCode)`

- `GeneratorType getTypeGenerator ()`
- `void setTypeGenerator (GeneratorType typeGenerator)`
- `ArrayList< Double > getEvaluation ()`
- `void setEvaluation (ArrayList< Double > evaluation)`
- `int getNumber ()`
- `void setNumber (int number)`
- `State clone ()`
- `Object getCopy ()`
- `boolean Comparator (State state)`
- `double Distance (State state)`

Atributos protegidos

- `GeneratorType typeGenerator`
- `ArrayList< Double > evaluation`
- `int number`
- `ArrayList< Object > code`

6.96.1. Documentación de constructores y destructores

6.96.1.1. State() [1/3]

```
problem.definition.State.State (
    State ps)
```

Gráfico de llamadas de esta función:

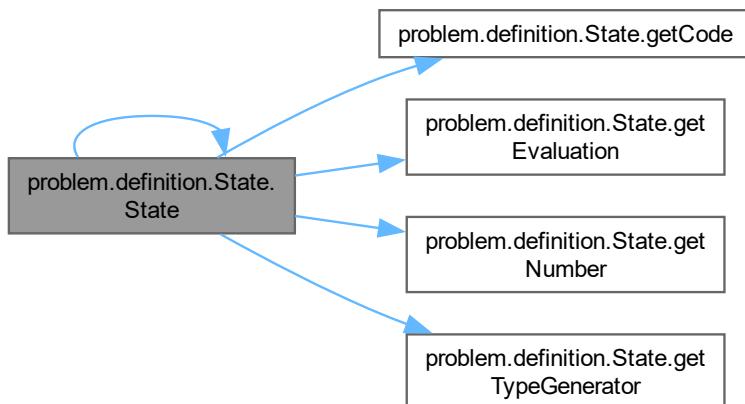
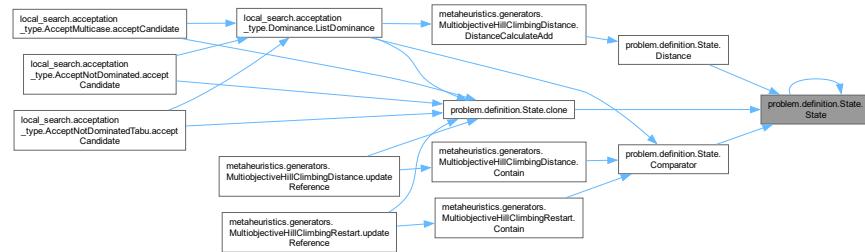


Gráfico de llamadas a esta función:



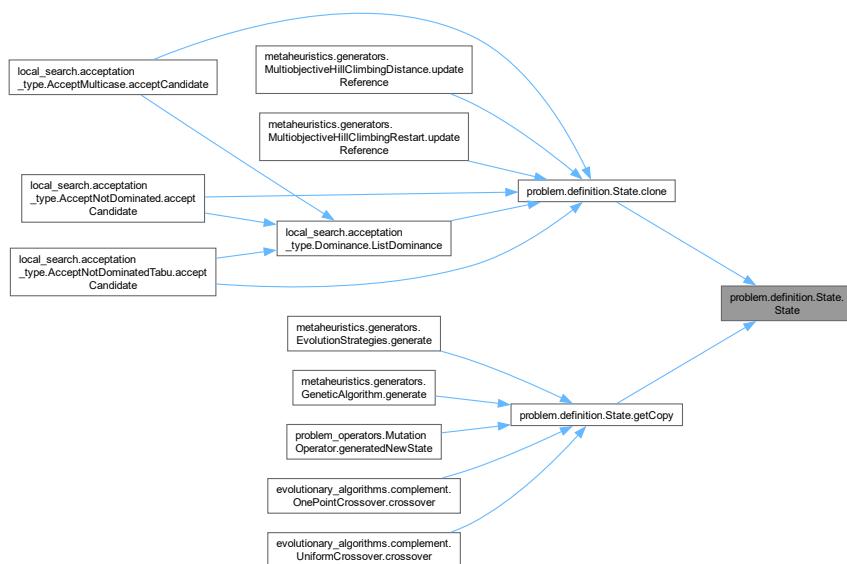
6.96.1.2. State() [2/3]

```
problem.definition.State.State (
    ArrayList< Object > code)
```

6.96.1.3. State() [3/3]

```
problem.definition.State.State ()
```

Gráfico de llamadas a esta función:



6.96.2. Documentación de funciones miembro

6.96.2.1. clone()

```
State problem.definition.State.clone ()
```

Gráfico de llamadas de esta función:

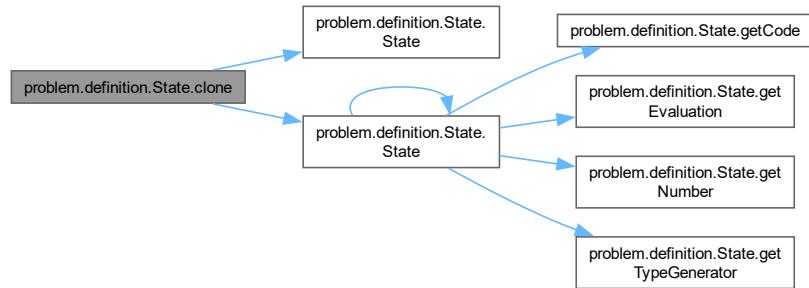
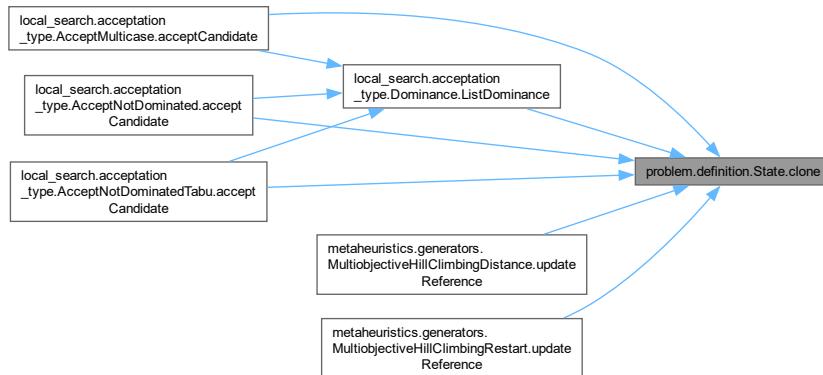


Gráfico de llamadas a esta función:



6.96.2.2. Comparator()

```
boolean problem.definition.State.Comparator (
    State state)
```

Gráfico de llamadas de esta función:

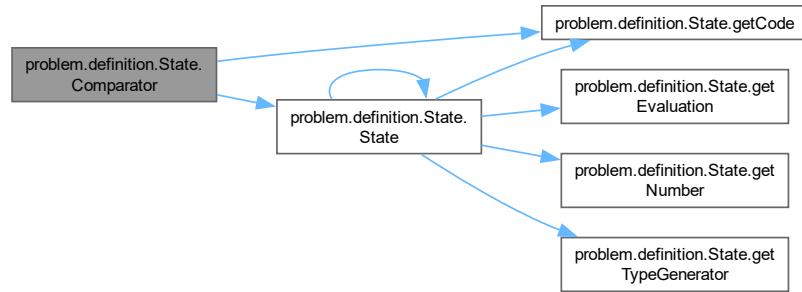
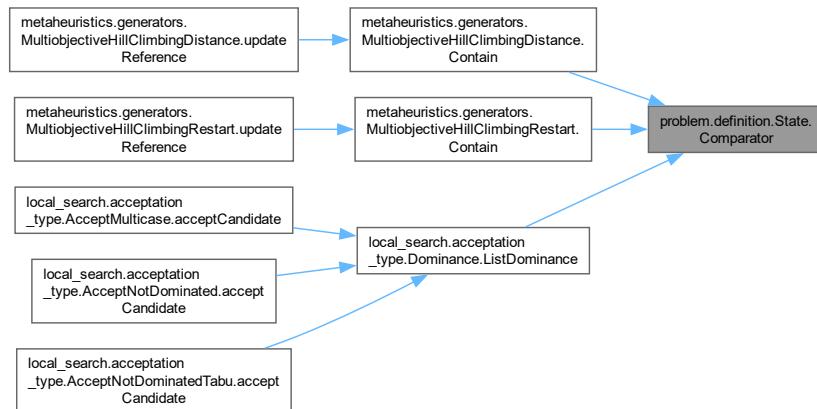


Gráfico de llamadas a esta función:



6.96.2.3. Distance()

```
double problem.definition.State.Distance (
    State state)
```

Gráfico de llamadas de esta función:

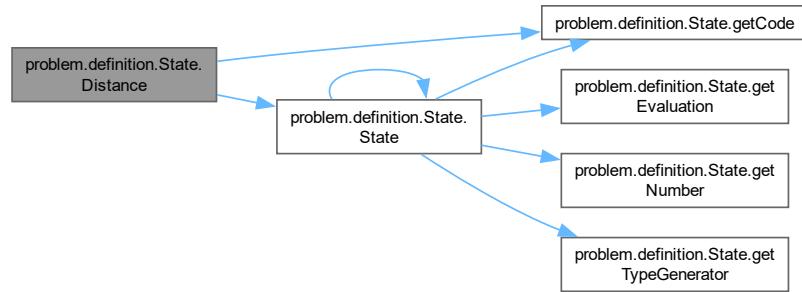
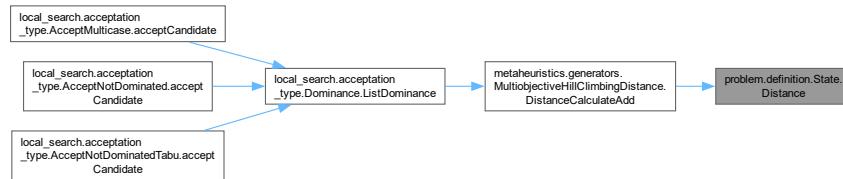


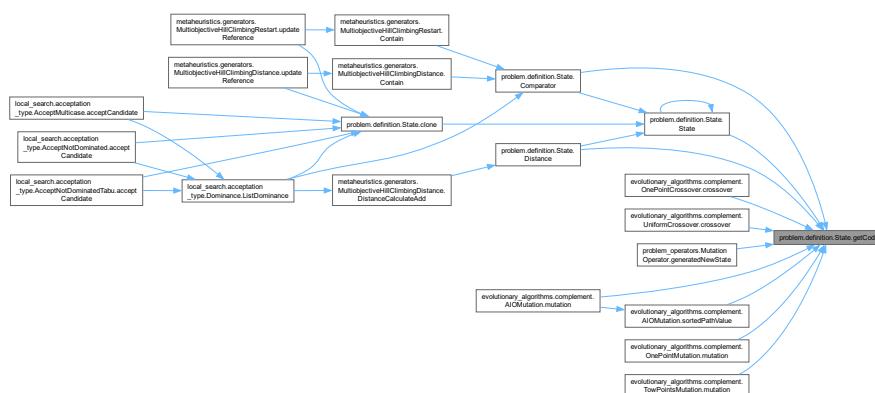
Gráfico de llamadas a esta función:



6.96.2.4. getCode()

```
ArrayList< Object > problem.definition.State.getCode ()
```

Gráfico de llamadas a esta función:



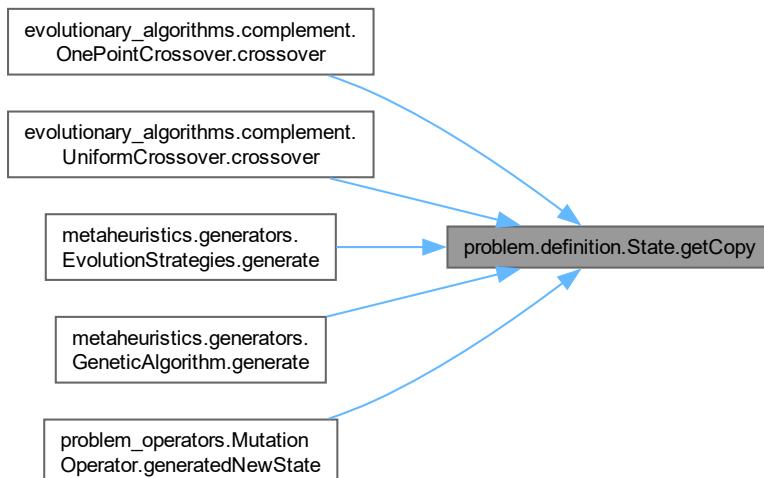
6.96.2.5. `getCopy()`

```
Object problem.definition.State.getCopy ()
```

Gráfico de llamadas de esta función:



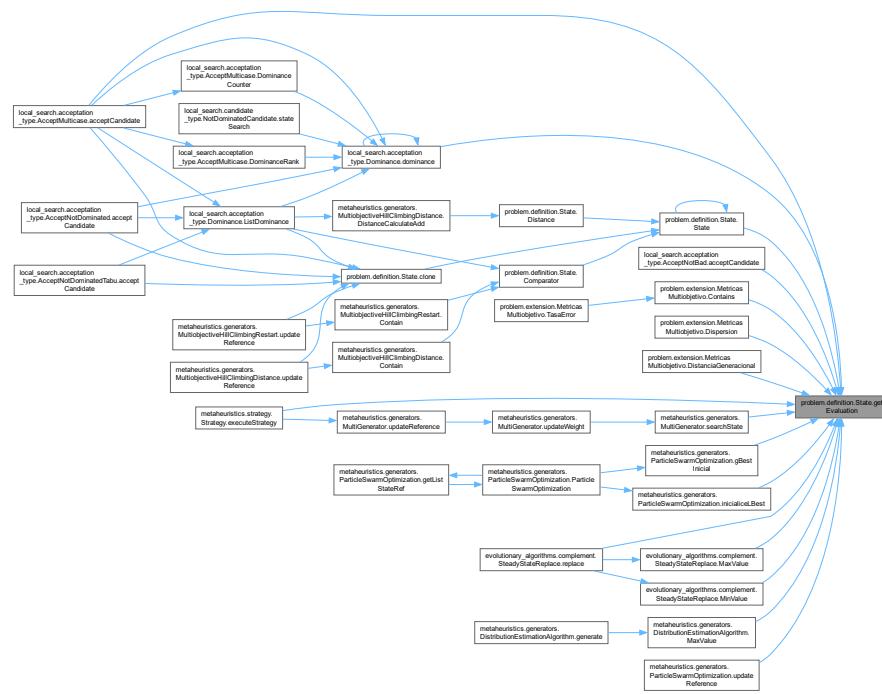
Gráfico de llamadas a esta función:



6.96.2.6. `getEvaluation()`

```
ArrayList< Double > problem.definition.State.getEvaluation ()
```

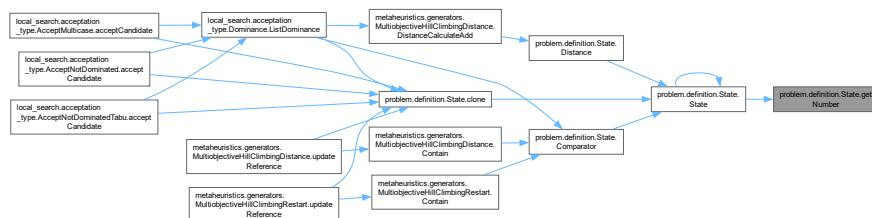
Gráfico de llamadas a esta función:



6.96.2.7. `getNumber()`

```
int problem.definition.State.getNumber ()
```

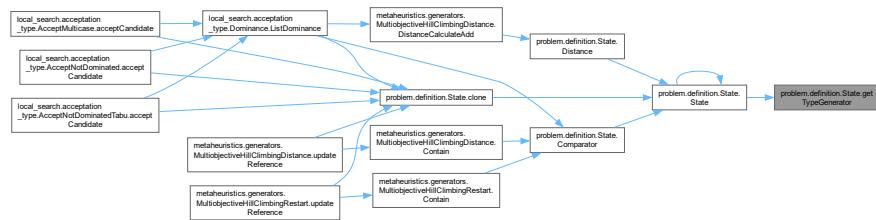
Gráfico de llamadas a esta función:



6.96.2.8. `getTypeGenerator()`

```
GeneratorType problem.definition.State.getTypeGenerator ()
```

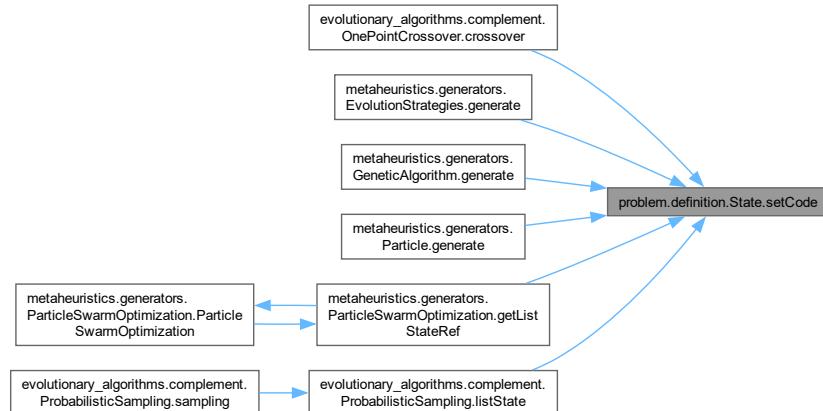
Gráfico de llamadas a esta función:



6.96.2.9. setCode()

```
void problem.definition.State.setCode (  
    ArrayList< Object > listCode)
```

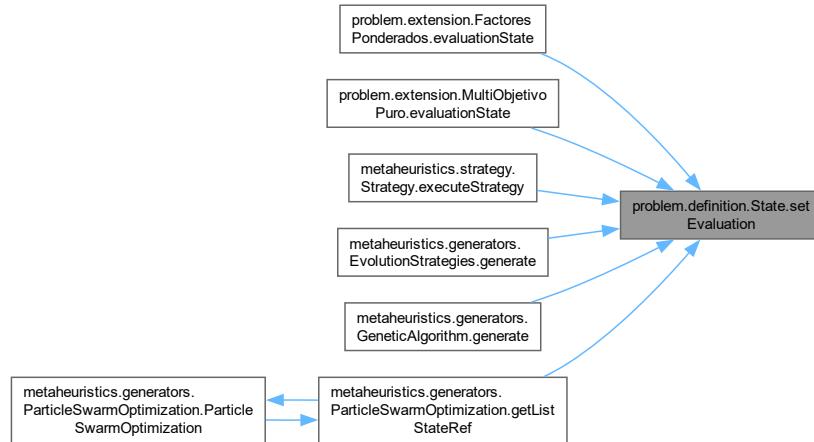
Gráfico de llamadas a esta función:



6.96.2.10. setEvaluation()

```
void problem.definition.State.setEvaluation (  
    ArrayList< Double > evaluation)
```

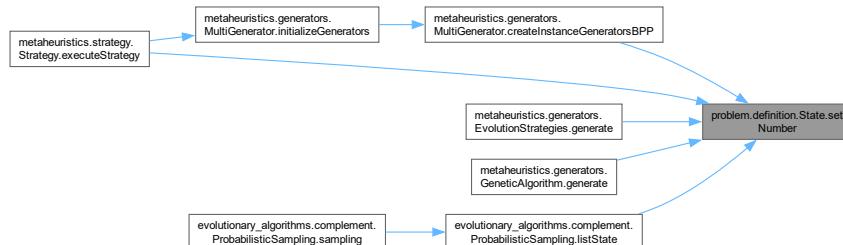
Gráfico de llamadas a esta función:



6.96.2.11. `setNumber()`

```
void problem.definition.State.setNumber (
    int number)
```

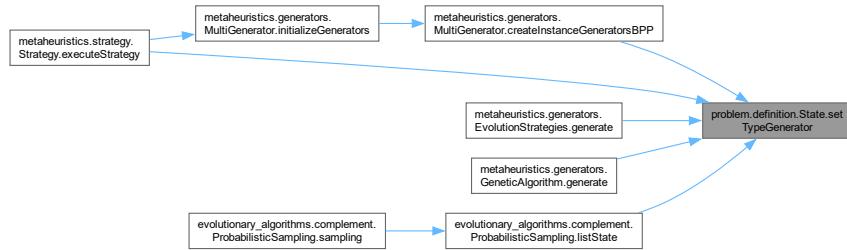
Gráfico de llamadas a esta función:



6.96.2.12. `setTypeGenerator()`

```
void problem.definition.State.setTypeGenerator (
    GeneratorType typeGenerator)
```

Gráfico de llamadas a esta función:



6.96.3. Documentación de datos miembro

6.96.3.1. code

```
ArrayList<Object> problem.definition.State.code [protected]
```

6.96.3.2. evaluation

```
ArrayList<Double> problem.definition.State.evaluation [protected]
```

6.96.3.3. number

```
int problem.definition.State.number [protected]
```

6.96.3.4. typeGenerator

```
GeneratorType problem.definition.State.typeGenerator [protected]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/problem/definition/State.java](#)

6.97. Referencia de la clase

evolutionary_algorithms.complement.SteadyStateReplace

Diagrama de herencia de evolutionary_algorithms.complement.SteadyStateReplace

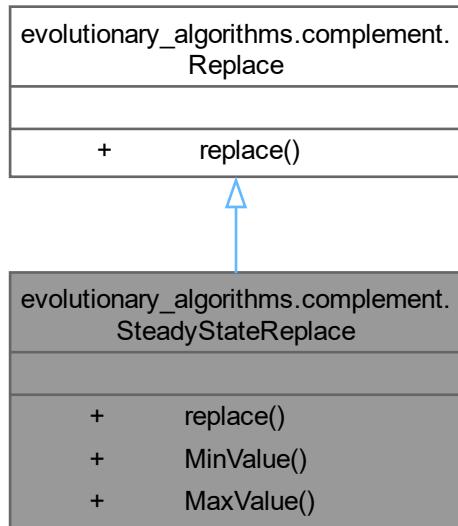
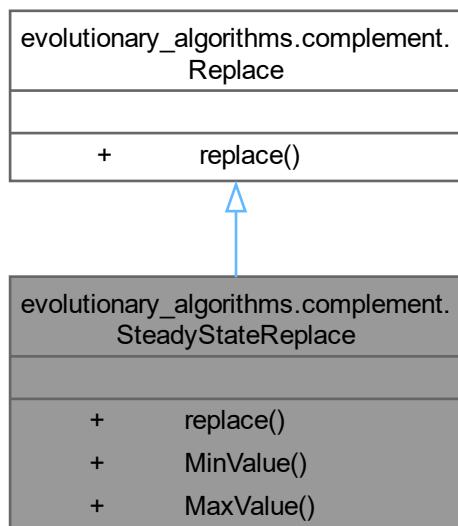


Diagrama de colaboración de `evolutionary_algorithms.complement.SteadyStateReplace`:



Métodos públicos

- `List< State > replace (State stateCandidate, List< State > listState)`
- `State MinValue (List< State > listState)`
- `State MaxValue (List< State > listState)`

6.97.1. Documentación de funciones miembro

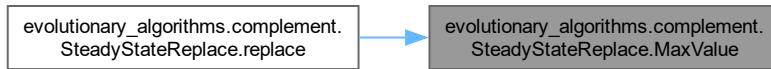
6.97.1.1. `MaxValue()`

```
State evolutionary_algorithms.complement.SteadyStateReplace.MaxValue (
    List< State > listState)
```

Gráfico de llamadas de esta función:



Gráfico de llamadas a esta función:



6.97.1.2. `MinValue()`

```
State evolutionary_algorithms.complement.SteadyStateReplace.MinValue (
    List< State > listState)
```

Gráfico de llamadas de esta función:

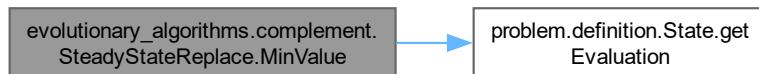
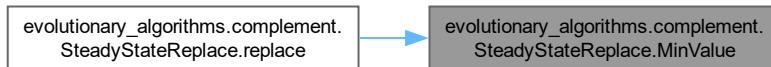


Gráfico de llamadas a esta función:

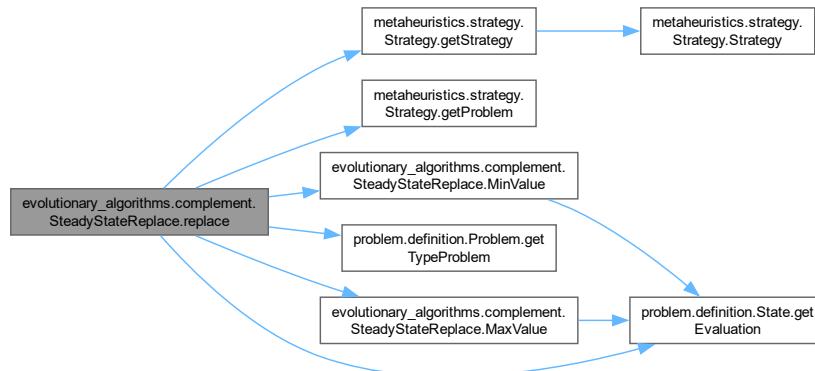


6.97.1.3. replace()

```
List< State > evolutionary_algorithms.complement.SteadyStateReplace.replace (
    State stateCandidate,
    List< State > listState)
```

Reimplementado de [evolutionary_algorithms.complement.Replace](#).

Gráfico de llamadas de esta función:

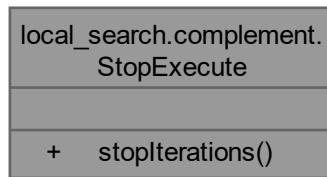


La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/SteadyStateReplace.java](#)

6.98. Referencia de la clase local_search.complement.StopExecute

Diagrama de colaboración de local_search.complement.StopExecute:



Métodos públicos

- Boolean [stopIterations](#) (int countIterationsCurrent, int countmaxIterations)

6.98.1. Documentación de funciones miembro

6.98.1.1. stopIterations()

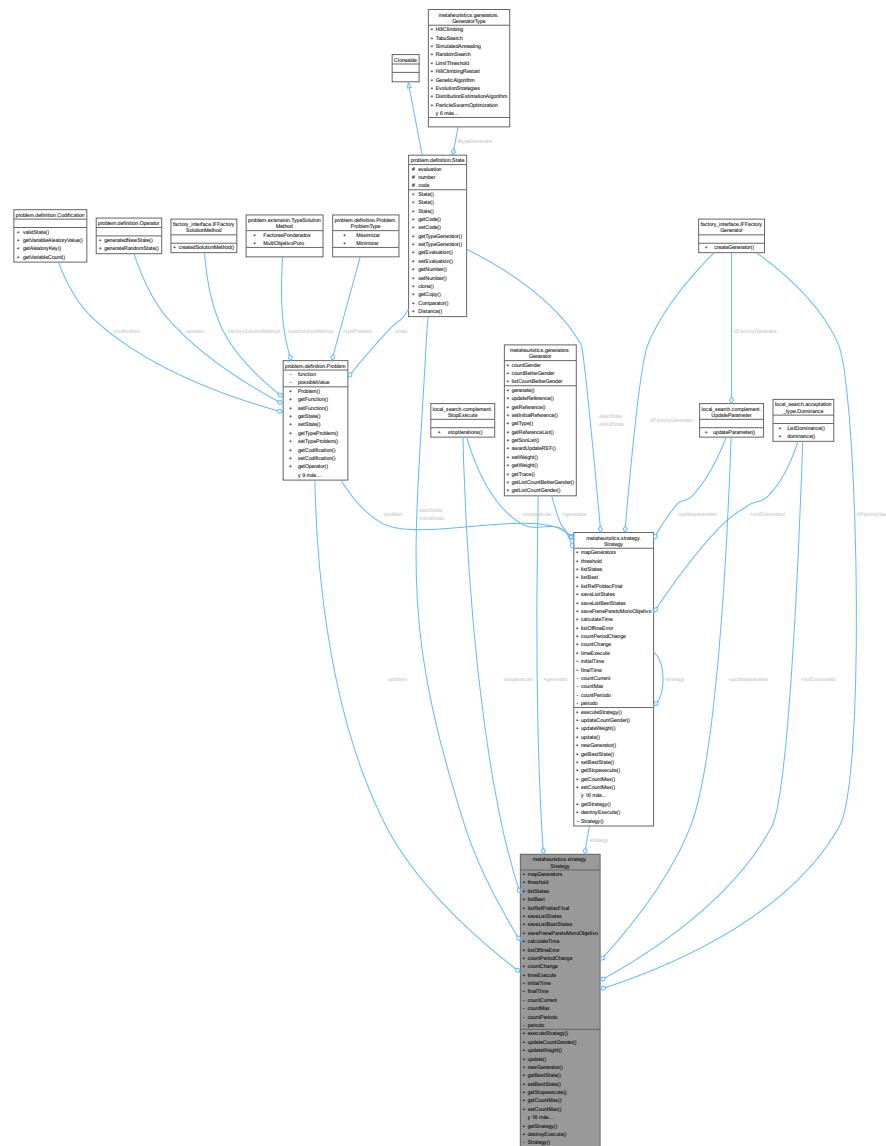
```
Boolean local_search.complement.StopExecute.stopIterations (
    int countIterationsCurrent,
    int countmaxIterations)
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/complement/StopExecute.java](#)

6.99. Referencia de la clase metaheuristics.strategy.Strategy

Diagrama de colaboración de metaheuristics.strategy.Strategy:



Métodos públicos

- void **executeStrategy** (int countmaxIterations, int countIterationsChange, int operatornumber, **GeneratorType** generatorType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- void **updateCountGender** ()
- void **updateWeight** ()
- void **update** (Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- Generator **newGenerator** (**GeneratorType** Generatortype) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

- `State getBestState ()`
- `void setBestState (State besState)`
- `StopExecute getStopexecute ()`
- `int getCountMax ()`
- `void setCountMax (int countMax)`
- `void setStopexecute (StopExecute stopexecute)`
- `UpdateParameter getUpdateparameter ()`
- `void setUpdateparameter (UpdateParameter updateparameter)`
- `Problem getProblem ()`
- `void setProblem (Problem problem)`
- `ArrayList< String > getListKey ()`
- `void initializeGenerators () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `void initialize () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
- `int getCountCurrent ()`
- `void setCountCurrent (int countCurrent)`
- `double getThreshold ()`
- `void setThreshold (double threshold)`
- `void calculateOffLinePerformance (float sumMax, int countOff)`
- `void updateRef (GeneratorType generatorType)`
- `void updateRefMultiG ()`
- `void updateRefGenerator (Generator generator)`

Métodos públicos estáticos

- `static Strategy getStrategy ()`
- `static void destroyExecute ()`

Atributos públicos

- `SortedMap< GeneratorType, Generator > mapGenerators`
- `Generator generator`
- `double threshold`
- `ArrayList< State > listStates`
- `ArrayList< State > listBest`
- `List< State > listRefPoblacFinal = new ArrayList<State> ()`
- `Dominance notDominated`
- `boolean saveListStates`
- `boolean saveListBestStates`
- `boolean saveFreneParetoMonoObjetivo`
- `boolean calculateTime`
- `float[] listOfflineError = new float[100]`
- `int countPeriodChange = 0`
- `int countChange = 0`

Atributos públicos estáticos

- `static long timeExecute`

Métodos privados

- [Strategy \(\)](#)

Atributos privados

- [State bestState](#)
- [Problem problem](#)
- [StopExecute stopexecute](#)
- [UpdateParameter updateparameter](#)
- [IFFactoryGenerator ifFactoryGenerator](#)
- [int countCurrent](#)
- [int countMax](#)
- [State initialState](#)
- [int countPeriodo](#)
- [int periodo](#)

Atributos estáticos privados

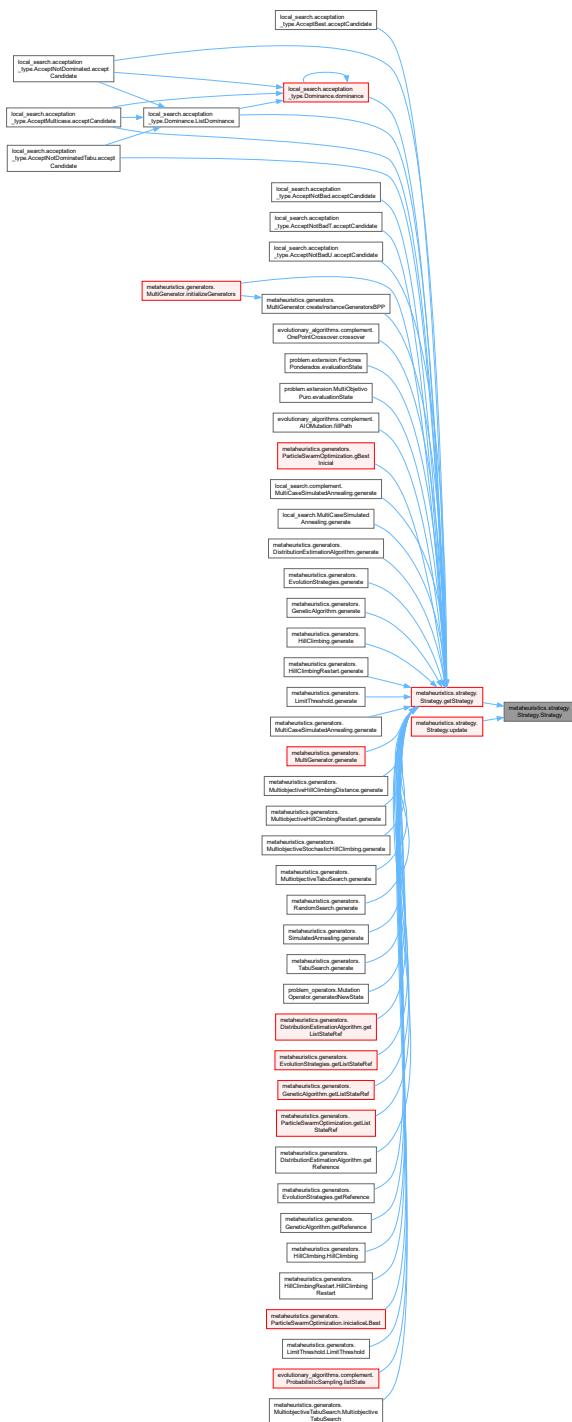
- [static Strategy strategy = null](#)

6.99.1. Documentación de constructores y destructores

6.99.1.1. Strategy()

```
metaheuristics.strategy.Strategy.Strategy () [private]
```

Gráfico de llamadas a esta función:

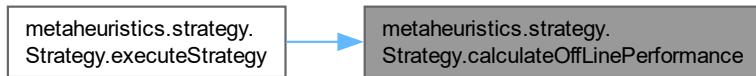


6.99.2. Documentación de funciones miembro

6.99.2.1. calculateOffLinePerformance()

```
void metaheuristics.strategy.Strategy.calculateOffLinePerformance (
    float sumMax,
    int countOff)
```

Gráfico de llamadas a esta función:



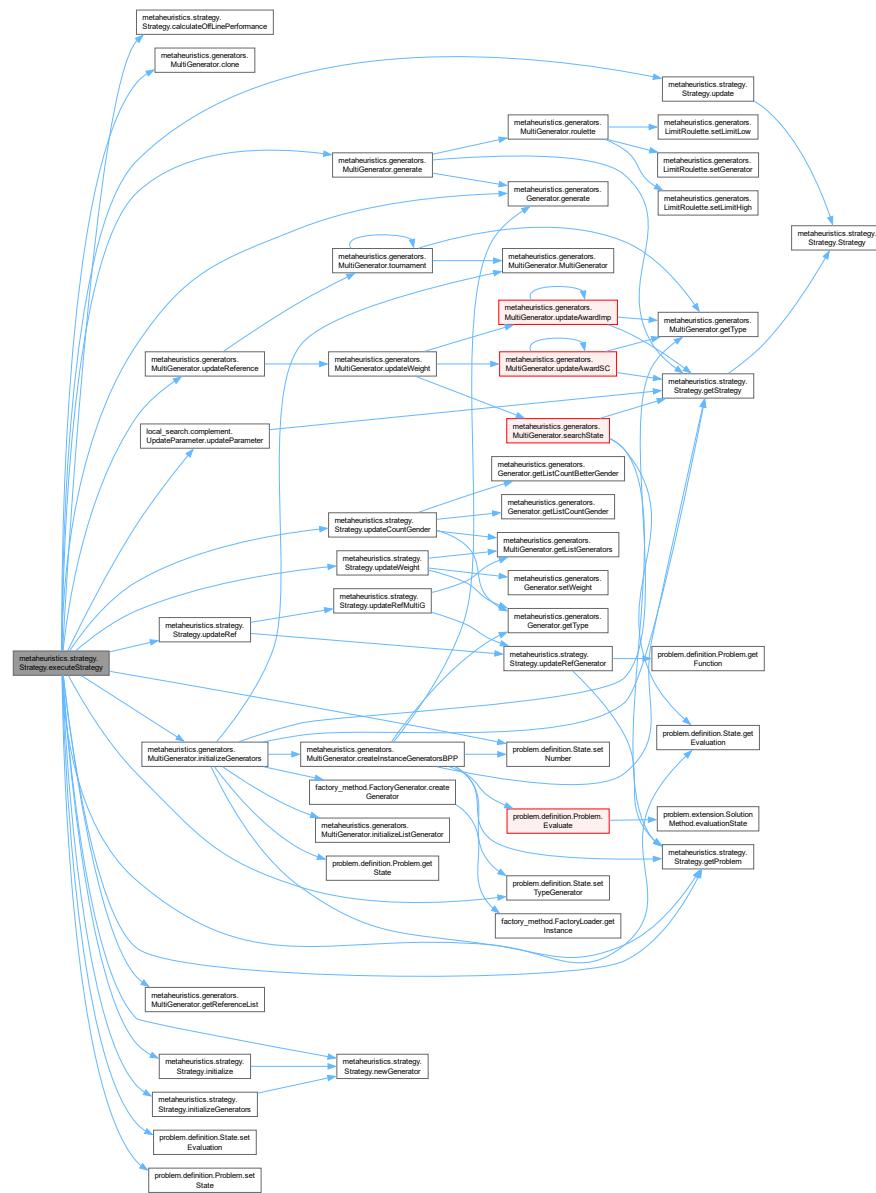
6.99.2.2. **destroyExecute()**

```
void metaheuristics.strategy.Strategy.destroyExecute () [static]
```

6.99.2.3. **executeStrategy()**

```
void metaheuristics.strategy.Strategy.executeStrategy (\n    int countmaxIterations,\n    int countIterationsChange,\n    int operatornumber,\n    GeneratorType generatorType) throws IllegalArgumentException, SecurityException,\nClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,\nNoSuchMethodException
```

Gráfico de llamadas de esta función:



6.99.2.4. getBestState()

```
State metaheuristics.strategy.Strategy.getBestState ()
```

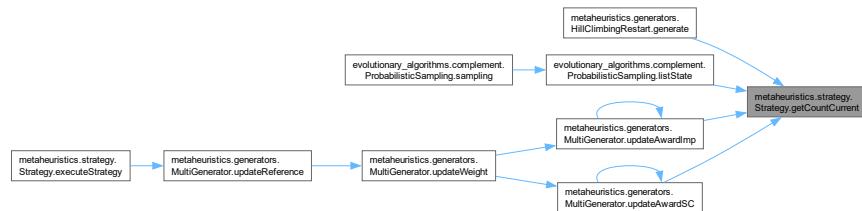
Gráfico de llamadas a esta función:



6.99.2.5. getCountCurrent()

```
int metaheuristics.strategy.Strategy.getCountCurrent ()
```

Gráfico de llamadas a esta función:



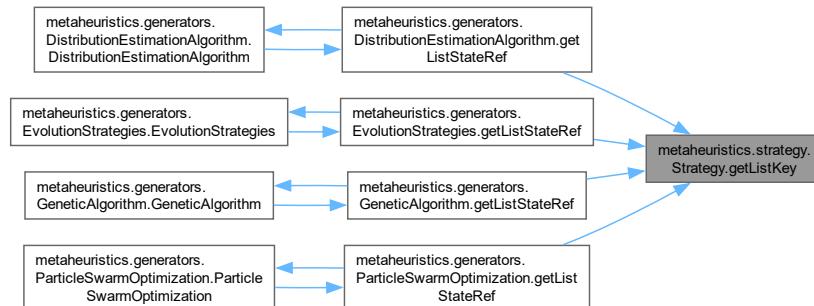
6.99.2.6. getCountMax()

```
int metaheuristics.strategy.Strategy.getCountMax ()
```

6.99.2.7. getListKey()

```
ArrayList< String > metaheuristics.strategy.Strategy.getListKey ()
```

Gráfico de llamadas a esta función:



6.99.2.8. getProblem()

```
Problem metaheuristics.strategy.Strategy.getProblem ()
```

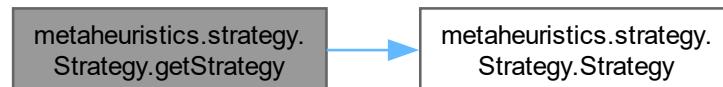
6.99.2.9. getStopexecute()

```
StopExecute metaheuristics.strategy.Strategy.getStopexecute ()
```

6.99.2.10. `getStrategy()`

```
Strategy metaheuristics.strategy.Strategy.getStrategy () [static]
```

Gráfico de llamadas de esta función:



6.99.2.11. `getThreshold()`

```
double metaheuristics.strategy.Strategy.getThreshold ()
```

Gráfico de llamadas a esta función:



6.99.2.12. `getUpdateparameter()`

```
UpdateParameter metaheuristics.strategy.Strategy.getUpdateparameter ()
```

6.99.2.13. `initialize()`

```
void metaheuristics.strategy.Strategy.initialize () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Gráfico de llamadas de esta función:

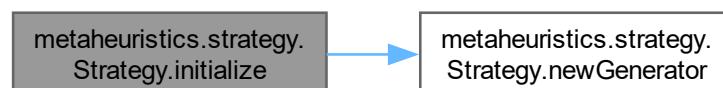


Gráfico de llamadas a esta función:



6.99.2.14. initializeGenerators()

```
void metaheuristics.strategy.Strategy.initializeGenerators () throws IllegalArgumentException,
SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException,
InvocationTargetException, NoSuchMethodException
```

Gráfico de llamadas de esta función:



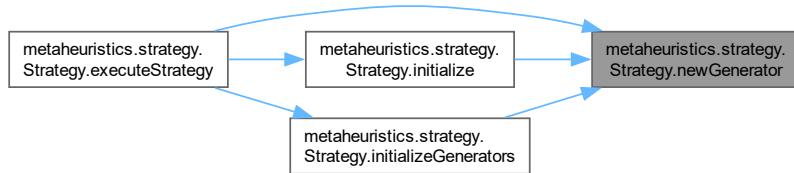
Gráfico de llamadas a esta función:



6.99.2.15. newGenerator()

```
Generator metaheuristics.strategy.Strategy.newGenerator (
    GeneratorType Generatortype) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Gráfico de llamadas a esta función:



6.99.2.16. setBestState()

```
void metaheuristics.strategy.Strategy.setBestState (
    State bestState)
```

6.99.2.17. setCountCurrent()

```
void metaheuristics.strategy.Strategy.setCountCurrent (
    int countCurrent)
```

6.99.2.18. setCountMax()

```
void metaheuristics.strategy.Strategy.setCountMax (
    int countMax)
```

6.99.2.19. setProblem()

```
void metaheuristics.strategy.Strategy.setProblem (
    Problem problem)
```

6.99.2.20. setStopexecute()

```
void metaheuristics.strategy.Strategy.setStopexecute (
    StopExecute stopexecute)
```

6.99.2.21. setThreshold()

```
void metaheuristics.strategy.Strategy.setThreshold (
    double threshold)
```

6.99.2.22. `setUpdtparameter()`

```
void metaheuristics.strategy.Strategy.setUpdtparameter (
    UpdateParameter updateparameter)
```

6.99.2.23. `update()`

```
void metaheuristics.strategy.Strategy.update (
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Gráfico de llamadas de esta función:

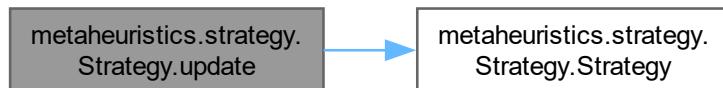
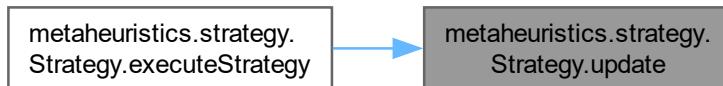


Gráfico de llamadas a esta función:



6.99.2.24. `updateCountGender()`

```
void metaheuristics.strategy.Strategy.updateCountGender ()
```

Gráfico de llamadas de esta función:

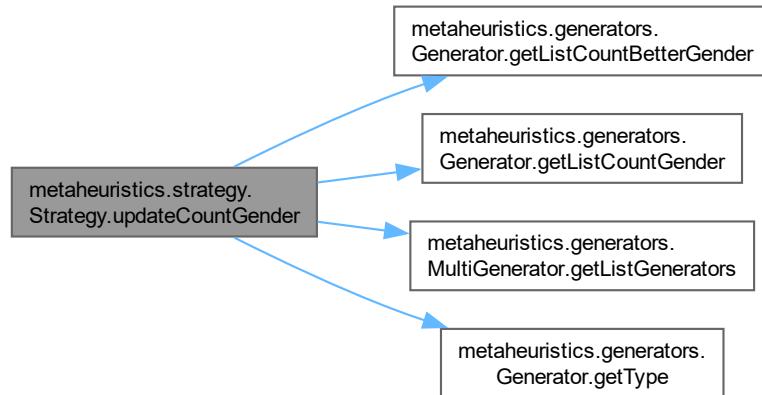


Gráfico de llamadas a esta función:



6.99.2.25. updateRef()

```
void metaheuristics.strategy.Strategy.updateRef (
    GeneratorType generatorType)
```

Gráfico de llamadas de esta función:

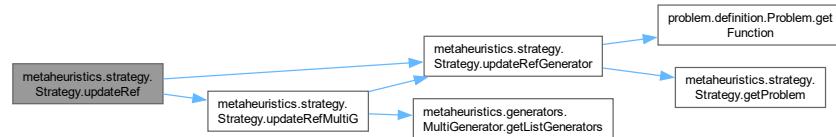
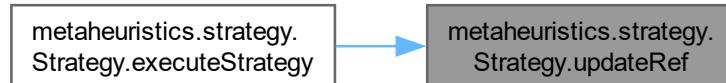


Gráfico de llamadas a esta función:



6.99.2.26. updateRefGenerator()

```
void metaheuristics.strategy.Strategy.updateRefGenerator (
    Generator generator)
```

Gráfico de llamadas de esta función:

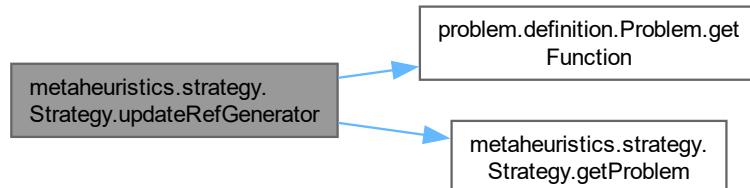


Gráfico de llamadas a esta función:



6.99.2.27. updateRefMultiG()

```
void metaheuristics.strategy.Strategy.updateRefMultiG ()
```

Gráfico de llamadas de esta función:

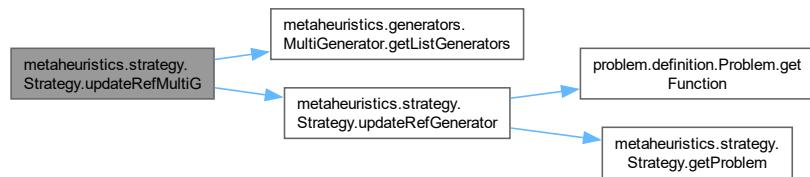
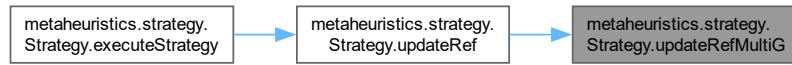


Gráfico de llamadas a esta función:



6.99.2.28. updateWeight()

```
void metaheuristics.strategy.Strategy.updateWeight ()
```

Gráfico de llamadas de esta función:

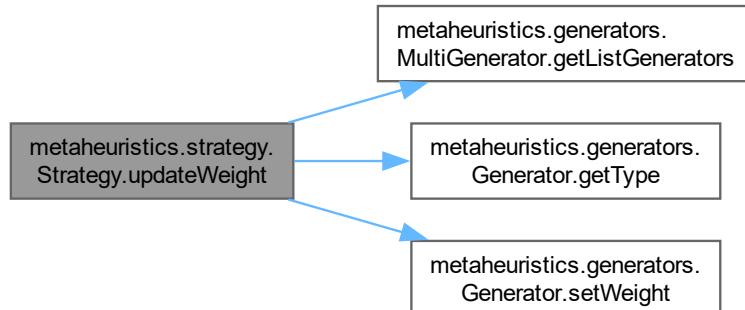


Gráfico de llamadas a esta función:



6.99.3. Documentación de datos miembro

6.99.3.1. bestState

```
State metaheuristics.strategy.Strategy.bestState [private]
```

6.99.3.2. calculateTime

```
boolean metaheuristics.strategy.Strategy.calculateTime
```

6.99.3.3. countChange

```
int metaheuristics.strategy.Strategy.countChange = 0
```

6.99.3.4. countCurrent

```
int metaheuristics.strategy.Strategy.countCurrent [private]
```

6.99.3.5. countMax

```
int metaheuristics.strategy.Strategy.countMax [private]
```

6.99.3.6. countPeriodChange

```
int metaheuristics.strategy.Strategy.countPeriodChange = 0
```

6.99.3.7. countPeriodo

```
int metaheuristics.strategy.Strategy.countPeriodo [private]
```

6.99.3.8. generator

```
Generator metaheuristics.strategy.Strategy.generator
```

6.99.3.9. ifFactoryGenerator

```
IFFactoryGenerator metaheuristics.strategy.Strategy.ifFactoryGenerator [private]
```

6.99.3.10. initialState

```
State metaheuristics.strategy.Strategy.initialState [private]
```

6.99.3.11. listBest

```
ArrayList<State> metaheuristics.strategy.Strategy.listBest
```

6.99.3.12. listOfflineError

```
float [] metaheuristics.strategy.Strategy.listOfflineError = new float[100]
```

6.99.3.13. listRefPoblacFinal

```
List<State> metaheuristics.strategy.Strategy.listRefPoblacFinal = new ArrayList<State> ()
```

6.99.3.14. listStates

```
ArrayList<State> metaheuristics.strategy.Strategy.listStates
```

6.99.3.15. mapGenerators

```
SortedMap<GeneratorType, Generator> metaheuristics.strategy.Strategy.mapGenerators
```

6.99.3.16. notDominated

```
Dominance metaheuristics.strategy.Strategy.notDominated
```

6.99.3.17. periodo

```
int metaheuristics.strategy.Strategy.periodo [private]
```

6.99.3.18. problem

```
Problem metaheuristics.strategy.Strategy.problem [private]
```

6.99.3.19. saveFreneParetoMonoObjetivo

```
boolean metaheuristics.strategy.Strategy.saveFreneParetoMonoObjetivo
```

6.99.3.20. saveListBestStates

```
boolean metaheuristics.strategy.Strategy.saveListBestStates
```

6.99.3.21. saveListStates

```
boolean metaheuristics.strategy.Strategy.saveListStates
```

6.99.3.22. stopexecute

```
StopExecute metaheuristics.strategy.Strategy.stopexecute [private]
```

6.99.3.23. strategy

```
Strategy metaheuristics.strategy.Strategy.strategy = null [static], [private]
```

6.99.3.24. threshold

```
double metaheuristics.strategy.Strategy.threshold
```

6.99.3.25. timeExecute

```
long metaheuristics.strategy.Strategy.timeExecute [static]
```

6.99.3.26. updateparameter

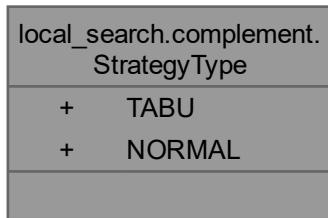
```
UpdateParameter metaheuristics.strategy.Strategy.updateparameter [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/strategy/Strategy.java](#)

6.100. Referencia de la enumeración local_search.complement.StrATEGYType

Diagrama de colaboración de local_search.complement.StrATEGYType:



Atributos públicos

- [TABU](#)
- [NORMAL](#)

6.100.1. Documentación de datos miembro

6.100.1.1. NORMAL

`local_search.complement.StrATEGYType.NORMAL`

6.100.1.2. TABU

`local_search.complement.StrATEGYType.TABU`

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/local_search/complement/StrATEGYType.java](#)

6.101. Referencia de la clase metaheuristics.generators.TabuSearch

Diagrama de herencia de metaheuristics.generators.TabuSearch

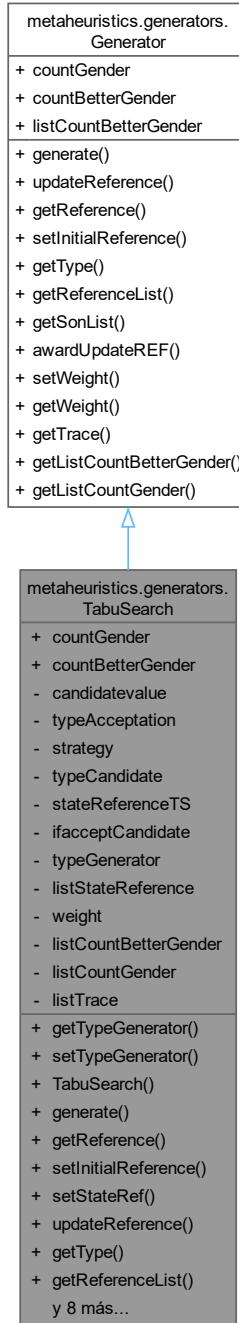
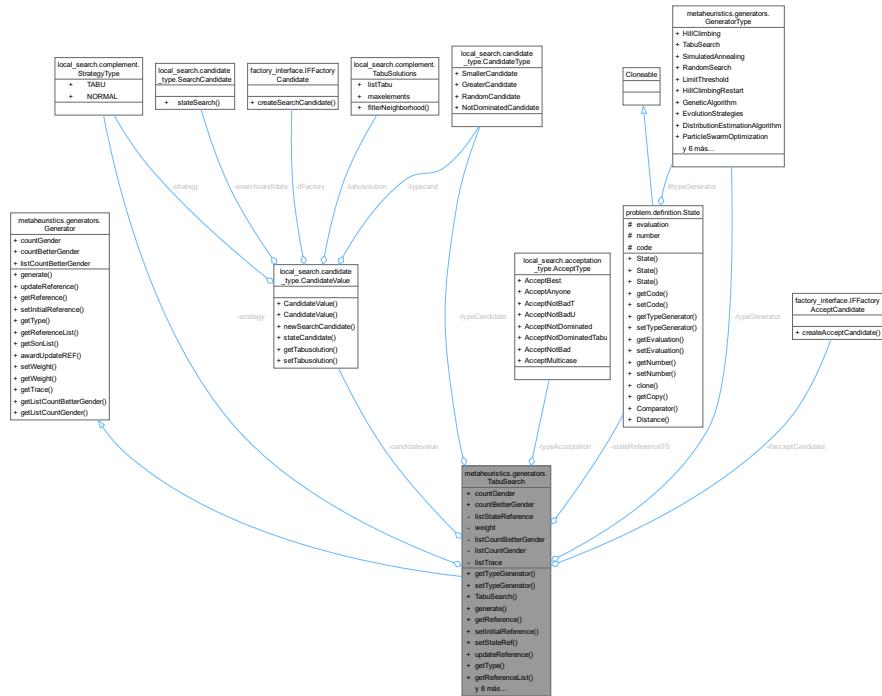


Diagrama de colaboración de metaheuristics.generators.TabuSearch:



Métodos públicos

- `GeneratorType getTypeGenerator ()`
 - `void setTypeGenerator (GeneratorType typeGenerator)`
 - `TabuSearch ()`
 - `State generate (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
 - `State getReference ()`
 - `void setInitialReference (State stateInitialRef)`
 - `void setStateRef (State stateRef)`
 - `void updateReference (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException`
 - `GeneratorType getType ()`
 - `List< State > getReferenceList ()`
 - `List< State > getSonList ()`
 - `void setTypeCandidate (CandidateType typeCandidate)`
 - `boolean awardUpdateREF (State stateCandidate)`
 - `float getWeight ()`
 - `void setWeight (float weight)`
 - `int[] getListCountBetterGender ()`
 - `int[] getListCountGender ()`
 - `float[] getTrace ()`

Atributos públicos estáticos

- static int countGender = 0
 - static int countBetterGender = 0

Atributos privados

- CandidateValue candidatevalue
- AcceptType typeAcceptation
- StrategyType strategy
- CandidateType typeCandidate
- State stateReferenceTS
- IFFactoryAcceptCandidate ifacceptCandidate
- GeneratorType typeGenerator
- List< State > listStateReference = new ArrayList<State>()
- float weight
- int[] listCountBetterGender = new int[10]
- int[] listCountGender = new int[10]
- float[] listTrace = new float[1200000]

Otros miembros heredados

Atributos públicos heredados de [metaheuristics.generators.Generator](#)

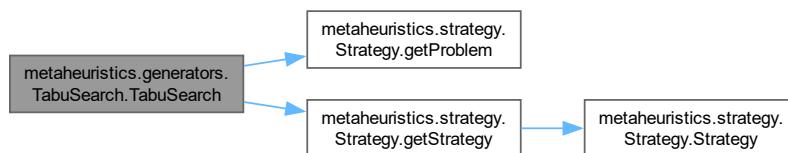
- int countGender
- int countBetterGender
- int[] listCountBetterGender

6.101.1. Documentación de constructores y destructores

6.101.1.1. TabuSearch()

```
metaheuristics.generators.TabuSearch.TabuSearch ()
```

Gráfico de llamadas de esta función:



6.101.2. Documentación de funciones miembro

6.101.2.1. awardUpdateREF()

```
boolean metaheuristics.generators.TabuSearch.awardUpdateREF (
    State stateCandidate)
```

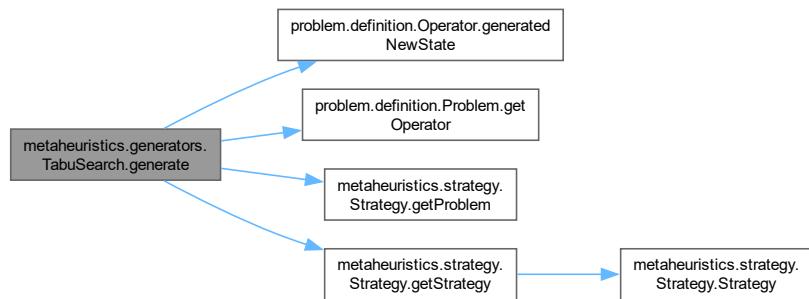
Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.2. generate()

```
State metaheuristics.generators.TabuSearch.generate (
    Integer operatornumber) throws IllegalArgumentException, SecurityException, Class←
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.101.2.3. getListCountBetterGender()

```
int[] metaheuristics.generators.TabuSearch.getListCountBetterGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.4. getListCountGender()

```
int[] metaheuristics.generators.TabuSearch.getListCountGender ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.5. getReference()

```
State metaheuristics.generators.TabuSearch.getReference ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.6. getReferenceList()

```
List< State > metaheuristics.generators.TabuSearch.getReferenceList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.7. `getSonList()`

```
List< State > metaheuristics.generators.TabuSearch.getSonList ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.8. `getTrace()`

```
float[ ] metaheuristics.generators.TabuSearch.getTrace ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.9. `getType()`

```
GeneratorType metaheuristics.generators.TabuSearch.getType ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.10. `getTypeGenerator()`

```
GeneratorType metaheuristics.generators.TabuSearch.getTypeGenerator ()
```

6.101.2.11. `getWeight()`

```
float metaheuristics.generators.TabuSearch.getWeight ()
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.12. `setInitialReference()`

```
void metaheuristics.generators.TabuSearch.setInitialReference (
    State stateInitialRef)
```

Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.13. `setStateRef()`

```
void metaheuristics.generators.TabuSearch.setStateRef (
    State stateRef)
```

6.101.2.14. `setTypeCandidate()`

```
void metaheuristics.generators.TabuSearch.setTypeCandidate (
    CandidateType typeCandidate)
```

6.101.2.15. `setTypeGenerator()`

```
void metaheuristics.generators.TabuSearch.setTypeGenerator (
    GeneratorType typeGenerator)
```

6.101.2.16. `setWeight()`

```
void metaheuristics.generators.TabuSearch.setWeight (
    float weight)
```

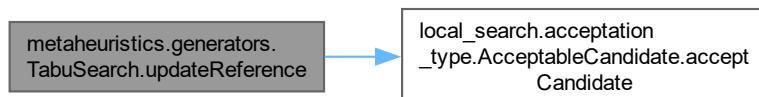
Reimplementado de [metaheuristics.generators.Generator](#).

6.101.2.17. `updateReference()`

```
void metaheuristics.generators.TabuSearch.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Reimplementado de [metaheuristics.generators.Generator](#).

Gráfico de llamadas de esta función:



6.101.3. Documentación de datos miembro

6.101.3.1. `candidatevalue`

```
CandidateValue metaheuristics.generators.TabuSearch.candidatevalue [private]
```

6.101.3.2. `countBetterGender`

```
int metaheuristics.generators.TabuSearch.countBetterGender = 0 [static]
```

6.101.3.3. `countGender`

```
int metaheuristics.generators.TabuSearch.countGender = 0 [static]
```

6.101.3.4. ifacceptCandidate

```
IFFactoryAcceptCandidate metaheuristics.generators.TabuSearch.ifacceptCandidate [private]
```

6.101.3.5. listCountBetterGender

```
int [] metaheuristics.generators.TabuSearch.listCountBetterGender = new int[10] [private]
```

6.101.3.6. listCountGender

```
int [] metaheuristics.generators.TabuSearch.listCountGender = new int[10] [private]
```

6.101.3.7. listStateReference

```
List<State> metaheuristics.generators.TabuSearch.listStateReference = new ArrayList<State>()  
[private]
```

6.101.3.8. listTrace

```
float [] metaheuristics.generators.TabuSearch.listTrace = new float[1200000] [private]
```

6.101.3.9. stateReferenceTS

```
State metaheuristics.generators.TabuSearch.stateReferenceTS [private]
```

6.101.3.10. strategy

```
StrategyType metaheuristics.generators.TabuSearch.strategy [private]
```

6.101.3.11. typeAcceptation

```
AcceptType metaheuristics.generators.TabuSearch.typeAcceptation [private]
```

6.101.3.12. typeCandidate

```
CandidateType metaheuristics.generators.TabuSearch.typeCandidate [private]
```

6.101.3.13. typeGenerator

```
GeneratorType metaheuristics.generators.TabuSearch.typeGenerator [private]
```

6.101.3.14. weight

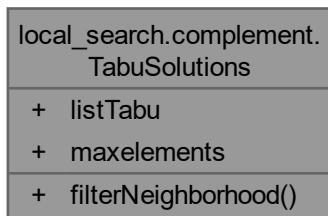
```
float metaheuristics.generators.TabuSearch.weight [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/metaheuristics/generators/TabuSearch.java](#)

6.102. Referencia de la clase local_search.complement.TabuSolutions

Diagrama de colaboración de local_search.complement.TabuSolutions:



Métodos públicos

- [List< State > filterNeighborhood \(List< State > listNeighborhood\) throws Exception](#)

Atributos públicos estáticos

- static [List< State > listTabu = new ArrayList<State>\(\)](#)
- static int [maxelements](#)

6.102.1. Documentación de funciones miembro

6.102.1.1. filterNeighborhood()

```
List< State > local_search.complement.TabuSolutions.filterNeighborhood (
    List< State > listNeighborhood) throws Exception
```

6.102.2. Documentación de datos miembro

6.102.2.1. listTabu

```
List<State> local_search.complement.TabuSolutions.listTabu = new ArrayList<State>() [static]
```

6.102.2.2. maxelements

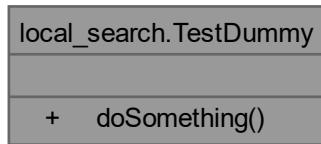
```
int local_search.complement.TabuSolutions.maxelements [static]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/complement/TabuSolutions.java](#)

6.103. Referencia de la clase local_search.TestDummy

Diagrama de colaboración de local_search.TestDummy:



Métodos públicos

- [void doSomething \(\)](#)

6.103.1. Documentación de funciones miembro

6.103.1.1. doSomething()

```
void local_search.TestDummy.doSomething ()
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/TestDummy.java](#)

6.104. Referencia de la clase evolutionary_algorithms.complement.TowPointsMutation

Diagrama de herencia de evolutionary_algorithms.complement.TowPointsMutation

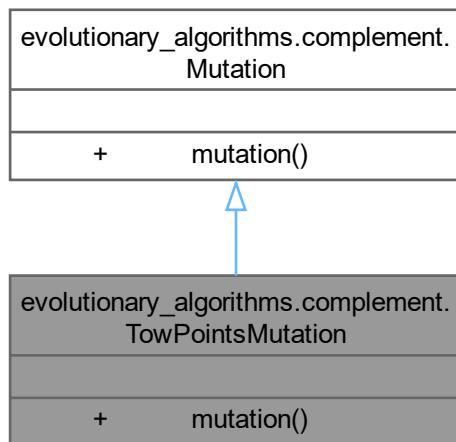
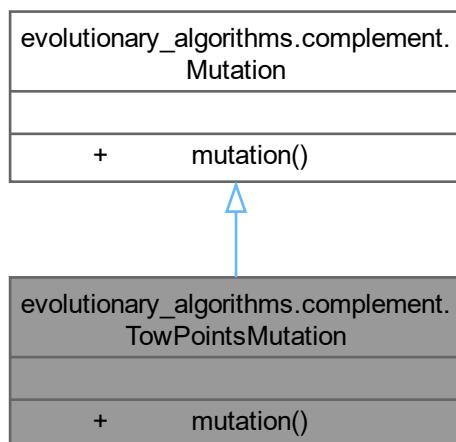


Diagrama de colaboración de `evolutionary_algorithms.complement.TowPointsMutation`:



Métodos públicos

- [State mutation \(State newwind, double PM\)](#)

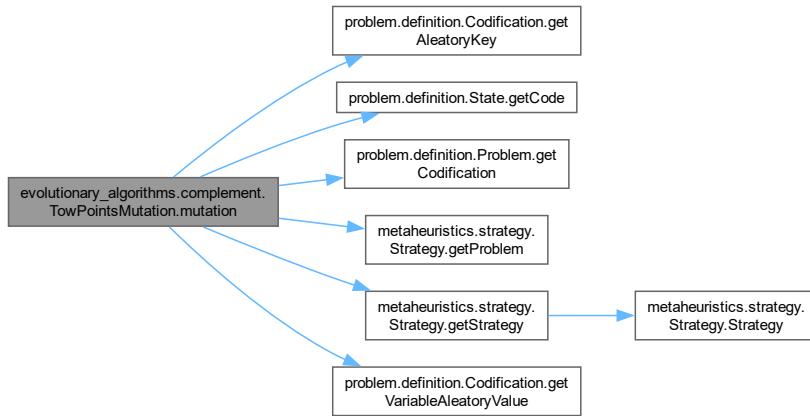
6.104.1. Documentación de funciones miembro

6.104.1.1. mutation()

```
State evolutionary_algorithms.complement.TowPointsMutation.mutation (
    State newind,
    double PM)
```

Reimplementado de [evolutionary_algorithms.complement.Mutation](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- src/main/java/evolutionary_algorithms/complement/TowPointsMutation.java

6.105. Referencia de la clase evolutionary_algorithms.complement.TruncationSelection

Diagrama de herencia de evolutionary_algorithms.complement.TruncationSelection

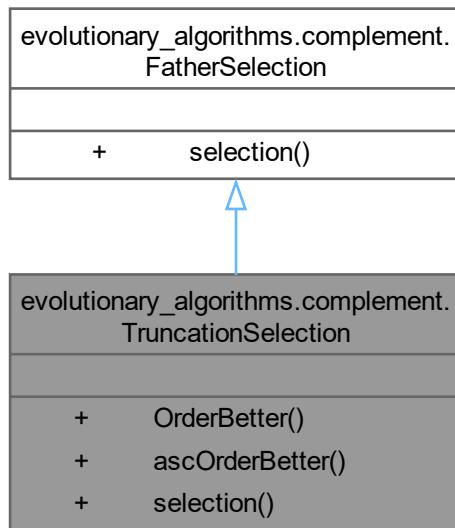
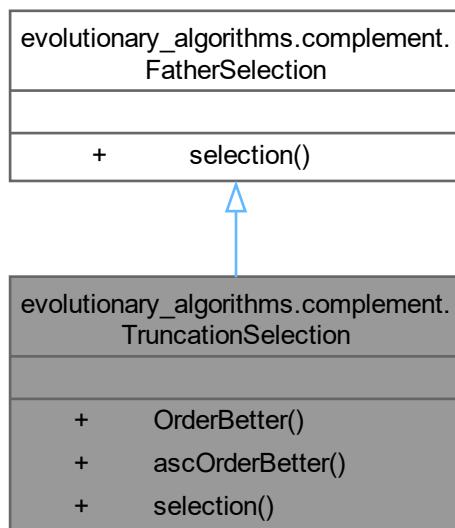


Diagrama de colaboración de evolutionary_algorithms.complement.TruncationSelection:



Métodos públicos

- List< State > OrderBetter (List< State > listState)
- List< State > ascOrderBetter (List< State > listState)
- List< State > selection (List< State > listState, int truncation)

6.105.1. Documentación de funciones miembro

6.105.1.1. ascOrderBetter()

```
List< State > evolutionary_algorithms.complement.TruncationSelection.ascOrderBetter (
    List< State > listState)
```

Gráfico de llamadas a esta función:



6.105.1.2. OrderBetter()

```
List< State > evolutionary_algorithms.complement.TruncationSelection.OrderBetter (
    List< State > listState)
```

Gráfico de llamadas a esta función:

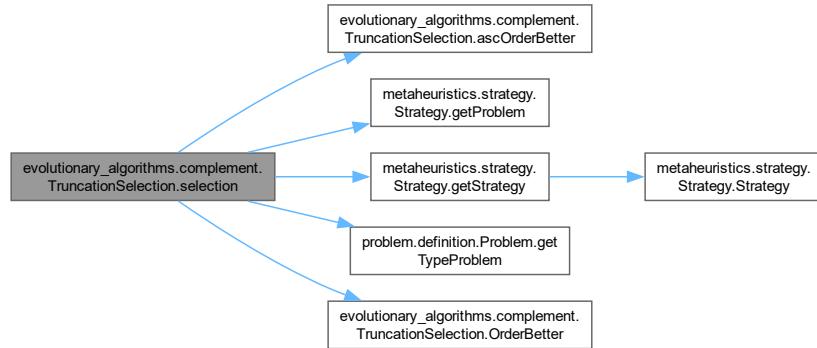


6.105.1.3. selection()

```
List< State > evolutionary_algorithms.complement.TruncationSelection.selection (
    List< State > listState,
    int truncation)
```

Reimplementado de [evolutionary_algorithms.complement.FatherSelection](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/TruncationSelection.java](#)

6.106. Referencia de la clase config.tspDynamic.TSPState

Diagrama de colaboración de config.tspDynamic.TSPState:

config.tspDynamic.TSPState
- value - idCity
+ getValue() + setValue() + getIdCity() + setIdCity()

Métodos públicos

- int [getValue \(\)](#)
- void [setValue \(int value\)](#)
- int [getIdCity \(\)](#)
- void [setIdCity \(int idCity\)](#)

Atributos privados

- int [value](#)
- int [idCity](#)

6.106.1. Documentación de funciones miembro

6.106.1.1. [getIdCity\(\)](#)

```
int config.tspDynamic.TSPState.getIdCity ()
```

6.106.1.2. [getValue\(\)](#)

```
int config.tspDynamic.TSPState.getValue ()
```

6.106.1.3. [setIdCity\(\)](#)

```
void config.tspDynamic.TSPState.setIdCity (
    int idCity)
```

6.106.1.4. [setValue\(\)](#)

```
void config.tspDynamic.TSPState.setValue (
    int value)
```

6.106.2. Documentación de datos miembro

6.106.2.1. [idCity](#)

```
int config.tspDynamic.TSPState.idCity [private]
```

6.106.2.2. [value](#)

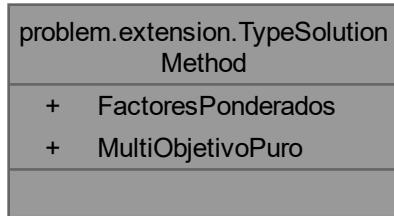
```
int config.tspDynamic.TSPState.value [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/config/tspDynamic/TSPState.java](#)

6.107. Referencia de la enumeración problem.extension.TypeSolutionMethod

Diagrama de colaboración de problem.extension.TypeSolutionMethod:



Atributos públicos

- FactoresPonderados
- MultiObjetivoPuro

6.107.1. Documentación de datos miembro

6.107.1.1. FactoresPonderados

problem.extension.TypeSolutionMethod.FactoresPonderados

6.107.1.2. MultiObjetivoPuro

problem.extension.TypeSolutionMethod.MultiObjetivoPuro

La documentación de esta enumeración está generada del siguiente archivo:

- src/main/java/problem/extension/[TypeSolutionMethod.java](#)

6.108. Referencia de la clase

evolutionary_algorithms.complement.UniformCrossover

Diagrama de herencia de evolutionary_algorithms.complement.UniformCrossover

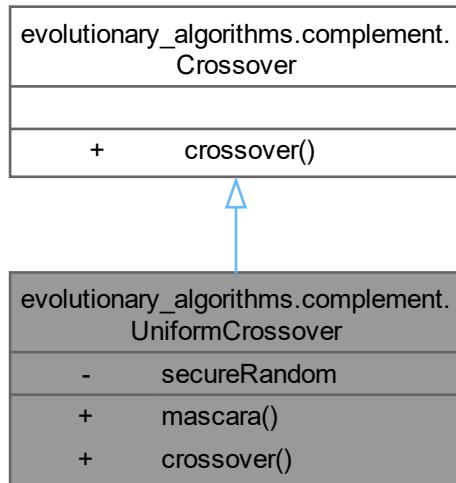
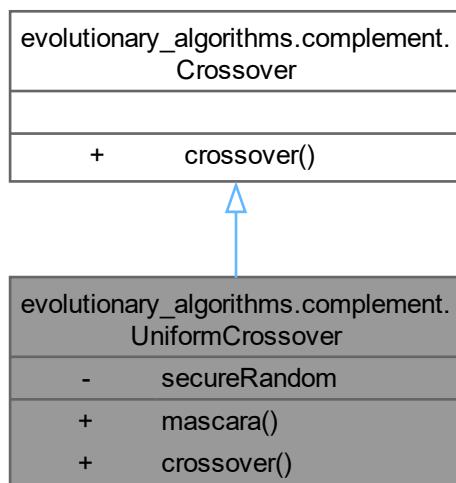


Diagrama de colaboración de `evolutionary_algorithms.complement.UniformCrossover`:



Métodos públicos

- `int[] mascara (int length)`
- `State crossover (State father1, State father2, double PC)`

Atributos estáticos privados

- `static final SecureRandom secureRandom = new SecureRandom()`

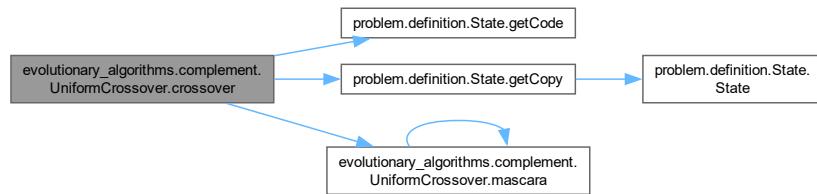
6.108.1. Documentación de funciones miembro

6.108.1.1. `crossover()`

```
State evolutionary_algorithms.complement.UniformCrossover.crossover (
    State father1,
    State father2,
    double PC)
```

Reimplementado de [evolutionary_algorithms.complement.Crossover](#).

Gráfico de llamadas de esta función:



6.108.1.2. `mascara()`

```
int[] evolutionary_algorithms.complement.UniformCrossover.mascara (
    int length)
```

Gráfico de llamadas de esta función:

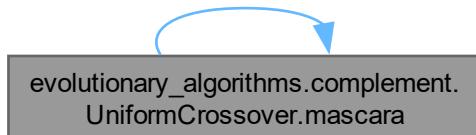


Gráfico de llamadas a esta función:



6.108.2. Documentación de datos miembro

6.108.2.1. secureRandom

```
final SecureRandom evolutionary_algorithms.complement.UniformCrossover.secureRandom = new
SecureRandom() [static], [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/evolutionary_algorithms/complement/UniformCrossover.java](#)

6.109. Referencia de la clase **evolutionary_algorithms.complement.Univariate**

Diagrama de herencia de `evolutionary_algorithms.complement.Univariate`

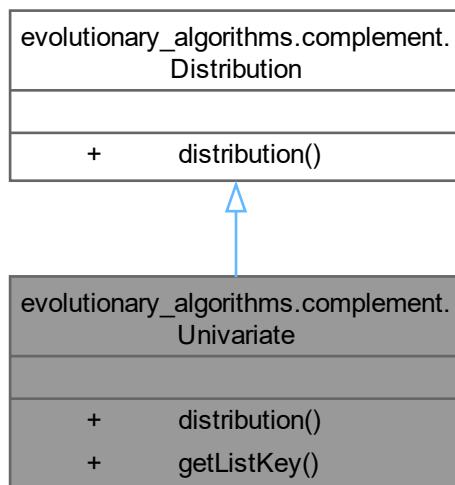
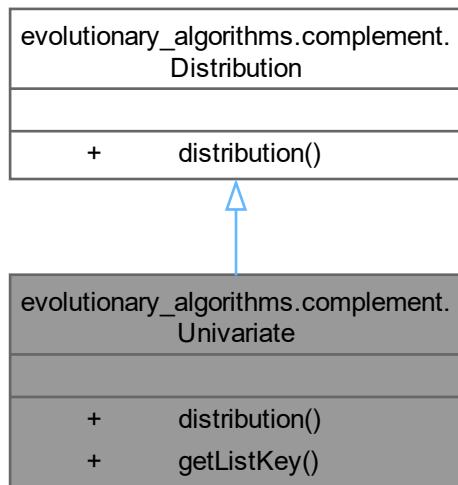


Diagrama de colaboración de `evolutionary_algorithms.complement.Univariate`:



Métodos públicos

- `List< Probability > distribution (List< State > fathers)`
- `List< String > getListKey () (SortedMap< String, Object > map)`

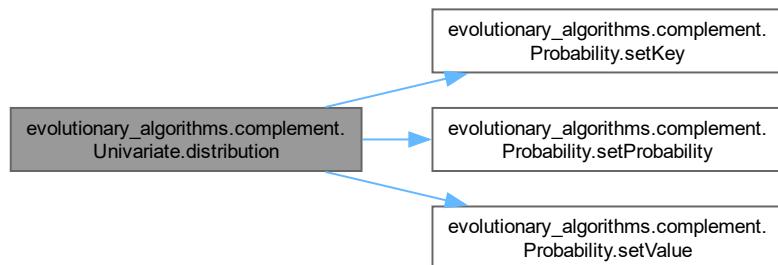
6.109.1. Documentación de funciones miembro

6.109.1.1. `distribution()`

```
List< Probability > evolutionary_algorithms.complement.Univariate.distribution (
    List< State > fathers)
```

Reimplementado de [evolutionary_algorithms.complement.Distribution](#).

Gráfico de llamadas de esta función:



6.109.1.2. getListKey()

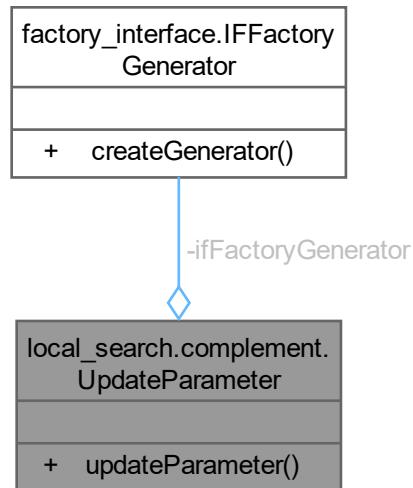
```
List< String > evolutionary_algorithms.complement.Univariate.getListKey (
    SortedMap< String, Object > map)
```

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/evolutionary_algorithms/complement/[Univariate.java](#)

6.110. Referencia de la clase local_search.complement.UpdateParameter

Diagrama de colaboración de local_search.complement.UpdateParameter:



Métodos públicos estáticos

- static Integer [updateParameter](#) (Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

Atributos estáticos privados

- static [IFFactoryGenerator](#) ifFactoryGenerator

6.110.1. Documentación de funciones miembro

6.110.1.1. updateParameter()

```
Integer local_search.complement.UpdateParameter.updateParameter (
    Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException [static]
```

Gráfico de llamadas de esta función:



Gráfico de llamadas a esta función:



6.110.2. Documentación de datos miembro

6.110.2.1. ifFactoryGenerator

```
IFFactoryGenerator local_search.complement.UpdateParameter.ifFactoryGenerator [static], [private]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/local_search/complement/UpdateParameter.java](#)

Capítulo 7

Documentación de archivos

7.1. Referencia del archivo

`src/main/java/config/tspDynamic/TSPState.java`

Clases

- class `config.tspDynamic.TSPState`

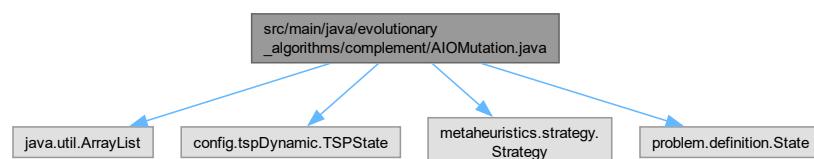
Paquetes

- package `config.tspDynamic`

7.2. Referencia del archivo `src/main/java/evolutionary_algorithms/complement/AIOMutation.java`

```
import java.util.ArrayList;
import config.tspDynamic.TSPState;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
```

Gráfico de dependencias incluidas en `AIOMutation.java`:



Clases

- class `evolutionary_algorithms.complement.AIOMutation`

Paquetes

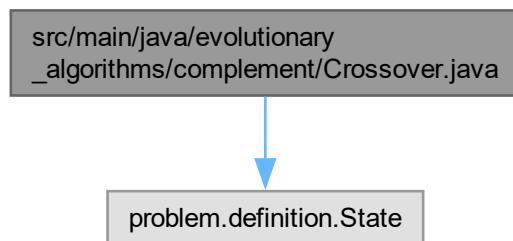
- package [evolutionary_algorithms.complement](#)

7.3. Referencia del archivo

src/main/java/evolutionary_algorithms/complement/Crossover.java

```
import problem.definition.State;
```

Gráfico de dependencias incluidas en Crossover.java:



Clases

- class [evolutionary_algorithms.complement.Crossover](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.4. Referencia del archivo **src/main/java/evolutionary_← algorithms/complement/CrossoverType.java**

Clases

- enum [evolutionary_algorithms.complement.CrossoverType](#)

Paquetes

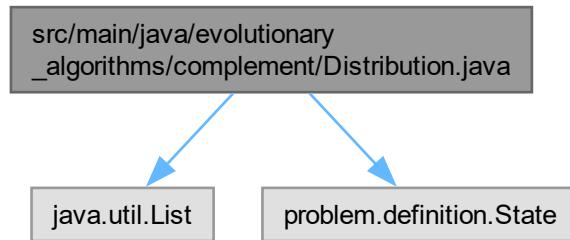
- package [evolutionary_algorithms.complement](#)

7.5. Referencia del archivo

`src/main/java/evolutionary_algorithms/complement/Distribution.java`

```
import java.util.List;
import problem.definition.State;
```

Gráfico de dependencias incluidas en Distribution.java:



Clases

- class [evolutionary_algorithms.complement.Distribution](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.6. Referencia del archivo `src/main/java/evolutionary_algorithms/complement/DistributionType.java`

Clases

- enum [evolutionary_algorithms.complement.DistributionType](#)

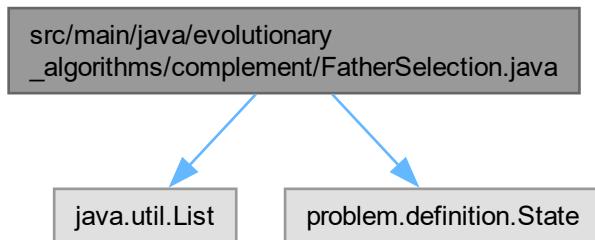
Paquetes

- package [evolutionary_algorithms.complement](#)

7.7. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/FatherSelection.java

```
import java.util.List;
import problem.definition.State;
```

Gráfico de dependencias incluidas en FatherSelection.java:



Clases

- class [evolutionary_algorithms.complement.FatherSelection](#)

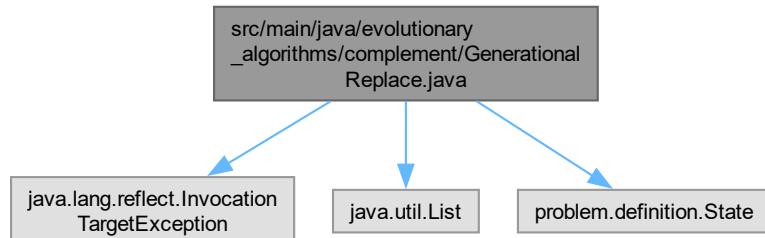
Paquetes

- package [evolutionary_algorithms.complement](#)

7.8. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/GenerationalReplace.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.List;
import problem.definition.State;
```

Gráfico de dependencias incluidas en GenerationalReplace.java:



Clases

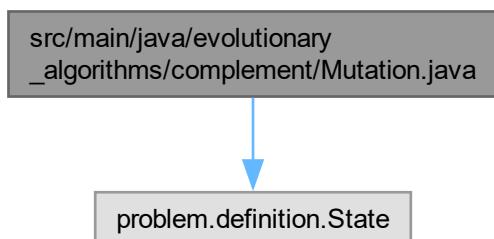
- class `evolutionary_algorithms.complement.GenerationalReplace`

Paquetes

- package `evolutionary_algorithms.complement`

7.9. Referencia del archivo `src/main/java/evolutionary_algorithms/complement/Mutation.java`

`import problem.definition.State;`
Gráfico de dependencias incluidas en `Mutation.java`:

**Clases**

- class `evolutionary_algorithms.complement.Mutation`

Paquetes

- package `evolutionary_algorithms.complement`

7.10. Referencia del archivo `src/main/java/evolutionary_algorithms/complement/MutationType.java`

Clases

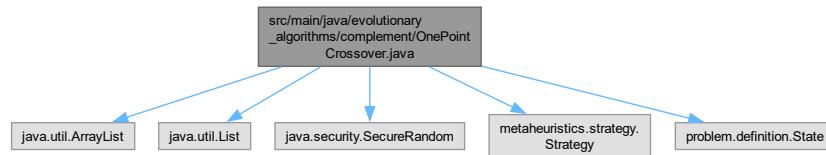
- enum `evolutionary_algorithms.complement.MutationType`

Paquetes

- package `evolutionary_algorithms.complement`

7.11. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/OnePointCrossover.java

```
import java.util.ArrayList;
import java.util.List;
import java.security.SecureRandom;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
Gráfico de dependencias incluidas en OnePointCrossover.java:
```



Clases

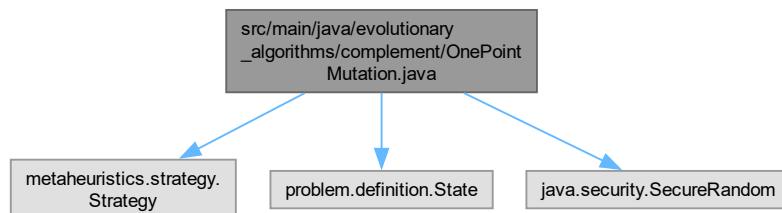
- class [evolutionary_algorithms.complement.OnePointCrossover](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.12. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/OnePointMutation.java

```
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import java.security.SecureRandom;
Gráfico de dependencias incluidas en OnePointMutation.java:
```



Clases

- class [evolutionary_algorithms.complement.OnePointMutation](#)

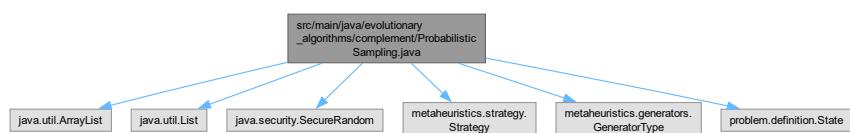
Paquetes

- package [evolutionary_algorithms.complement](#)

7.13. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/ProbabilisticSampling.java

```
import java.util.ArrayList;
import java.util.List;
import java.security.SecureRandom;
import metaheuristics.strategy.Strategy;
import metaheuristics.generators.GeneratorType;
import problem.definition.State;
```

Gráfico de dependencias incluidas en ProbabilisticSampling.java:



Clases

- class [evolutionary_algorithms.complement.ProbabilisticSampling](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.14. Referencia del archivo src/main/java/evolutionary_algorithms/complement/Probability.java

Clases

- class [evolutionary_algorithms.complement.Probability](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.15. Referencia del archivo

src/main/java/evolutionary_algorithms/complement/Range.java

Clases

- class [evolutionary_algorithms.complement.Range](#)

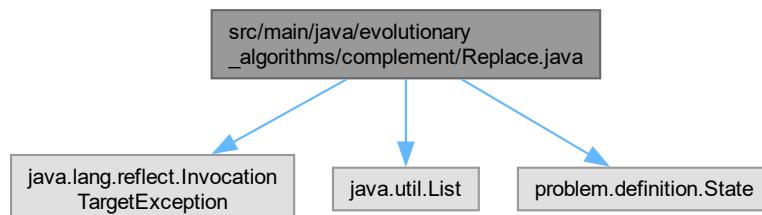
Paquetes

- package [evolutionary_algorithms.complement](#)

7.16. Referencia del archivo

src/main/java/evolutionary_algorithms/complement/Replace.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.List;
import problem.definition.State;
Gráfico de dependencias incluidas en Replace.java:
```



Clases

- class [evolutionary_algorithms.complement.Replace](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.17. Referencia del archivo src/main/java/evolutionary_algorithms/complement/ReplaceType.java

Clases

- enum [evolutionary_algorithms.complement.ReplaceType](#)

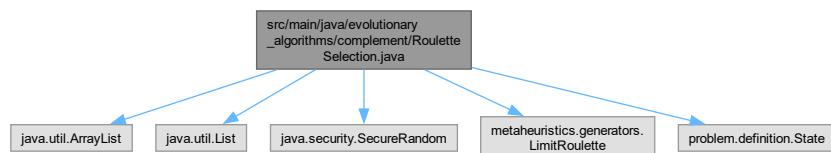
Paquetes

- package [evolutionary_algorithms.complement](#)

7.18. Referencia del archivo `src/main/java/evolutionary_algorithms/complement/RouletteSelection.java`

```
import java.util.ArrayList;
import java.util.List;
import java.security.SecureRandom;
import metaheuristics.generators.LimitRoulette;
import problem.definition.State;
```

Gráfico de dependencias incluidas en RouletteSelection.java:



Clases

- class [evolutionary_algorithms.complement.RouletteSelection](#)

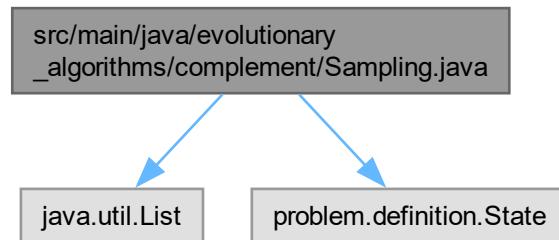
Paquetes

- package [evolutionary_algorithms.complement](#)

7.19. Referencia del archivo `src/main/java/evolutionary_algorithms/complement/Sampling.java`

```
import java.util.List;
import problem.definition.State;
```

Gráfico de dependencias incluidas en Sampling.java:



Clases

- class [evolutionary_algorithms.complement.Sampling](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.20. Referencia del archivo [src/main/java/evolutionary_algorithms/complement/SamplingType.java](#)

Clases

- enum [evolutionary_algorithms.complement.SamplingType](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.21. Referencia del archivo [src/main/java/evolutionary_algorithms/complement/SelectionType.java](#)

Clases

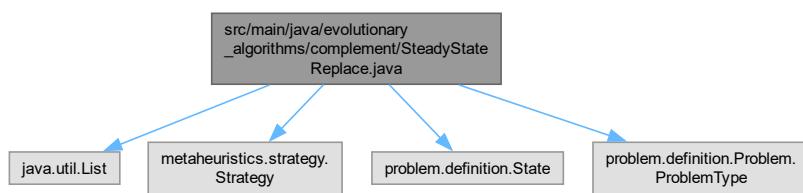
- enum [evolutionary_algorithms.complement.SelectionType](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.22. Referencia del archivo [src/main/java/evolutionary_algorithms/complement/SteadyStateReplace.java](#)

```
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
Gráfico de dependencias incluidas en SteadyStateReplace.java:
```



Clases

- class [evolutionary_algorithms.complement.SteadyStateReplace](#)

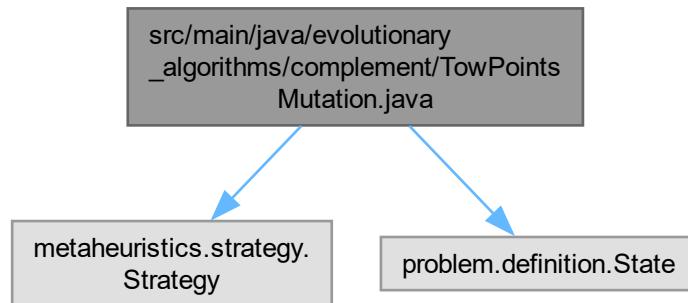
Paquetes

- package [evolutionary_algorithms.complement](#)

7.23. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/TowPointsMutation.java

```
import metaheuristics.strategy.Strategy;
import problem.definition.State;
```

Gráfico de dependencias incluidas en TowPointsMutation.java:



Clases

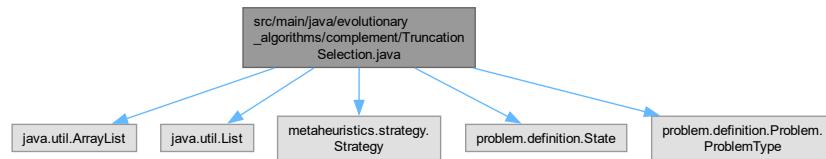
- class [evolutionary_algorithms.complement.TowPointsMutation](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.24. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/TruncationSelection.java

```
import java.util.ArrayList;
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
Gráfico de dependencias incluidas en TruncationSelection.java:
```



Clases

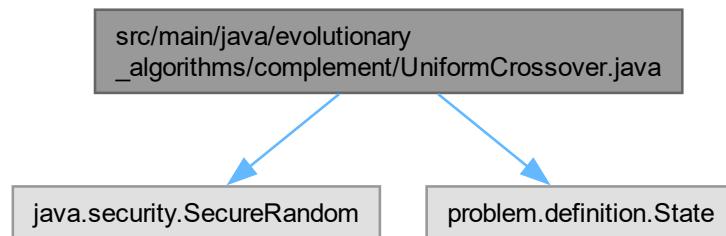
- class [evolutionary_algorithms.complement.TruncationSelection](#)

Paquetes

- package [evolutionary_algorithms.complement](#)

7.25. Referencia del archivo src/main/java/evolutionary_← algorithms/complement/UniformCrossover.java

```
import java.security.SecureRandom;
import problem.definition.State;
Gráfico de dependencias incluidas en UniformCrossover.java:
```



Clases

- class `evolutionary_algorithms.complement.UniformCrossover`

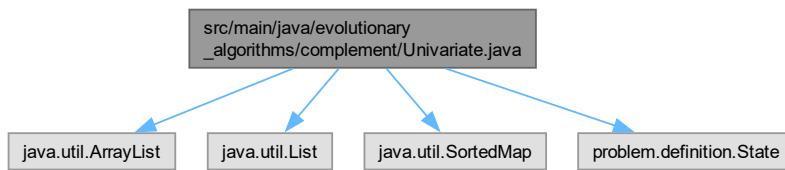
Paquetes

- package `evolutionary_algorithms.complement`

7.26. Referencia del archivo**`src/main/java/evolutionary_algorithms/complement/Univariate.java`**

```
import java.util.ArrayList;
import java.util.List;
import java.util.SortedMap;
import problem.definition.State;
```

Gráfico de dependencias incluidas en `Univariate.java`:

**Clases**

- class `evolutionary_algorithms.complement.Univariate`

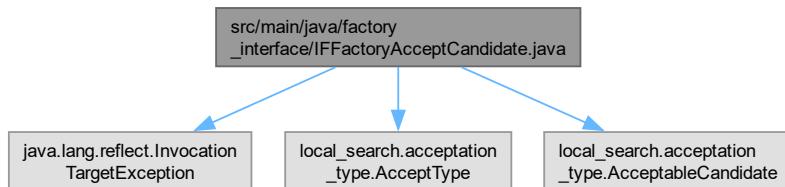
Paquetes

- package `evolutionary_algorithms.complement`

7.27. Referencia del archivo**`src/main/java/factory_interface/IFFactoryAcceptCandidate.java`**

```
import java.lang.reflect.InvocationTargetException;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
```

Gráfico de dependencias incluidas en `IFFactoryAcceptCandidate.java`:



Clases

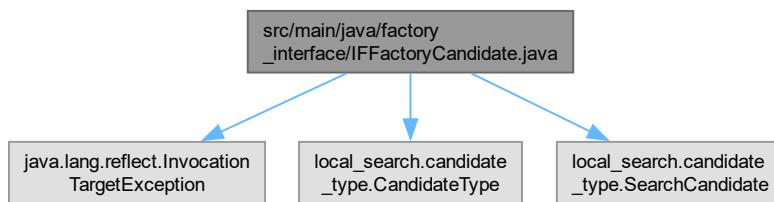
- interface [factory_interface.IFFactoryAcceptCandidate](#)

Paquetes

- package [factory_interface](#)
@(#)[IFFactoryAcceptCandidate.java](#)

7.28. Referencia del archivo**src/main/java/factory_interface/IFFactoryCandidate.java**

```
import java.lang.reflect.InvocationTargetException;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.SearchCandidate;
Gráfico de dependencias incluidas en IFFactoryCandidate.java:
```

**Clases**

- interface [factory_interface.IFFactoryCandidate](#)

Paquetes

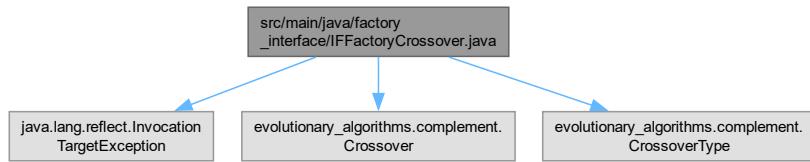
- package [factory_interface](#)
@(#)[IFFactoryAcceptCandidate.java](#)

7.29. Referencia del archivo**src/main/java/factory_interface/IFFactoryCrossover.java**

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Crossover;
```

```
import evolutionary_algorithms.complement.CrossoverType;
```

Gráfico de dependencias incluidas en IFFactoryCrossover.java:



Clases

- interface [factory_interface.IFFactoryCrossover](#)

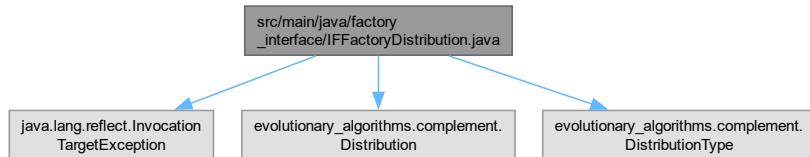
Paquetes

- package [factory_interface](#)
 `@(#)` [IFFactoryAcceptCandidate.java](#)

7.30. Referencia del archivo

src/main/java/factory_interface/IFFactoryDistribution.java

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Distribution;
import evolutionary_algorithms.complement.DistributionType;
Gráfico de dependencias incluidas en IFFactoryDistribution.java:
```



Clases

- interface [factory_interface.IFFactoryDistribution](#)

Paquetes

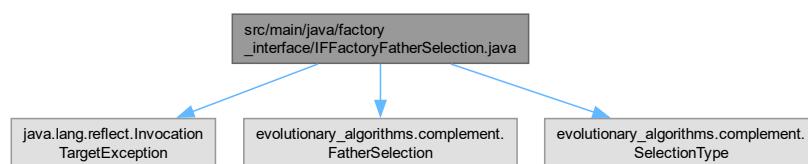
- package [factory_interface](#)
 `@(#)` [IFFactoryAcceptCandidate.java](#)

7.31. Referencia del archivo

src/main/java/factory_interface/IFFactoryFatherSelection.java

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.FatherSelection;
import evolutionary_algorithms.complement.SelectionType;
```

Gráfico de dependencias incluidas en IFFactoryFatherSelection.java:



Clases

- interface [factory_interface.IFFactoryFatherSelection](#)

Paquetes

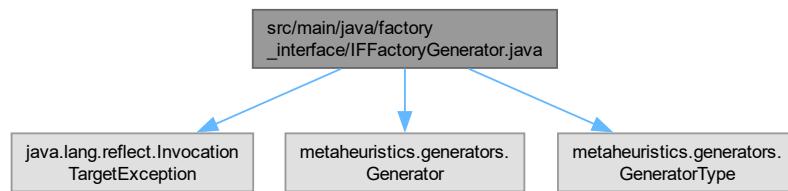
- package [factory_interface](#)
`@(#)` [IFFactoryAcceptCandidate.java](#)

7.32. Referencia del archivo

src/main/java/factory_interface/IFFactoryGenerator.java

```
import java.lang.reflect.InvocationTargetException;
import metaheuristics.generators.Generator;
import metaheuristics.generators.GeneratorType;
```

Gráfico de dependencias incluidas en IFFactoryGenerator.java:



Clases

- interface [factory_interface.IFFactoryGenerator](#)

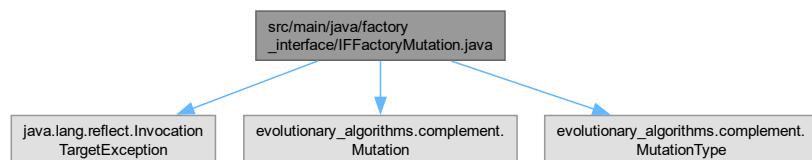
Paquetes

- package `factory_interface`
 @(#)`IFFactoryAcceptCandidate.java`

7.33. Referencia del archivo `src/main/java/factory_interface/IFFactoryMutation.java`

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Mutation;
import evolutionary_algorithms.complement.MutationType;
```

Gráfico de dependencias incluidas en `IFFactoryMutation.java`:



Clases

- interface `factory_interface.IFFactoryMutation`

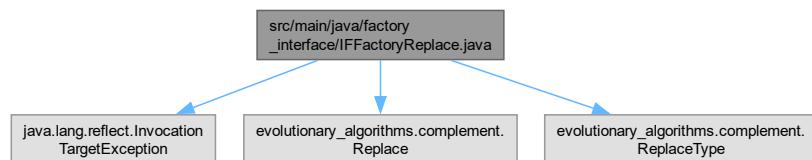
Paquetes

- package `factory_interface`
 @(#)`IFFactoryAcceptCandidate.java`

7.34. Referencia del archivo `src/main/java/factory_interface/IFFactoryReplace.java`

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Replace;
import evolutionary_algorithms.complement.ReplaceType;
```

Gráfico de dependencias incluidas en `IFFactoryReplace.java`:



Clases

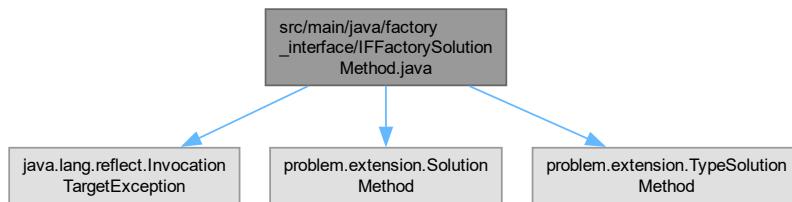
- interface [factory_interface.IFFactoryReplace](#)

Paquetes

- package [factory_interface](#)
 @(#)[IFFactoryAcceptCandidate.java](#)

7.35. Referencia del archivo**src/main/java/factory_interface/IFFactorySolutionMethod.java**

```
import java.lang.reflect.InvocationTargetException;
import problem.extension.SolutionMethod;
import problem.extension.TypeSolutionMethod;
Gráfico de dependencias incluidas en IFFactorySolutionMethod.java:
```

**Clases**

- interface [factory_interface.IFFactorySolutionMethod](#)

Paquetes

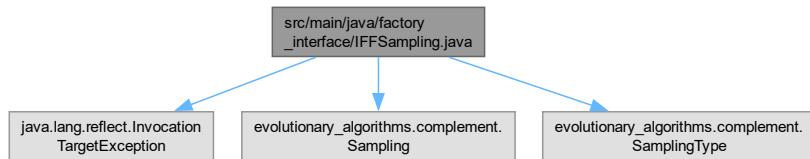
- package [factory_interface](#)
 @(#)[IFFactoryAcceptCandidate.java](#)

7.36. Referencia del archivo**src/main/java/factory_interface/IFFSampling.java**

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Sampling;
```

```
import evolutionary_algorithms.complement.SamplingType;
```

Gráfico de dependencias incluidas en IFFSampling.java:



Clases

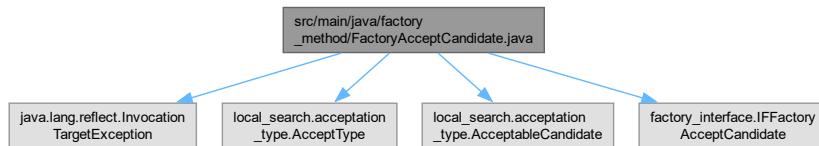
- interface [factory_interface.IFFSampling](#)

Paquetes

- package [factory_interface](#)
`@(#)` [IFFFactoryAcceptCandidate.java](#)

7.37. Referencia del archivo [src/main/java/factory_method/FactoryAcceptCandidate.java](#)

```
import java.lang.reflect.InvocationTargetException;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import factory_interface.IFFFactoryAcceptCandidate;
Gráfico de dependencias incluidas en FactoryAcceptCandidate.java:
```



Clases

- class [factory_method.FactoryAcceptCandidate](#)

Paquetes

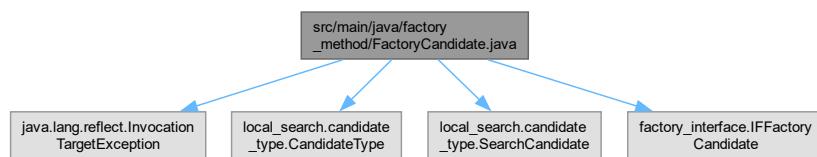
- package [factory_method](#)
`@(#)` [FactoryAcceptCandidate.java](#)

7.38. Referencia del archivo

src/main/java/factory_method/FactoryCandidate.java

```
import java.lang.reflect.InvocationTargetException;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.SearchCandidate;
import factory_interface.IFFactoryCandidate;
```

Gráfico de dependencias incluidas en FactoryCandidate.java:



Clases

- class [factory_method.FactoryCandidate](#)

Paquetes

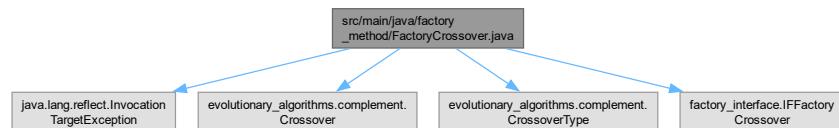
- package [factory_method](#)
 - [FactoryAcceptCandidate.java](#)

7.39. Referencia del archivo

src/main/java/factory_method/FactoryCrossover.java

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Crossover;
import evolutionary_algorithms.complement.CrossoverType;
import factory_interface.IFFactoryCrossover;
```

Gráfico de dependencias incluidas en FactoryCrossover.java:



Clases

- class [factory_method.FactoryCrossover](#)

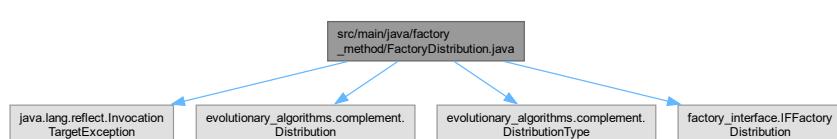
Paquetes

- package `factory_method`
`@(#)` `FactoryAcceptCandidate.java`

7.40. Referencia del archivo `src/main/java/factory_method/FactoryDistribution.java`

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Distribution;
import evolutionary_algorithms.complement.DistributionType;
import factory_interface.IFFactoryDistribution;
```

Gráfico de dependencias incluidas en `FactoryDistribution.java`:



Clases

- class `factory_method.FactoryDistribution`

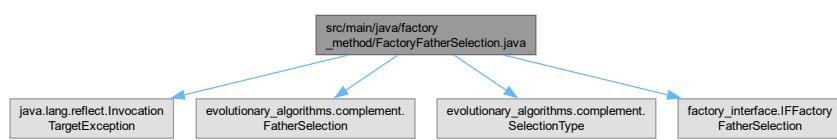
Paquetes

- package `factory_method`
`@(#)` `FactoryAcceptCandidate.java`

7.41. Referencia del archivo `src/main/java/factory_method/FactoryFatherSelection.java`

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.FatherSelection;
import evolutionary_algorithms.complement.SelectionType;
import factory_interface.IFFactoryFatherSelection;
```

Gráfico de dependencias incluidas en `FactoryFatherSelection.java`:



Clases

- class [factory_method.FactoryFatherSelection](#)

Paquetes

- package [factory_method](#)
 @(#) [FactoryAcceptCandidate.java](#)

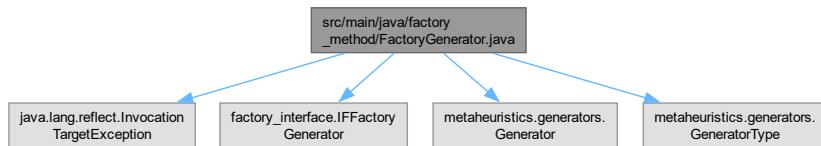
7.42. Referencia del archivo**src/main/java/factory_method/FactoryGenerator.java**

```
import java.lang.reflect.InvocationTargetException;
import factory_interface.IFFactoryGenerator;
```

```
import metaheuristics.generators.Generator;
```

```
import metaheuristics.generators.GeneratorType;
```

Gráfico de dependencias incluidas en FactoryGenerator.java:

**Clases**

- class [factory_method.FactoryGenerator](#)

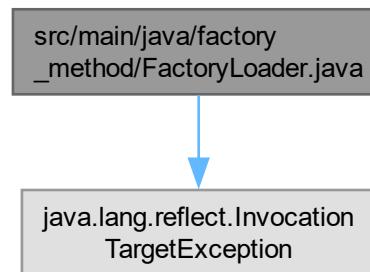
Paquetes

- package [factory_method](#)
 @(#) [FactoryAcceptCandidate.java](#)

7.43. Referencia del archivo src/main/java/factory_method/FactoryLoader.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactoryLoader.java:



Clases

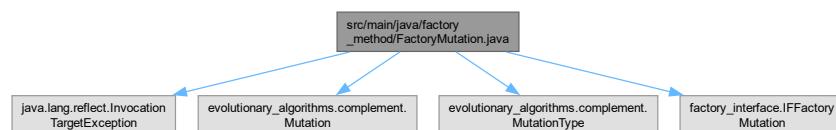
- class [factory_method.FactoryLoader](#)

Paquetes

- package [factory_method](#)
`@(#)` [FactoryAcceptCandidate.java](#)

7.44. Referencia del archivo src/main/java/factory_method/FactoryMutation.java

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Mutation;
import evolutionary_algorithms.complement.MutationType;
import factory_interface.IFFactoryMutation;
Gráfico de dependencias incluidas en FactoryMutation.java:
```



Clases

- class [factory_method.FactoryMutation](#)

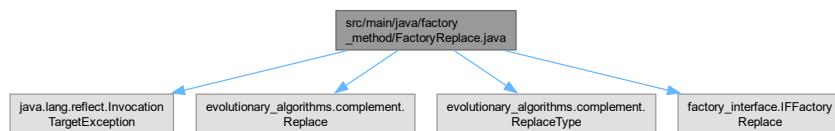
Paquetes

- package [factory_method](#)
`@(#)` [FactoryAcceptCandidate.java](#)

7.45. Referencia del archivo**src/main/java/factory_method/FactoryReplace.java**

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Replace;
import evolutionary_algorithms.complement.ReplaceType;
import factory_interface.IFFactoryReplace;
```

Gráfico de dependencias incluidas en FactoryReplace.java:

**Clases**

- class [factory_method.FactoryReplace](#)

Paquetes

- package [factory_method](#)
`@(#)` [FactoryAcceptCandidate.java](#)

7.46. Referencia del archivo**src/main/java/factory_method/FactorySampling.java**

```
import java.lang.reflect.InvocationTargetException;
import evolutionary_algorithms.complement.Sampling;
import evolutionary_algorithms.complement.SamplingType;
import factory_interface.IFFSampling;
```

Gráfico de dependencias incluidas en FactorySampling.java:



Clases

- class [factory_method.FactorySampling](#)

Paquetes

- package [factory_method](#)
`@(#)` [FactoryAcceptCandidate.java](#)

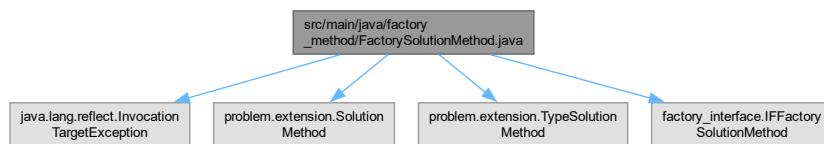
7.47. Referencia del archivo src/main/java/factory_method/FactorySolutionMethod.java

```
import java.lang.reflect.InvocationTargetException;
import problem.extension.SolutionMethod;
```

```
import problem.extension.TypeSolutionMethod;
```

```
import factory_interface.IFFactorySolutionMethod;
```

Gráfico de dependencias incluidas en FactorySolutionMethod.java:

**Clases**

- class [factory_method.FactorySolutionMethod](#)

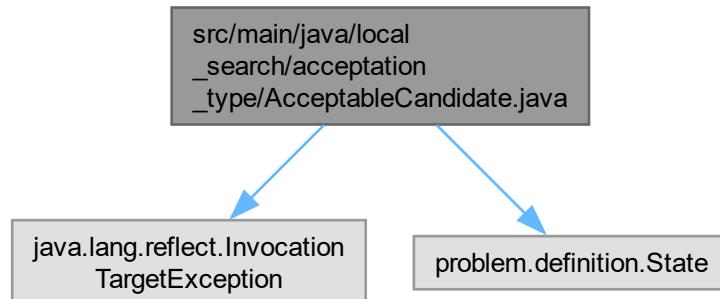
Paquetes

- package [factory_method](#)
`@(#)` [FactoryAcceptCandidate.java](#)

7.48. Referencia del archivo src/main/java/local_search/acceptation_type/AcceptableCandidate.java

```
import java.lang.reflect.InvocationTargetException;
import problem.definition.State;
```

Gráfico de dependencias incluidas en AcceptableCandidate.java:



Clases

- class [local_search.acceptation_type.AcceptableCandidate](#)

Paquetes

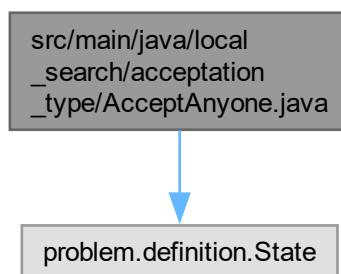
- package [local_search.acceptation_type](#)
 `@(#)` [AcceptAnyone.java](#)

7.49. Referencia del archivo

src/main/java/local_search/acceptation_type/AcceptAnyone.java

`import problem.definition.State;`

Gráfico de dependencias incluidas en AcceptAnyone.java:



Clases

- class [local_search.acceptation_type.AcceptAnyone](#)

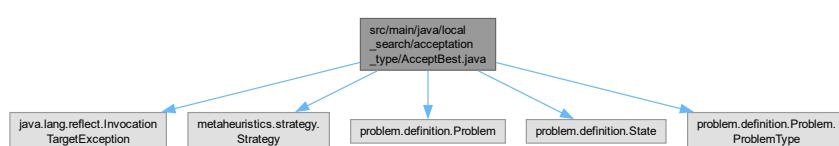
Paquetes

- package [local_search.acceptation_type](#)
@(#)[AcceptAnyone.java](#)

7.50. Referencia del archivo**src/main/java/local_search/acceptation_type/AcceptBest.java**

```
import java.lang.reflect.InvocationTargetException;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
```

Gráfico de dependencias incluidas en AcceptBest.java:

**Clases**

- class [local_search.acceptation_type.AcceptBest](#)

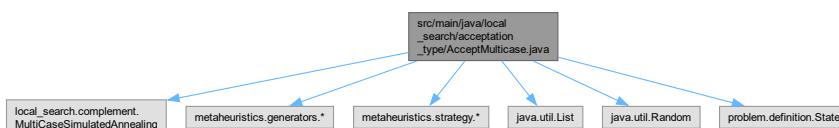
Paquetes

- package [local_search.acceptation_type](#)
@(#)[AcceptAnyone.java](#)

7.51. Referencia del archivo**src/main/java/local_search/acceptation_type/AcceptMulticase.java**

```
import local_search.complement.MultiCaseSimulatedAnnealing;
import metaheuristics.generators.*;
import metaheuristics.strategy.*;
import java.util.List;
import java.util.Random;
import problem.definition.State;
```

Gráfico de dependencias incluidas en AcceptMulticase.java:



Clases

- class [local_search.acceptation_type.AcceptMulticase](#)

Implementa la lógica de aceptación de candidatos para el algoritmo de Recocido Simulado Multicaso.

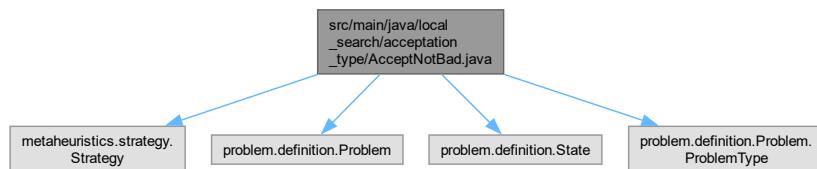
Paquetes

- package [local_search.acceptation_type](#)
`@(#)` [AcceptAnyone.java](#)

7.52. Referencia del archivo

src/main/java/local_search/acceptation_type/AcceptNotBad.java

```
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
Gráfico de dependencias incluidas en AcceptNotBad.java:
```



Clases

- class [local_search.acceptation_type.AcceptNotBad](#)

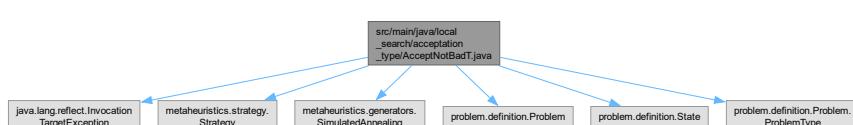
Paquetes

- package [local_search.acceptation_type](#)
`@(#)` [AcceptAnyone.java](#)

7.53. Referencia del archivo

src/main/java/local_search/acceptation_type/AcceptNotBadT.java

```
import java.lang.reflect.InvocationTargetException;
import metaheuristics.strategy.Strategy;
import metaheuristics.generators.SimulatedAnnealing;
import problem.definition.Problem;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
Gráfico de dependencias incluidas en AcceptNotBadT.java:
```



Clases

- class [local_search.acceptation_type.AcceptNotBadT](#)

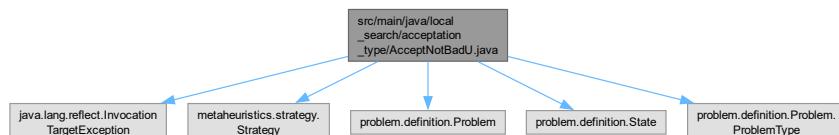
Paquetes

- package [local_search.acceptation_type](#)
@(#) [AcceptAnyone.java](#)

7.54. Referencia del archivo

src/main/java/local_search/acceptation_type/AcceptNotBadU.java

```
import java.lang.reflect.InvocationTargetException;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
Gráfico de dependencias incluidas en AcceptNotBadU.java:
```



Clases

- class [local_search.acceptation_type.AcceptNotBadU](#)

Paquetes

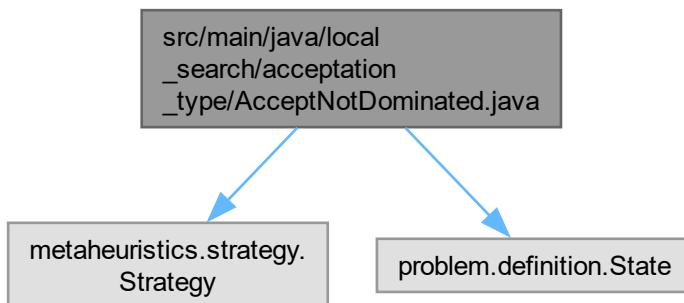
- package [local_search.acceptation_type](#)
 `@(#)` [AcceptAnyone.java](#)

7.55. Referencia del archivo [src/main/java/local_search/acceptation_type/AcceptNotDominated.java](#)

```
import metaheuristics.strategy.Strategy;
```

```
import problem.definition.State;
```

Gráfico de dependencias incluidas en AcceptNotDominated.java:



Clases

- class [local_search.acceptation_type.AcceptNotDominated](#)

Paquetes

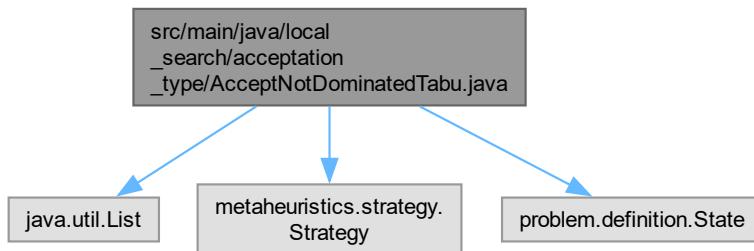
- package [local_search.acceptation_type](#)
 `@(#)` [AcceptAnyone.java](#)

7.56. Referencia del archivo [src/main/java/local_search/acceptation_type/AcceptNotDominatedTabu.java](#)

```
import java.util.List;  
import metaheuristics.strategy.Strategy;
```

```
import problem.definition.State;
```

Gráfico de dependencias incluidas en AcceptNotDominatedTabu.java:



Clases

- class [local_search.acceptation_type.AcceptNotDominatedTabu](#)

Paquetes

- package [local_search.acceptation_type](#)
 @(#) [AcceptAnyone.java](#)

7.57. Referencia del archivo

src/main/java/local_search/acceptation_type/AcceptType.java

Clases

- enum [local_search.acceptation_type.AcceptType](#)

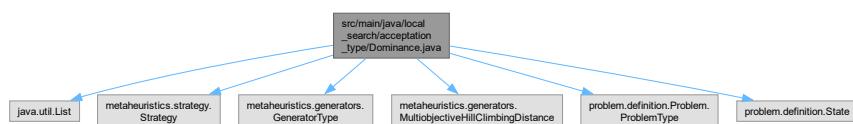
Paquetes

- package [local_search.acceptation_type](#)
 @(#) [AcceptAnyone.java](#)

7.58. Referencia del archivo

src/main/java/local_search/acceptation_type/Dominance.java

```
import java.util.List;
import metaheuristics.strategy.Strategy;
import metaheuristics.generators.GeneratorType;
import metaheuristics.generators.MutliobjectiveHillClimbingDistance;
import problem.definition.Problem.ProblemType;
import problem.definition.State;
Gráfico de dependencias incluidas en Dominance.java:
```



Clases

- class [local_search.acceptation_type.Dominance](#)

Paquetes

- package [local_search.acceptation_type](#)
 @(#) [AcceptAnyone.java](#)

7.59. Referencia del archivo

src/main/java/local_search/candidate_type/CandidateType.java

Clases

- enum [local_search.candidate_type.CandidateType](#)

Paquetes

- package [local_search.candidate_type](#)
 @(#) [TypeCandidate.java](#)

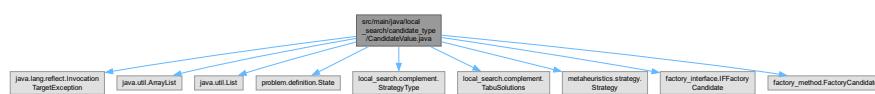
7.60. Referencia del archivo

src/main/java/local_search/candidate_type/CandidateValue.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import problem.definition.State;
import local_search.complement.StrategyType;
import local_search.complement.TabuSolutions;
import metaheuristics.strategy.Strategy;
import factory_interface.IFFactoryCandidate;
import factory_method.FactoryCandidate;
Gráfico de dependencias incluidas en CandidateValue.java:

```



Clases

- class [local_search.candidate_type.CandidateValue](#)

Paquetes

- package [local_search.candidate_type](#)
@(#)*TypeCandidate.java*

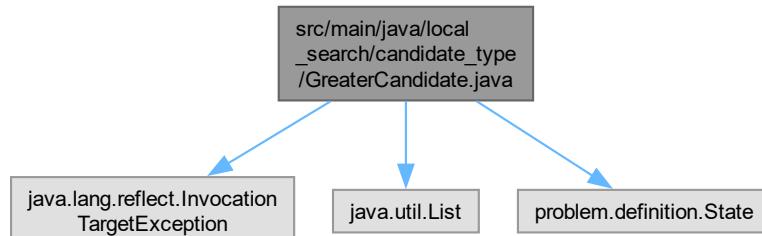
7.61. Referencia del archivo

src/main/java/local_search/candidate_type/GreaterCandidate.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.List;
import problem.definition.State;
Gráfico de dependencias incluidas en GreaterCandidate.java:

```



Clases

- class [local_search.candidate_type.GreaterCandidate](#)

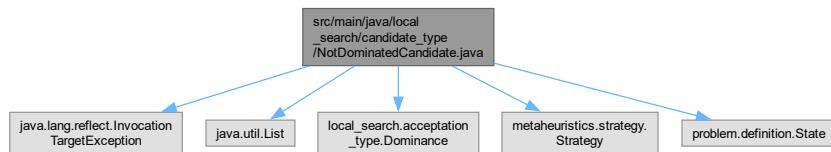
Paquetes

- package [local_search.candidate_type](#)
@(#) [TypeCandidate.java](#)

7.62. Referencia del archivo [src/main/java/local_search/candidate_type/NotDominatedCandidate.java](#)

```
import java.lang.reflect.InvocationTargetException;
import java.util.List;
import local_search.acceptation_type.Dominance;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
```

Gráfico de dependencias incluidas en NotDominatedCandidate.java:

**Clases**

- class [local_search.candidate_type.NotDominatedCandidate](#)

Paquetes

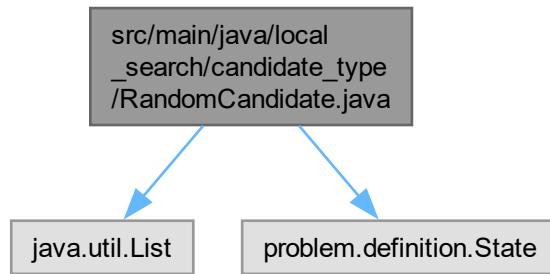
- package [local_search.candidate_type](#)
@(#) [TypeCandidate.java](#)

7.63. Referencia del archivo

src/main/java/local_search/candidate_type/RandomCandidate.java

```
import java.util.List;
import problem.definition.State;
```

Gráfico de dependencias incluidas en RandomCandidate.java:



Clases

- class [local_search.candidate_type.RandomCandidate](#)

Paquetes

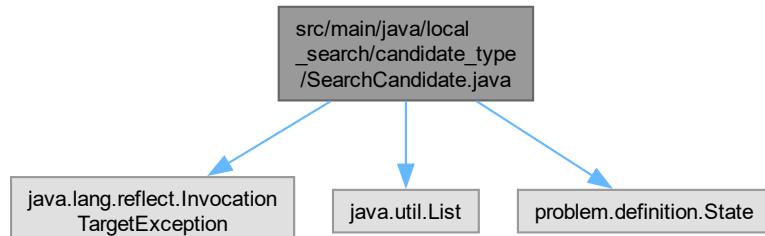
- package [local_search.candidate_type](#)
 @(#) TypeCandidate.java

7.64. Referencia del archivo

src/main/java/local_search/candidate_type/SearchCandidate.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.List;
import problem.definition.State;
```

Gráfico de dependencias incluidas en SearchCandidate.java:



Clases

- class [local_search.candidate_type.SearchCandidate](#)

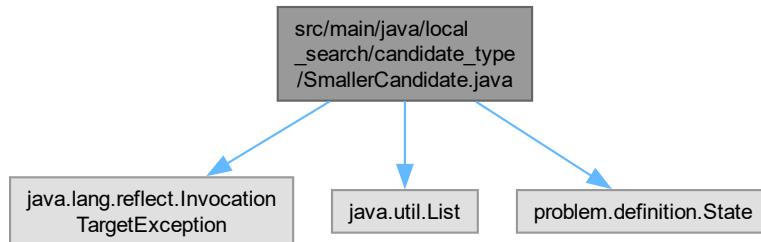
Paquetes

- package [local_search.candidate_type](#)
 @(#) *TypeCandidate.java*

7.65. Referencia del archivo

[src/main/java/local_search/candidate_type/SmallerCandidate.java](#)

```
import java.lang.reflect.InvocationTargetException;
import java.util.List;
import problem.definition.State;
Gráfico de dependencias incluidas en SmallerCandidate.java:
```



Clases

- class [local_search.candidate_type.SmallerCandidate](#)

Paquetes

- package [local_search.candidate_type](#)
 @(#) *TypeCandidate.java*

7.66. Referencia del archivo src/main/java/local_search/complement/ MultiCaseSimulatedAnnealing.java

```

import metaheuristics.generators.Generator;
import metaheuristics.generators.GeneratorType;
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;

```

Gráfico de dependencias incluidas en MultiCaseSimulatedAnnealing.java:



Clases

- class `local_search.complement.MultiCaseSimulatedAnnealing`

Paquetes

- package `local_search.complement`
 - `@(#) Strategy.java`

7.67. Referencia del archivo src/main/java/local_search/MultiCaseSimulatedAnnealing.java

```

import metaheuristics.generators.Generator;
import metaheuristics.generators.GeneratorType;
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;

```

```
import problem.definition.State;
```

Gráfico de dependencias incluidas en MultiCaseSimulatedAnnealing.java:



Clases

- class [local_search.MultiCaseSimulatedAnnealing](#)

Paquetes

- package [local_search](#)

7.68. Referencia del archivo src/main/java/metaheuristics/generators/← MultiCaseSimulatedAnnealing.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;
```

Gráfico de dependencias incluidas en MultiCaseSimulatedAnnealing.java:



Clases

- class [metaheuristics.generators.MultiCaseSimulatedAnnealing](#)

Paquetes

- package [metaheuristics.generators](#)

7.69. Referencia del archivo `src/main/java/local_search/complement/StopExecute.java`

Clases

- class `local_search.complement.StopExecute`

Paquetes

- package `local_search.complement`
 `@(#)` *Strategy.java*

7.70. Referencia del archivo `src/main/java/local_search/complement/StrategyType.java`

Clases

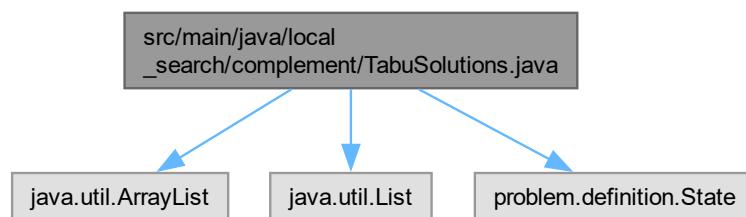
- enum `local_search.complement.StrategyType`

Paquetes

- package `local_search.complement`
 `@(#)` *Strategy.java*

7.71. Referencia del archivo `src/main/java/local_search/complement/TabuSolutions.java`

```
import java.util.ArrayList;
import java.util.List;
import problem.definition.State;
Gráfico de dependencias incluidas en TabuSolutions.java:
```



Clases

- class [local_search.complement.TabuSolutions](#)

Paquetes

- package [local_search.complement](#)
 @(#)[Strategy.java](#)

7.72. Referencia del archivo

src/main/java/local_search/complement/UpdateParameter.java

```
import factory_interface.IFFactoryGenerator;
import factory_method.FactoryGenerator;
import java.lang.reflect.InvocationTargetException;
import metaheuristics.strategy.Strategy;
import metaheuristics.generators.DistributionEstimationAlgorithm;
import metaheuristics.generators.EvolutionStrategies;
import metaheuristics.generators.GeneratorType;
import metaheuristics.generators.GeneticAlgorithm;
import metaheuristics.generators.ParticleSwarmOptimization;
```

Gráfico de dependencias incluidas en UpdateParameter.java:



Clases

- class [local_search.complement.UpdateParameter](#)

Paquetes

- package [local_search.complement](#)
 @(#)[Strategy.java](#)

7.73. Referencia del archivo **src/main/java/local_search/TestDummy.java**

Clases

- class [local_search.TestDummy](#)

Paquetes

- package [local_search](#)

7.74. Referencia del archivo src/main/java/metaheuristics/generators/[←](#) DistributionEstimationAlgorithm.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import problem.definition.Problem.Type;
import evolutionary_algorithms.complement.DistributionType;
import evolutionary_algorithms.complement.FatherSelection;
import evolutionary_algorithms.complement.Replace;
import evolutionary_algorithms.complement.ReplaceType;
import evolutionary_algorithms.complement.Sampling;
import evolutionary_algorithms.complement.SamplingType;
import evolutionary_algorithms.complement.SelectionType;
import factory_interface.IFFSampling;
import factory_interface.IFFFactoryFatherSelection;
import factory_interface.IFFFactoryReplace;
import factory_method.FactoryFatherSelection;
import factory_method.FactoryReplace;
import factory_method.FactorySampling;
Gráfico de dependencias incluidas en DistributionEstimationAlgorithm.java:
```



Clases

- class [metaheuristics.generators.DistributionEstimationAlgorithm](#)

Paquetes

- package [metaheuristics.generators](#)

7.75. Referencia del archivo src/main/java/metaheuristics/generators/EvolutionStrategies.java

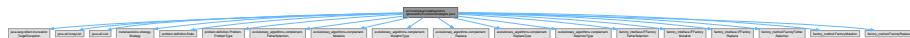
```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import problem.definition.Problem.Type;
import evolutionary_algorithms.complement.FatherSelection;
import evolutionary_algorithms.complement.Mutation;
import evolutionary_algorithms.complement.MutationType;
import evolutionary_algorithms.complement.Replace;
import evolutionary_algorithms.complement.ReplaceType;
import evolutionary_algorithms.complement.SelectionType;
import factory_interface.IFFFactoryFatherSelection;
```

```

import factory_interface.IFFactoryMutation;
import factory_interface.IFFactoryReplace;
import factory_method.FactoryFatherSelection;
import factory_method.FactoryMutation;
import factory_method.FactoryReplace;

```

Gráfico de dependencias incluidas en EvolutionStrategies.java:



Clases

- class [metaheuristics.generators.EvolutionStrategies](#)

Paquetes

- package [metaheuristics.generators](#)

7.76. Referencia del archivo

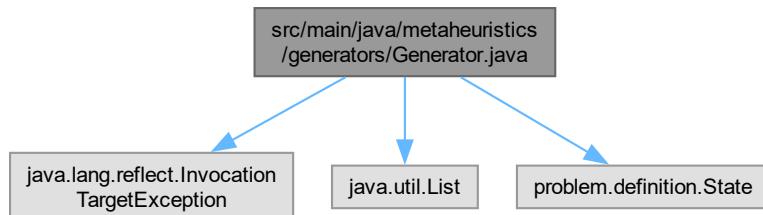
src/main/java/metaheuristics/generators/Generator.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.List;
import problem.definition.State;

```

Gráfico de dependencias incluidas en Generator.java:



Clases

- class [metaheuristics.generators.Generator](#)

Paquetes

- package [metaheuristics.generators](#)

7.77. Referencia del archivo

[src/main/java/metaheuristics/generators/GeneratorType.java](#)

Clases

- enum [metaheuristics.generators.GeneratorType](#)

Paquetes

- package [metaheuristics.generators](#)

7.78. Referencia del archivo

[src/main/java/metaheuristics/generators/GeneticAlgorithm.java](#)

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import problem.definition.ProblemType;
import evolutionary_algorithms.complement.Crossover;
import evolutionary_algorithms.complement.CrossoverType;
import evolutionary_algorithms.complement.FatherSelection;
import evolutionary_algorithms.complement.Mutation;
import evolutionary_algorithms.complement.MutationType;
import evolutionary_algorithms.complement.Replace;
import evolutionary_algorithms.complement.ReplaceType;
import evolutionary_algorithms.complement.SelectionType;
import factory_interface.IFFactoryCrossover;
import factory_interface.IFFactoryFatherSelection;
import factory_interface.IFFactoryMutation;
import factory_interface.IFFactoryReplace;
import factory_method.FactoryCrossover;
import factory_method.FactoryFatherSelection;
import factory_method.FactoryMutation;
import factory_method.FactoryReplace;
```

Gráfico de dependencias incluidas en GeneticAlgorithm.java:



Clases

- class [metaheuristics.generators.GeneticAlgorithm](#)

Paquetes

- package [metaheuristics.generators](#)

7.79. Referencia del archivo

src/main/java/metaheuristics/generators/HillClimbing.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;

```

Gráfico de dependencias incluidas en HillClimbing.java:



Clases

- class [metaheuristics.generators.HillClimbing](#)

Paquetes

- package [metaheuristics.generators](#)

7.80. Referencia del archivo

src/main/java/metaheuristics/generators/HillClimbingRestart.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import problem.definition.Problem.ProblemType;

```

Gráfico de dependencias incluidas en HillClimbingRestart.java:



Clases

- class [metaheuristics.generators.HillClimbingRestart](#)

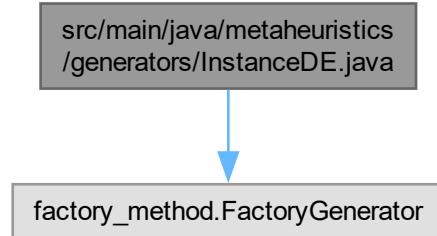
Paquetes

- package [metaheuristics.generators](#)

7.81. Referencia del archivo**src/main/java/metaheuristics/generators/InstanceDE.java**

```
import factory_method.FactoryGenerator;
```

Gráfico de dependencias incluidas en InstanceDE.java:

**Clases**

- class [metaheuristics.generators.InstanceDE](#)

Paquetes

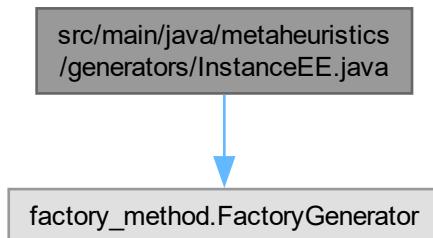
- package [metaheuristics.generators](#)

7.82. Referencia del archivo

src/main/java/metaheuristics/generators/InstanceEE.java

```
import factory_method.FactoryGenerator;
```

Gráfico de dependencias incluidas en InstanceEE.java:



Clases

- class [metaheuristics.generators.InstanceEE](#)

Paquetes

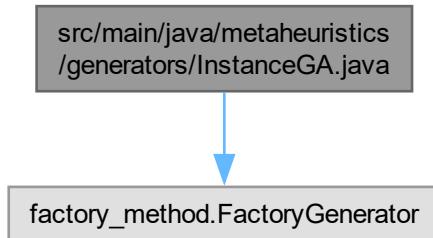
- package [metaheuristics.generators](#)

7.83. Referencia del archivo

src/main/java/metaheuristics/generators/InstanceGA.java

```
import factory_method.FactoryGenerator;
```

Gráfico de dependencias incluidas en InstanceGA.java:



Clases

- class [metaheuristics.generators.InstanceGA](#)

Paquetes

- package [metaheuristics.generators](#)

7.84. Referencia del archivo**src/main/java/metaheuristics/generators/LimitRoulette.java****Clases**

- class [metaheuristics.generators.LimitRoulette](#)

Paquetes

- package [metaheuristics.generators](#)

7.85. Referencia del archivo**src/main/java/metaheuristics/generators/LimitThreshold.java**

```

import java.lang.reflect.InvocationTargetException;
import java.util.List;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
Gráfico de dependencias incluidas en LimitThreshold.java:
```

**Clases**

- class [metaheuristics.generators.LimitThreshold](#)

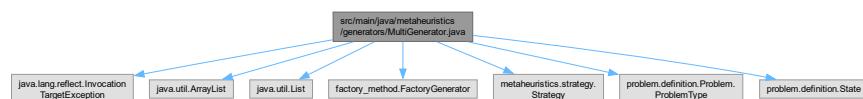
Paquetes

- package [metaheuristics.generators](#)

7.86. Referencia del archivo

[src/main/java/metaheuristics/generators/MultiGenerator.java](#)

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import factory_method.FactoryGenerator;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem.ProblemType;
import problem.definition.State;
Gráfico de dependencias incluidas en MultiGenerator.java:
```



Clases

- class [metaheuristics.generators.MultiGenerator](#)

Paquetes

- package [metaheuristics.generators](#)

7.87. Referencia del archivo [src/main/java/metaheuristics/generators/](#) ↵

[MultiobjectiveHillClimbingDistance.java](#)

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import problem.definition.State;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
Gráfico de dependencias incluidas en MultiobjectiveHillClimbingDistance.java:
```



7.88 Referencia del archivo src/main/java/metaheuristics/generators/MultiobjectiveHillClimbingRestart.java

Clases

- class [metaheuristics.generators.MultiobjectiveHillClimbingDistance](#)

Paquetes

- package [metaheuristics.generators](#)

7.88. Referencia del archivo src/main/java/metaheuristics/generators/ MultiobjectiveHillClimbingRestart.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import problem.definition.State;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en MultiobjectiveHillClimbingRestart.java:



Clases

- class [metaheuristics.generators.MultiobjectiveHillClimbingRestart](#)

Paquetes

- package [metaheuristics.generators](#)

7.89. Referencia del archivo src/main/java/metaheuristics/generators/ MultiobjectiveStochasticHillClimbing.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import local_search.acceptation_type.AcceptType;
```

```

import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.State;

```

Gráfico de dependencias incluidas en MultiobjectiveStochasticHillClimbing.java:



Clases

- class `metaheuristics.generators.MultiobjectiveStochasticHillClimbing`

Paquetes

- package `metaheuristics.generators`

7.90. Referencia del archivo src/main/java/metaheuristics/generators/← MultiobjectiveTabuSearch.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import local_search.complement.TabuSolutions;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;

```

Gráfico de dependencias incluidas en MultiobjectiveTabuSearch.java:



Clases

- class `metaheuristics.generators.MultiobjectiveTabuSearch`

Paquetes

- package `metaheuristics.generators`

7.91. Referencia del archivo

`src/main/java/metaheuristics/generators/MyTestClass.java`

Clases

- class `metaheuristics.generators.MyTestClass`

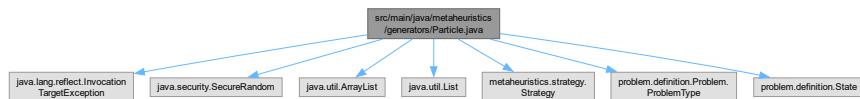
Paquetes

- package `metaheuristics.generators`

7.92. Referencia del archivo

`src/main/java/metaheuristics/generators/Particle.java`

```
import java.lang.reflect.InvocationTargetException;
import java.security.SecureRandom;
import java.util.ArrayList;
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem.ProblemType;
import problem.definition.State;
Gráfico de dependencias incluidas en Particle.java:
```



Clases

- class `metaheuristics.generators.Particle`

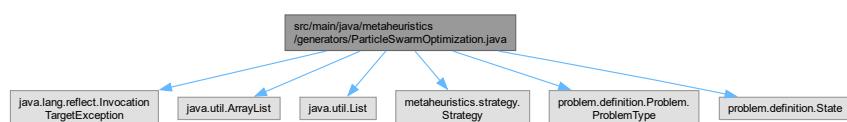
Representa una partícula en el algoritmo de Optimización por Enjambre de Partículas (PSO).

Paquetes

- package `metaheuristics.generators`

7.93. Referencia del archivo src/main/java/metaheuristics/generators/← ParticleSwarmOptimization.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem.ProblemType;
import problem.definition.State;
Gráfico de dependencias incluidas en ParticleSwarmOptimization.java:
```



Clases

- class [metaheuristics.generators.ParticleSwarmOptimization](#)
Implementación del algoritmo de Optimización por Enjambre de Partículas (PSO).

Paquetes

- package [metaheuristics.generators](#)

7.94. Referencia del archivo src/main/java/metaheuristics/generators/RandomSearch.java

```
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrATEGY;
import metaheuristics.strategy.StrATEGY;
import problem.definition.State;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
Gráfico de dependencias incluidas en RandomSearch.java:
```



Clases

- class [metaheuristics.generators.RandomSearch](#)

Paquetes

- package [metaheuristics.generators](#)

7.95. Referencia del archivo**src/main/java/metaheuristics/generators/SimpleTestClass.java****Clases**

- class [metaheuristics.generators.SimpleTestClass](#)

Paquetes

- package [metaheuristics.generators](#)

7.96. Referencia del archivo**src/main/java/metaheuristics/generators/SimulatedAnnealing.java**

```

import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;

```

Gráfico de dependencias incluidas en SimulatedAnnealing.java:

**Clases**

- class [metaheuristics.generators.SimulatedAnnealing](#)

Paquetes

- package [metaheuristics.generators](#)

7.97. Referencia del archivo

src/main/java/metaheuristics/generators/TabuSearch.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import local_search.acceptation_type.AcceptType;
import local_search.acceptation_type.AcceptableCandidate;
import local_search.candidate_type.CandidateType;
import local_search.candidate_type.CandidateValue;
import local_search.complement.StrategyType;
import local_search.complement.TabuSolutions;
import metaheuristics.strategy.Strategy;
import problem.definition.Problem;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
import factory_interface.IFFactoryAcceptCandidate;
import factory_method.FactoryAcceptCandidate;
Gráfico de dependencias incluidas en TabuSearch.java:

```



Clases

- class [metaheuristics.generators.TabuSearch](#)

Paquetes

- package [metaheuristics.generators](#)

7.98. Referencia del archivo

src/main/java/metaheuristics/strategy/Strategy.java

```

import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import java.util.SortedMap;
import java.util.TreeMap;
import factory_interface.IFFactoryGenerator;
import factory_method.FactoryGenerator;
import problem.definition.Problem;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
import local_search.acceptation_type.Dominance;
import local_search.complement.StopExecute;
import local_search.complement.UpdateParameter;
import metaheuristics.generators.DistributionEstimationAlgorithm;
import metaheuristics.generators.EvolutionStrategies;
import metaheuristics.generators.Generator;

```

```
import metaheuristics.generators.GeneratorType;
import metaheuristics.generators.GeneticAlgorithm;
import metaheuristics.generators.MultiGenerator;
import metaheuristics.generators.ParticleSwarmOptimization;
import metaheuristics.generators.RandomSearch;
```

Gráfico de dependencias incluidas en Strategy.java:



Clases

- class [metaheuristics.strategy.Strategy](#)

Paquetes

- package [metaheuristics.strategy](#)

7.99. Referencia del archivo src/main/java/problem/definition/Codification.java

Clases

- class [problem.definition.Codification](#)

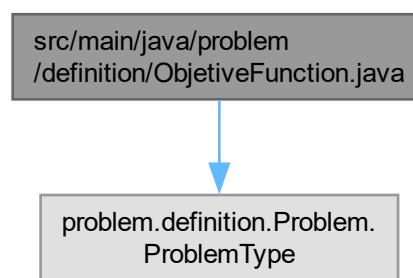
Paquetes

- package [problem.definition](#)

7.100. Referencia del archivo src/main/java/problem/definition/ObjetiveFunction.java

```
import problem.definition.Problem.ProblemType;
```

Gráfico de dependencias incluidas en ObjetiveFunction.java:



Clases

- class [problem.definition.ObjetiveFunction](#)

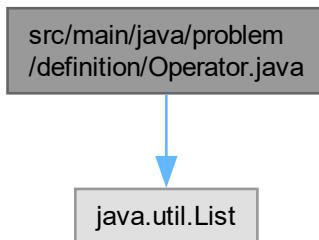
Paquetes

- package [problem.definition](#)

7.101. Referencia del archivo**src/main/java/problem/definition/Operator.java**

```
import java.util.List;
```

Gráfico de dependencias incluidas en Operator.java:

**Clases**

- class [problem.definition.Operator](#)

Paquetes

- package [problem.definition](#)

7.102. Referencia del archivo**src/main/java/problem/definition/Problem.java**

```
import java.lang.reflect.InvocationTargetException;
```

```
import java.util.ArrayList;
```

```
import problem.extension.SolutionMethod;
```

```
import problem.extension.TypeSolutionMethod;
```

```
import factory_interface.IFFactorySolutionMethod;
```

```
import factory_method.FactorySolutionMethod;
```

Gráfico de dependencias incluidas en Problem.java:



Clases

- class [problem.definition.Problem](#)
- enum [problem.definition.Problem.Type](#)

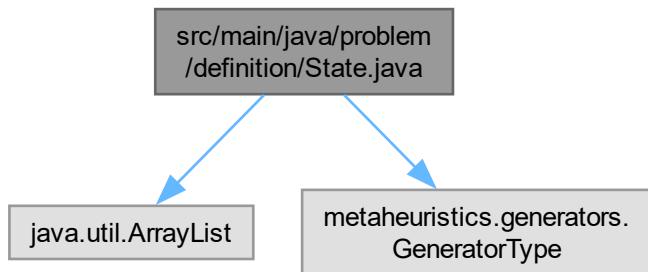
Paquetes

- package [problem.definition](#)

7.103. Referencia del archivo src/main/java/problem/definition/State.java

```
import java.util.ArrayList;
import metaheuristics.generators.GeneratorType;
```

Gráfico de dependencias incluidas en State.java:

**Clases**

- class [problem.definition.State](#)

Paquetes

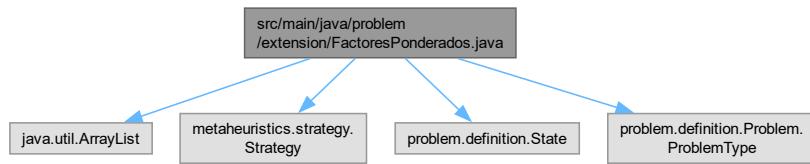
- package [problem.definition](#)

7.104. Referencia del archivo src/main/java/problem/extension/FactoresPonderados.java

```
import java.util.ArrayList;
import metaheuristics.strategy.Strategy;
import problem.definition.State;
```

```
import problem.definition.Problem.ProblemType;
```

Gráfico de dependencias incluidas en FactoresPonderados.java:



Clases

- class [problem.extension.FactoresPonderados](#)

Paquetes

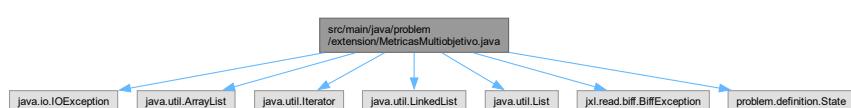
- package [problem.extension](#)

7.105. Referencia del archivo

src/main/java/problem/extension/MetricasMultiobjetivo.java

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import jxl.read.biff.BiffException;
import problem.definition.State;
```

Gráfico de dependencias incluidas en MetricasMultiobjetivo.java:



Clases

- class [problem.extension.MetricasMultiobjetivo](#)

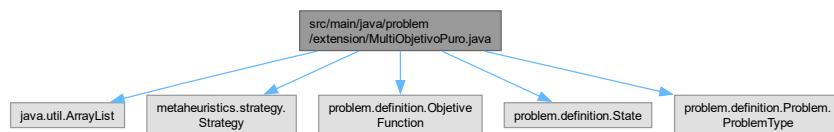
Paquetes

- package [problem.extension](#)

7.106. Referencia del archivo `src/main/java/problem/extension/MultiObjetivoPuro.java`

```
import java.util.ArrayList;
import metaheuristics.strategy.Strategy;
import problem.definition.ObjetiveFunction;
import problem.definition.State;
import problem.definition.Problem.ProblemType;
```

Gráfico de dependencias incluidas en `MultiObjetivoPuro.java`:



Clases

- class [problem.extension.MultiObjetivoPuro](#)

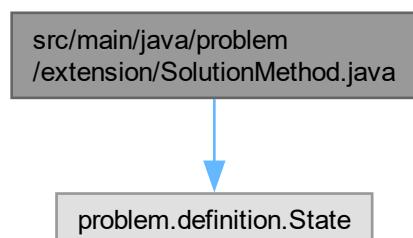
Paquetes

- package [problem.extension](#)

7.107. Referencia del archivo `src/main/java/problem/extension/SolutionMethod.java`

```
import problem.definition.State;
```

Gráfico de dependencias incluidas en `SolutionMethod.java`:



Clases

- class [problem.extension.SolutionMethod](#)

Paquetes

- package [problem.extension](#)

7.108. Referencia del archivo

src/main/java/problem/extension/TypeSolutionMethod.java

Clases

- enum [problem.extension.TypeSolutionMethod](#)

Paquetes

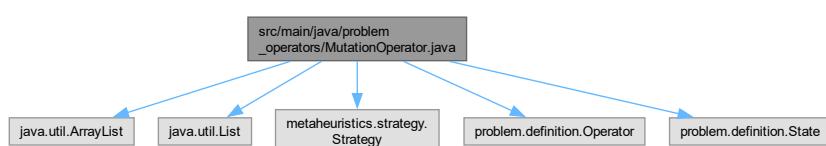
- package [problem.extension](#)

7.109. Referencia del archivo

src/main/java/problem_operators/MutationOperator.java

```
import java.util.ArrayList;
import java.util.List;
import metaheuristics.strategy.Strategy;
import problem.definition.Operator;
import problem.definition.State;
```

Gráfico de dependencias incluidas en MutationOperator.java:



Clases

- class [problem_operators.MutationOperator](#)

Paquetes

- package [problem_operators](#)