

# SCIP

Stefan Vigerske, Humboldt University Berlin, Germany

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>671</b>
<b>2</b>	<b>Model requirements</b>	<b>671</b>
<b>3</b>	<b>Usage</b>	<b>672</b>
3.1	Specification of SCIP Options	672
3.2	Specification of Indicators	672
<b>4</b>	<b>Special Features</b>	<b>673</b>
4.1	SCIP interactive shell	673
4.2	Emphasis Settings	673
4.3	Solution Pool	673
4.4	Solving process tracing	674
4.5	Notes on solving MINLPs with SCIP	674
4.5.1	Special options for convex MINLPs	674
<b>5</b>	<b>Detailed Options Description</b>	<b>674</b>

---

## 1 Introduction

SCIP (Solving Constraint Integer Programs) is developed at the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB). The SCIP main developer had been Tobias Achterberg, current main developers are Timo Berthold, Gerald Gamrath, Ambros Gleixner, Stefan Heinz, Marc Pfetsch, Stefan Vigerske, Michael Winkler, and Kati Wolter. Since SCIP is distributed under the ZIB Academic License, it is only available for users with a **GAMS academic license**.

SCIP is a framework for Constraint Integer Programming oriented towards the needs of Mathematical Programming experts who want to have total control of the solution process and access detailed information down to the guts of the solver. SCIP can also be used as a pure MIP or MINLP solver or as a framework for branch-cut-and-price. Within GAMS, the MIP and MINLP solving facilities of SCIP are available.

For more detailed information, we refer to [1, 2, 3, 4, 5, 6, 8] and the SCIP web site <http://scip.zib.de>, especially the list of papers listed at <http://scip.zib.de/related.shtml>.

GAMS/SCIP uses the linear solver SOPLEX [9] as LP solver, the COIN-OR Interior Point Optimizer IPOPT [7] as nonlinear solver, and CPPAD to compute derivatives of nonlinear functions.

## 2 Model requirements

SCIP supports continuous, binary, integer, semi-continuous, semi-integer, and indicator variables, special ordered sets, and branching priorities for discrete variables.

### 3 Usage

The following statement can be used inside your GAMS program to specify using SCIP

```
Option MIP = SCIP;      { or QCP or NLP or MIQCP or MINLP or ... }
```

The above statement should appear before the Solve statement. If SCIP was specified as the default solver during GAMS installation, the above statement is not necessary.

GAMS/SCIP currently does not support the GAMS Branch-and-Cut-and-Heuristic (BCH) Facility. If you need to use GAMS/SCIP with BCH, please consider to use a GAMS system of version  $\leq 23.3$ , available at [http://www.gams.com/download/download\\_old.htm](http://www.gams.com/download/download_old.htm).

#### 3.1 Specification of SCIP Options

GAMS/SCIP supports the GAMS parameters `reslim`, `iterlim`, `nodlim`, `optcr`, `optca`, and `workspace`. Further, the option `threads` can be used to control the number of threads used in the linear algebra routines of IPOPT (only under Linux and Windows) and for solving LPs (if CPLEX' barrier solver is used).

Options can be specified by a SCIP options file. A SCIP options file consists of one option or comment per line. A pound sign (#) at the beginning of a line causes the entire line to be ignored. Otherwise, the line will be interpreted as an option name and value separated by an equal sign (=) and any amount of white space (blanks or tabs). Further, string values have to be enclosed in quotation marks.

A small example for a `scip.opt` file is:

```
propagating/probing/maxprerounds = 0
separating/maxrounds             = 0
separating/maxroundsroot         = 0
```

It causes GAMS/SCIP to disable probing during presolve and to turn off all cut generators.

#### 3.2 Specification of Indicators

Indicators are a modeling tool to specify that certain equations in a model must only be satisfied if certain binary variables take a specified value. Indicators are not supported by the GAMS language, but can be passed to SCIP via a separate file. The name of that file is specified via the option `gams/indicatorfile` in a SCIP option file.

The indicator specification file declares for some equations, for which value of which binary variables the equation is "switched on". The syntax is

```
indic equation$variable onval
```

where `equation` is the name of the equation, `variable` is the name of the binary variable, and `onval` is either 0 or 1. The line specifies that `equation` has to hold whenever `variable` takes value `onval`.

For example, assume a GAMS model contains a set of equations of the form

```
equ1(i,j,k)$(ord(i)<ord(j)).. lhs =l= rhs;
```

To specify that they only have to be satisfied if a binary variable

```
bin1(i,k)
```

takes the value 1, the indicator specification file should contain the line

```
indic equ1(i,j,k)$bin1(i,k) 1
```

More documentation can be found at <http://www.gams.com/solvers/cpxindic.htm>. In difference to the GAMS/CPLEX interface, the indicator specifications need to be in a separate file for SCIP.

Currently, indicators can only be used for linear equations.

## 4 Special Features

### 4.1 SCIP interactive shell

The interactive shell in SCIP is a powerful tool that allows the user to display various information (e.g., branching statistics, presolved model), load emphasis settings, interrupt a solve to change parameters or trigger a restart, write the model in various file formats, start SCIPs solution counter, and many more things.

When setting the option `gams/interactive` to a nonempty string, the GAMS/SCIP interface opens the interactive shell of SCIP after having load the GAMS problem and parameters and passes the value of the `gams/interactive` parameter to the SCIP interactive shell.

By default, SCIP behaves as if `gams/interactive` has been set to `"optimize write gamssol quit"`, that is, SCIP is requested to solve the problem, then to pass the solution back to GAMS, and to quit.

An example use of the SCIP interactive shell feature via GAMS is to add the following line to your SCIP options file:

```
gams/interactive = "write prob orig.lp presolve write transprob presol.mps opt write gamssol quit"
```

This instructs SCIP to write the original problem to the file `orig.lp` in LP format, to presolve the instance, to write the presolved problem to the file `presolved.mps` in MPS format, to solve the instance, to write the solution out to GAMS, and to finish.

By omitting the `quit` command, SCIPs interactive shell remains open and awaits user input. The command `help` prints a list of available commands. Note, that on Windows, GAMS need to be called with the option `interactivesolver = 1` to allow user input for the solver process.

A tutorial on using the SCIP shell is available at <http://scip.zib.de/doc/html/SHELL.shtml>.

### 4.2 Emphasis Settings

SCIP includes various emphasis settings, which are predefined values for a set of SCIP parameters. Such predefined settings are available for setting the effort that SCIP should spend for, e.g., presolving, separation, or heuristics.

The emphasis settings are not available as single parameters, but can be set via SCIPs interactive shell. E.g., writing `set heuristics emphasis` in the shell displays the available emphasis settings for heuristics (aggressive, fast, off) and expects the user to input which setting to use. Further, general emphasis settings are available in the `set emphasis` menu, some of them giving predefined settings similar to the CPLEX option `mipemphasis`.

Further, option files that specify all available emphasis settings are available at <http://www.gams.com/~svigerske/scip3.0>.

### 4.3 Solution Pool

When SCIP solves a problem, it may find several solutions, whereof only the best one is available to the GAMS user via the variable level values in the GAMS model. If the option `gams/dumpsolutions` is specified, then all alternative solutions found by SCIP are writting into GDX files and an index file with the name given by the `dumpsolutions` option is written.

The GAMS testlib model `dumpsol` shows an example use for this option via GUROBI. It can easily be adapted to be used with SCIP.

## 4.4 Solving process tracing

The option `gams/solvetrace/file` can be used to specify the name of a file where information about the progress of the branch-and-bound tree search in SCIP is stored. The file is created and updated during the solution process, so it may also be used to monitor the progress of SCIP while it still solves the model.

New entries are written periodically, depending on how many nodes have been processed or how much time has been elapsed since the last entry was written. Each entry contains information on the current primal and dual bound.

## 4.5 Notes on solving MINLPs with SCIP

SCIP includes capabilities to handle nonlinear equations specified that are specified via algebraic expressions. Thus, external functions are not supported. Further, not all GAMS operands are supported yet, including trigonometric functions (`sin`, `cos`, ...).

Nonconvex MINLPs are solved by SCIP via a spatial branch-and-bound algorithm using convex relaxations. The tightness of a convex relaxation depends heavily on the variable bounds, thus tight bounds for the nonlinear variables are crucial for SCIP.

### 4.5.1 Special options for convex MINLPs

Convex MINLPs are much easier to solve for SCIP, provided it recognizes the convexity of the model. So far, only a simple convexity check is implemented in SCIP, which may not give a conclusive answer in all cases. However, the option `constraints/nonlinear/assumeconvex = TRUE` can be used to tell SCIP that it should assume all nonlinear constraints to be of convex type. This may help to improve solving times for convex MINLPs considerably.

Another useful feature especially for convex MINLPs is to enable the generation of cuts in the solution of the NLP relaxation in the root node and to consider using these cuts during the whole solution process. This is achieved by the parameters

```
constraints/quadratic/sepanlpmincont = 0
constraints/soc/sepanlpmincont = 0
constraints/nonlinear/sepanlpmincont = 0
constraints/abspower/sepanlpmincont = 0
separating/poolfreq = 1
```

## 5 Detailed Options Description

SCIP supports a large set of options. Sample option files can be obtained from <http://www.gams.com/~svigerske/scip3.0>.

In the following we give a detailed list of most SCIP options.

### GAMS interface specific options

<code>gams/dumpsolutions</code> (string)	
name of solutions index.gdx file for writing all solutions	
<code>gams/indicatorfile</code> (string)	
name of GAMS options file that contains definitions on indicators	
<code>gams/interactive</code> (string)	
command to be issued to the SCIP shell instead of issuing a solve command	
<code>gams/mipstart</code> (boolean)	TRUE
whether to try GAMS variable level values as initial primal solution	
<code>gams/resolvenlp</code> (boolean)	TRUE
whether to resolve MINLP with fixed discrete variables if best solution violates some constraints	

gams/solvetrace/file (string)  
 name of file where to write branch-and-bound trace information too

gams/solvetrace/nodefreq ( $0 \leq \text{integer}$ ) 100  
 frequency in number of nodes when to write branch-and-bound trace information, 0 to disable

gams/solvetrace/timefreq ( $0 \leq \text{real}$ ) 5  
 frequency in seconds when to write branch-and-bound trace information, 0.0 to disable

## Branching

branching/allfullstrong/maxbounddist ( $0 \leq \text{real} \leq 1$ ) 1  
 maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)

branching/allfullstrong/maxdepth ( $-1 \leq \text{integer}$ ) -1  
 maximal depth level, up to which branching rule <allfullstrong> should be used (-1 for no limit)

branching/allfullstrong/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) -1000  
 priority of branching rule <allfullstrong>

branching/clamp ( $0 \leq \text{real} \leq 0.5$ ) 0.2  
 minimal relative distance of branching point to bounds when branching on a continuous variable

branching/delaypscostupdate (boolean) TRUE  
 should updating pseudo costs for continuous variables be delayed to the time after separation?

branching/fullstrong/maxbounddist ( $0 \leq \text{real} \leq 1$ ) 1  
 maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)

branching/fullstrong/maxdepth ( $-1 \leq \text{integer}$ ) -1  
 maximal depth level, up to which branching rule <fullstrong> should be used (-1 for no limit)

branching/fullstrong/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) 0  
 priority of branching rule <fullstrong>

branching/inference/maxbounddist ( $0 \leq \text{real} \leq 1$ ) 1  
 maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)

branching/inference/maxdepth ( $-1 \leq \text{integer}$ ) -1  
 maximal depth level, up to which branching rule <inference> should be used (-1 for no limit)

branching/inference/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) 1000  
 priority of branching rule <inference>

branching/inference/useweightedsum (boolean) TRUE  
 should a weighted sum of inference, conflict and cutoff weights be used?

branching/leastinf/maxbounddist ( $0 \leq \text{real} \leq 1$ ) 1  
 maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)

branching/leastinf/maxdepth ( $-1 \leq \text{integer}$ ) -1  
 maximal depth level, up to which branching rule <leastinf> should be used (-1 for no limit)

branching/leastinf/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) 50  
 priority of branching rule <leastinf>

branching/lpgainnormalize (character) s  
 strategy for normalization of LP gain when updating pseudocosts of continuous variables (divide by movement of 'l'p value, reduction in 'd'omain width, or reduction in domain width of 's'ibling)

branching/mostinf/maxbounddist ( $0 \leq \text{real} \leq 1$ ) 1  
 maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)

branching/mostinf/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level, up to which branching rule <mostinf> should be used (-1 for no limit)	
branching/mostinf/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	100
priority of branching rule <mostinf>	
branching/preferbinary (boolean)	FALSE
should branching on binary variables be preferred?	
branching/pscost/maxbounddist ( $0 \leq \text{real} \leq 1$ )	1
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)	
branching/pscost/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level, up to which branching rule <pscost> should be used (-1 for no limit)	
branching/pscost/narymaxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth where to do n-ary branching, -1 to turn off	
branching/pscost/naryminwidth ( $0 \leq \text{real} \leq 1$ )	0.001
minimal domain width in children when doing n-ary branching, relative to global bounds	
branching/pscost/narywidthfactor ( $1 \leq \text{real}$ )	2
factor of domain width in n-ary branching when creating nodes with increasing distance from branching value	
branching/pscost/nchildren ( $2 \leq \text{integer}$ )	2
number of children to create in n-ary branching	
branching/pscost/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	2000
priority of branching rule <pscost>	
branching/pscost/strategy (character)	u
strategy for utilizing pseudo-costs of external branching candidates (multiply as in pseudo costs 'u'pdate rule, or by 'd'omain reduction, or by domain reduction of 's'ibling, or by 'v'ariable score)	
branching/random/maxbounddist ( $0 \leq \text{real} \leq 1$ )	1
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)	
branching/random/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level, up to which branching rule <random> should be used (-1 for no limit)	
branching/random/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-100000
priority of branching rule <random>	
branching/random/seed ( $0 \leq \text{integer}$ )	0
initial random seed value	
branching/relpscost/initcand ( $0 \leq \text{integer}$ )	100
maximal number of candidates initialized with strong branching per node	
branching/relpscost/inititer ( $0 \leq \text{integer}$ )	0
iteration limit for strong branching initializations of pseudo cost entries (0: auto)	
branching/relpscost/maxbounddist ( $0 \leq \text{real} \leq 1$ )	1
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying branching rule (0.0: only on current best node, 1.0: on all nodes)	
branching/relpscost/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level, up to which branching rule <relpscost> should be used (-1 for no limit)	
branching/relpscost/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	10000
priority of branching rule <relpscost>	
branching/relpscost/sbiterofs ( $0 \leq \text{integer}$ )	100000
additional number of allowed strong branching LP iterations	
branching/relpscost/sbiterquot ( $0 \leq \text{real}$ )	0.5

maximal fraction of strong branching LP iterations compared to node relaxation LP iterations

### Branching (advanced options)

branching/fullstrong/reevalage ( $0 \leq \text{integer}$ )	10
number of intermediate LPs solved to trigger reevaluation of strong branching value for a variable that was already evaluated at the current node	
branching/inference/conflictweight ( $0 \leq \text{real}$ )	1000
weight in score calculations for conflict score	
branching/inference/cutoffweight ( $0 \leq \text{real}$ )	1
weight in score calculations for cutoff score	
branching/inference/fractionals (boolean)	TRUE
should branching on LP solution be restricted to the fractional variables?	
branching/inference/inferenceweight (real)	1
weight in score calculations for inference score	
branching/pscost/maxscoreweight (real)	1.3
weight for maximum of scores of a branching candidate when building weighted sum of min/max/sum of scores	
branching/pscost/minscoreweight (real)	0.8
weight for minimum of scores of a branching candidate when building weighted sum of min/max/sum of scores	
branching/pscost/sumscoreweight (real)	0.1
weight for sum of scores of a branching candidate when building weighted sum of min/max/sum of scores	
branching/relpscost/conflictlengthweight (real)	0
weight in score calculations for conflict length score	
branching/relpscost/conflictweight (real)	0.01
weight in score calculations for conflict score	
branching/relpscost/cutoffweight (real)	0.0001
weight in score calculations for cutoff score	
branching/relpscost/inferenceweight (real)	0.0001
weight in score calculations for inference score	
branching/relpscost/maxbdchgs ( $-1 \leq \text{integer}$ )	5
maximal number of bound tightenings before the node is reevaluated (-1: unlimited)	
branching/relpscost/maxlookahead ( $1 \leq \text{integer}$ )	8
maximal number of further variables evaluated without better score	
branching/relpscost/maxreliable ( $0 \leq \text{real}$ )	8
maximal value for minimum pseudo cost size to regard pseudo cost value as reliable	
branching/relpscost/minreliable ( $0 \leq \text{real}$ )	1
minimal value for minimum pseudo cost size to regard pseudo cost value as reliable	
branching/relpscost/pscostweight (real)	1
weight in score calculations for pseudo cost score	
branching/scorefac ( $0 \leq \text{real} \leq 1$ )	0.167
branching score factor to weigh downward and upward gain prediction in sum score function	
branching/scorefunc (character)	p
branching score function ('s'um, 'p'roduct)	

### Conflict analysis

conflict/bounddisjunction/continuousfrac ( $0 \leq \text{real} \leq 1$ )	0.4
maximal percentage of continuous variables within a conflict	
conflict/enable (boolean)	TRUE
should conflict analysis be enabled?	

conflict/preferbinary (boolean)	FALSE
should binary conflicts be preferred?	
conflict/restartfac ( $0 \leq \text{real}$ )	1.5
factor to increase restartnum with after each restart	
conflict/restartnum ( $0 \leq \text{integer}$ )	0
number of successful conflict analysis calls that trigger a restart (0: disable conflict restarts)	
conflict/useboundlp (boolean)	FALSE
should bound exceeding LP conflict analysis be used?	
conflict/useinfp (boolean)	TRUE
should infeasible LP conflict analysis be used?	
conflict/useprop (boolean)	TRUE
should propagation conflict analysis be used?	
conflict/usepseudo (boolean)	TRUE
should pseudo solution conflict analysis be used?	
conflict/usesb (boolean)	FALSE
should infeasible/bound exceeding strong branching conflict analysis be used?	

### Conflict analysis (advanced options)

conflict/allowlocal (boolean)	TRUE
should conflict constraints be generated that are only valid locally?	
conflict/bounddisjunction/priority (integer)	-3000000
priority of conflict handler <bounddisjunction>	
conflict/depthscorefac (real)	1
score factor for depth level in bound relaxation heuristic of LP analysis	
conflict/dynamic (boolean)	TRUE
should the conflict constraints be subject to aging?	
conflict/fuipleveles ( $-1 \leq \text{integer}$ )	-1
number of depth levels up to which first UIP's are used in conflict analysis (-1: use All-FirstUIP rule)	
conflict/ignorerelaxedbd (boolean)	FALSE
should relaxed bounds be ignored?	
conflict/indicatorconflict/priority (integer)	200000
priority of conflict handler <indicatorconflict>	
conflict/interconss ( $-1 \leq \text{integer}$ )	-1
maximal number of intermediate conflict constraints generated in conflict graph (-1: use every intermediate constraint)	
conflict/keepreprop (boolean)	TRUE
should constraints be kept for repropagation even if they are too long?	
conflict/linear/priority (integer)	-1000000
priority of conflict handler <linear>	
conflict/logicor/priority (integer)	800000
priority of conflict handler <logicor>	
conflict/lpiterations ( $-1 \leq \text{integer}$ )	10
maximal number of LP iterations in each LP resolving loop (-1: no limit)	
conflict/maxconss ( $-1 \leq \text{integer}$ )	10
maximal number of conflict constraints accepted at an infeasible node (-1: use all generated conflict constraints)	
conflict/maxlploops ( $-1 \leq \text{integer}$ )	2
maximal number of LP resolving loops during conflict analysis (-1: no limit)	
conflict/maxvarsfac ( $0 \leq \text{real}$ )	0.1



maximal fraction of variables involved in a conflict constraint	
conflict/minmaxvars ( $0 \leq \text{integer}$ )	30
minimal absolute maximum of variables involved in a conflict constraint	
conflict/reconvlevels ( $-1 \leq \text{integer}$ )	-1
number of depth levels up to which UIP reconvergence constraints are generated (-1: generate reconvergence constraints in all depth levels)	
conflict/removable (boolean)	TRUE
should the conflict's relaxations be subject to LP aging and cleanup?	
conflict/repropagate (boolean)	TRUE
should earlier nodes be repropagated in order to replace branching decisions by deductions?	
conflict/scorefac ( $10^{-6} \leq \text{real} \leq 1$ )	0.98
factor to decrease importance of variables' earlier conflict scores	
conflict/separate (boolean)	TRUE
should the conflict constraints be separated?	
conflict/setppc/priority (integer)	700000
priority of conflict handler <setppc>	
conflict/settlelocal (boolean)	FALSE
should conflict constraints be attached only to the local subtree where they can be useful?	

### Constraints

constraints/abspower/branchminconvererror (boolean)	FALSE
whether to compute branching point such that the convexification error is minimized (after branching on 0.0)	
constraints/abspower/cutmaxrange ( $0 \leq \text{real}$ )	$10^7$
maximal coef range of a cut (maximal coefficient divided by minimal coefficient) in order to be added to LP relaxation	
constraints/abspower/dualpresolve (boolean)	TRUE
should dual presolve be applied?	
constraints/abspower/linfeasshift (boolean)	TRUE
whether to try to make solutions in check function feasible by shifting the linear variable z	
constraints/abspower/minefficacyenfofac ( $1 \leq \text{real}$ )	2
minimal target efficacy of a cut in order to add it to relaxation during enforcement as factor of feasibility tolerance (may be ignored)	
constraints/abspower/minefficacysepa ( $0 \leq \text{real}$ )	0.0001
minimal efficacy for a cut to be added to the LP during separation; overwrites separating/efficacy	
constraints/abspower/preferzerobranh ( $0 \leq \text{integer} \leq 3$ )	1
how much to prefer branching on 0.0 when sign of variable is not fixed yet: 0 no preference, 1 prefer if LP solution will be cutoff in both child nodes, 2 prefer always, 3 ensure always	
constraints/abspower/projectrefpoint (boolean)	TRUE
whether to project the reference point when linearizing an absolute power constraint in a convex region	
constraints/abspower/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/abspower/sepaefreq ( $-1 \leq \text{integer}$ )	1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/abspower/sepainboundsonly (boolean)	FALSE
whether to separate linearization cuts only in the variable bounds (does not affect enforcement)	
constraints/abspower/sepanlpmincont ( $0 \leq \text{real} \leq 2$ )	1
minimal required fraction of continuous variables in problem to use solution of NLP relaxation in root for separation	
constraints/and/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	

constraints/and/sepaftereq ( $-1 \leq \text{integer}$ )	1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/bivariate/linfeasshift (boolean)	TRUE
whether to try to make solutions in check function feasible by shifting a linear variable (esp. useful if constraint was actually objective function)	
constraints/bivariate/maxpropounds ( $0 \leq \text{integer}$ )	1
limit on number of propagation rounds for a single constraint within one round of SCIP propagation	
constraints/bivariate/minefficacyenfo ( $0 \leq \text{real}$ )	$2 \cdot 10^{-6}$
minimal target efficacy of a cut in order to add it to relaxation during enforcement (may be ignored)	
constraints/bivariate/minefficacysepa ( $0 \leq \text{real}$ )	0.0001
minimal efficacy for a cut to be added to the LP during separation; overwrites separating/efficacy	
constraints/bivariate/ninitlprefpoints ( $0 \leq \text{integer}$ )	3
number of reference points in each direction where to compute linear support for envelope in LP initialization	
constraints/bivariate/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/bivariate/sepaftereq ( $-1 \leq \text{integer}$ )	1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/bounddisjunction/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/bounddisjunction/sepaftereq ( $-1 \leq \text{integer}$ )	-1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/indicator/maxsepacuts ( $0 \leq \text{integer}$ )	100
maximal number of cuts separated per separation round	
constraints/indicator/maxsepacutsroot ( $0 \leq \text{integer}$ )	2000
maximal number of cuts separated per separation round in the root node	
constraints/indicator/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/indicator/sepaftereq ( $-1 \leq \text{integer}$ )	10
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/integral/propfreq ( $-1 \leq \text{integer}$ )	-1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/integral/sepaftereq ( $-1 \leq \text{integer}$ )	-1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/knapsack/maxrounds ( $-1 \leq \text{integer}$ )	5
maximal number of separation rounds per node (-1: unlimited)	
constraints/knapsack/maxroundsroot ( $-1 \leq \text{integer}$ )	-1
maximal number of separation rounds per node in the root node (-1: unlimited)	
constraints/knapsack/maxsepacuts ( $0 \leq \text{integer}$ )	50
maximal number of cuts separated per separation round	
constraints/knapsack/maxsepacutsroot ( $0 \leq \text{integer}$ )	200
maximal number of cuts separated per separation round in the root node	
constraints/knapsack/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/knapsack/sepaftereq ( $-1 \leq \text{integer}$ )	0
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/linear/maxrounds ( $-1 \leq \text{integer}$ )	5
maximal number of separation rounds per node (-1: unlimited)	

constraints/linear/maxroundsroot ( $-1 \leq \text{integer}$ )	-1
maximal number of separation rounds per node in the root node (-1: unlimited)	
constraints/linear/maxsepacuts ( $0 \leq \text{integer}$ )	50
maximal number of cuts separated per separation round	
constraints/linear/maxsepacutsroot ( $0 \leq \text{integer}$ )	200
maximal number of cuts separated per separation round in the root node	
constraints/linear/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/linear/sepafreq ( $-1 \leq \text{integer}$ )	0
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/linear/separateall (boolean)	FALSE
should all constraints be subject to cardinality cut generation instead of only the ones with non-zero dual value?	
constraints/linear/upgrade/knapsack (boolean)	TRUE
enable linear upgrading for constraint handler <knapsack>	
constraints/linear/upgrade/logicor (boolean)	TRUE
enable linear upgrading for constraint handler <logicor>	
constraints/linear/upgrade/setppc (boolean)	TRUE
enable linear upgrading for constraint handler <setppc>	
constraints/linear/upgrade/varbound (boolean)	TRUE
enable linear upgrading for constraint handler <varbound>	
constraints/logicor/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/logicor/sepafreq ( $-1 \leq \text{integer}$ )	0
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/nonlinear/cutmaxrange ( $0 \leq \text{real}$ )	$10^7$
maximal coef range of a cut (maximal coefficient divided by minimal coefficient) in order to be added to LP relaxation	
constraints/nonlinear/linfeasshift (boolean)	TRUE
whether to try to make solutions in check function feasible by shifting a linear variable (esp. useful if constraint was actually objective function)	
constraints/nonlinear/maxpropounds ( $0 \leq \text{integer}$ )	1
limit on number of propagation rounds for a single constraint within one round of SCIP propagation	
constraints/nonlinear/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/nonlinear/reformulate (boolean)	TRUE
whether to reformulate expression graph	
constraints/nonlinear/sepafreq ( $-1 \leq \text{integer}$ )	1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/nonlinear/sepanlpmincont ( $0 \leq \text{real} \leq 2$ )	1
minimal required fraction of continuous variables in problem to use solution of NLP relaxation in root for separation	
constraints/nonlinear/upgrade/abspower (boolean)	TRUE
enable nonlinear upgrading for constraint handler <abspower>	
constraints/nonlinear/upgrade/and (boolean)	TRUE
enable nonlinear upgrading for constraint handler <and>	
constraints/nonlinear/upgrade/bivariate (boolean)	FALSE
enable nonlinear upgrading for constraint handler <bivariate>	
constraints/nonlinear/upgrade/linear (boolean)	TRUE
enable nonlinear upgrading for constraint handler <linear>	

constraints/nonlinear/upgrade/quadratic (boolean)	TRUE
enable nonlinear upgrading for constraint handler <quadratic>	
constraints/quadratic/checkcurvature (boolean)	TRUE
whether multivariate quadratic functions should be checked for convexity/concavity	
constraints/quadratic/empathy4and ( $0 \leq \text{integer} \leq 2$ )	0
empathy level for using the AND constraint handler: 0 always avoid using AND; 1 use AND sometimes; 2 use AND as often as possible	
constraints/quadratic/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/quadratic/replacebinaryprod ( $0 \leq \text{integer}$ )	$\infty$
max. length of linear term which when multiplied with a binary variables is replaced by an auxiliary variable and a linear reformulation (0 to turn off)	
constraints/quadratic/sepaftereq ( $-1 \leq \text{integer}$ )	1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/quadratic/sepanlpmincont ( $0 \leq \text{real} \leq 2$ )	1
minimal required fraction of continuous variables in problem to use solution of NLP relaxation in root for separation	
constraints/quadratic/upgrade/abspower (boolean)	TRUE
enable quadratic upgrading for constraint handler <abspower>	
constraints/quadratic/upgrade/bivariate (boolean)	FALSE
enable quadratic upgrading for constraint handler <bivariate>	
constraints/quadratic/upgrade/bounddisjunction (boolean)	TRUE
enable quadratic upgrading for constraint handler <bounddisjunction>	
constraints/quadratic/upgrade/linear (boolean)	TRUE
enable quadratic upgrading for constraint handler <linear>	
constraints/quadratic/upgrade/soc (boolean)	TRUE
enable quadratic upgrading for constraint handler <soc>	
constraints/setppc/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/setppc/sepaftereq ( $-1 \leq \text{integer}$ )	0
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/soc/glineur (boolean)	TRUE
whether the Glineur Outer Approximation should be used instead of Ben-Tal Nemirovski	
constraints/soc/linfeasshift (boolean)	TRUE
whether to try to make solutions feasible in check by shifting the variable on the right hand side	
constraints/soc/minefficacy ( $0 \leq \text{real}$ )	0.0001
minimal efficacy of a cut to be added to LP in separation	
constraints/soc/nauxvars ( $0 \leq \text{integer}$ )	0
number of auxiliary variables to use when creating a linear outer approx. of a SOC3 constraint; 0 to turn off	
constraints/soc/nlpform (character)	a
which formulation to use when adding a SOC constraint to the NLP (a: automatic, q: nonconvex quadratic form, s: convex sqrt form, e: convex exponential-sqrt form, d: convex division form)	
constraints/soc/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/soc/sepaftereq ( $-1 \leq \text{integer}$ )	1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/soc/sepanlpmincont ( $0 \leq \text{real} \leq 2$ )	1
minimal required fraction of continuous variables in problem to use solution of NLP relaxation in root for separation	

constraints/SOS1/branchnonzeros (boolean)	FALSE
Branch on SOS constraint with most number of nonzeros?	
constraints/SOS1/branchsos (boolean)	TRUE
Use SOS1 branching in enforcing (otherwise leave decision to branching rules)?	
constraints/SOS1/branchweight (boolean)	FALSE
Branch on SOS cons. with highest nonzero-variable weight for branching (needs branchnonzeros = false)?	
constraints/SOS1/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/SOS1/sepaftereq ( $-1 \leq \text{integer}$ )	0
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/SOS2/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/SOS2/sepaftereq ( $-1 \leq \text{integer}$ )	0
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/superindicator/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/superindicator/sepaftereq ( $-1 \leq \text{integer}$ )	-1
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/varbound/propfreq ( $-1 \leq \text{integer}$ )	1
frequency for propagating domains (-1: never, 0: only in root node)	
constraints/varbound/sepaftereq ( $-1 \leq \text{integer}$ )	0
frequency for separating cuts (-1: never, 0: only in root node)	
constraints/varbound/usebdwidening (boolean)	TRUE
should bound widening be used in conflict analysis?	
<b>Constraints (advanced options)</b>	
constraints/abspower/addvarboundcons (boolean)	TRUE
should variable bound constraints be added for derived variable bounds?	
constraints/abspower/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/abspower/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/abspower/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/abspower/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/abspower/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/abspower/timingmask ( $1 \leq \text{integer} \leq 15$ )	15
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALLWAYS)	
constraints/agelimit ( $-1 \leq \text{integer}$ )	0
maximum age an unnecessary constraint can reach before it is deleted (0: dynamic, -1: keep all constraints)	
constraints/and/aggrlinearization (boolean)	FALSE
should an aggregated linearization be used?	
constraints/and/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	

<code>constraints/and/delayprop</code> (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
<code>constraints/and/delaysepa</code> (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
<code>constraints/and/dualpresolving</code> (boolean)	TRUE
should dual presolving be performed?	
<code>constraints/and/eagerfreq</code> ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
<code>constraints/and/enforcecuts</code> (boolean)	TRUE
should cuts be separated during LP enforcing?	
<code>constraints/and/linearize</code> (boolean)	FALSE
should the "and" constraint get linearized and removed (in presolving)?	
<code>constraints/and/maxprerounds</code> ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
<code>constraints/and/presolpairwise</code> (boolean)	TRUE
should pairwise constraint comparison be performed in presolving?	
<code>constraints/and/presolusehashing</code> (boolean)	TRUE
should hash table be used for detecting redundant constraints in advance	
<code>constraints/and/timingmask</code> ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
<code>constraints/bivariate/cutmaxrange</code> ( $0 \leq \text{real}$ )	$10^7$
maximal coef range of a cut (maximal coefficient divided by minimal coefficient) in order to be added to LP relaxation	
<code>constraints/bivariate/delaypresol</code> (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
<code>constraints/bivariate/delayprop</code> (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
<code>constraints/bivariate/delaysepa</code> (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
<code>constraints/bivariate/eagerfreq</code> ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
<code>constraints/bivariate/maxprerounds</code> ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
<code>constraints/bivariate/timingmask</code> ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
<code>constraints/bounddisjunction/delaypresol</code> (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
<code>constraints/bounddisjunction/delayprop</code> (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
<code>constraints/bounddisjunction/delaysepa</code> (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
<code>constraints/bounddisjunction/eagerfreq</code> ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	

constraints/bounddisjunction/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/bounddisjunction/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/disableenfops (boolean)	FALSE
should enforcement of pseudo solution be disabled?	
constraints/indicator/addcoupling (boolean)	TRUE
Add coupling constraints if big-M is small enough?	
constraints/indicator/addcouplingcons (boolean)	FALSE
Add initial coupling inequalities as linear constraints, if 'addcoupling' is true?	
constraints/indicator/addopposite (boolean)	FALSE
Add opposite inequality in nodes in which the binary variable has been fixed to 0?	
constraints/indicator/branchindicators (boolean)	FALSE
Branch on indicator constraints in enforcing?	
constraints/indicator/conflictsupgrade (boolean)	FALSE
Try to upgrade bounddisjunction conflicts by replacing slack variables?	
constraints/indicator/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/indicator/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/indicator/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/indicator/dualreductions (boolean)	TRUE
should dual reduction steps be performed?	
constraints/indicator/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/indicator/enforcecuts (boolean)	FALSE
In enforcing try to generate cuts (only if sepaalternativelp is true)?	
constraints/indicator/forcerestart (boolean)	FALSE
force restart if we have a max FS instance and gap is 1?	
constraints/indicator/generatebilinear (boolean)	FALSE
Do not generate indicator constraint, but a bilinear constraint instead?	
constraints/indicator/genlogicor (boolean)	FALSE
Generate logicor constraints instead of cuts?	
constraints/indicator/maxconditionaltlp ( $0 \leq \text{real}$ )	0
maximum estimated condition of the solution basis matrix of the alternative LP to be trustworthy (0.0 to disable check)	
constraints/indicator/maxcouplingvalue ( $0 \leq \text{real} \leq 10^9$ )	10000
maximum coefficient for binary variable in coupling constraint	
constraints/indicator/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/indicator/nolinconscont (boolean)	FALSE
decompose problem - do not generate linear constraint if all variables are continuous	
constraints/indicator/removeindicators (boolean)	FALSE
Remove indicator constraint if corresponding variable bound constraint has been added?	
constraints/indicator/restartfrac ( $0 \leq \text{real} \leq 1$ )	0.9

fraction of binary variables that need to be fixed before restart occurs (in forcerestart)	
constraints/indicator/sepaalternativelp (boolean) Separate using the alternative LP?	FALSE
constraints/indicator/sepacouplingcuts (boolean) Should the coupling inequalities be separated dynamically?	FALSE
constraints/indicator/sepacouplinglocal (boolean) Allow to use local bounds in order to separated coupling inequalities?	FALSE
constraints/indicator/sepacouplingvalue ( $0 \leq \text{real} \leq 10^9$ ) maximum coefficient for binary variable in separated coupling constraint	10000
constraints/indicator/timingmask ( $1 \leq \text{integer} \leq 15$ ) timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	1
constraints/indicator/trysolutions (boolean) Try to make solutions feasible by setting indicator variables?	TRUE
constraints/indicator/updatebounds (boolean) Update bounds of original variables for separation?	FALSE
constraints/integral/delaypresol (boolean) should presolving method be delayed, if other presolvers found reductions?	FALSE
constraints/integral/delayprop (boolean) should propagation method be delayed, if other propagators found reductions?	FALSE
constraints/integral/delaysepa (boolean) should separation method be delayed, if other separators found cuts?	FALSE
constraints/integral/eagerfreq ( $-1 \leq \text{integer}$ ) frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	-1
constraints/integral/maxprerounds ( $-1 \leq \text{integer}$ ) maximal number of presolving rounds the constraint handler participates in (-1: no limit)	0
constraints/integral/timingmask ( $1 \leq \text{integer} \leq 15$ ) timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	1
constraints/knapsack/delaypresol (boolean) should presolving method be delayed, if other presolvers found reductions?	FALSE
constraints/knapsack/delayprop (boolean) should propagation method be delayed, if other propagators found reductions?	FALSE
constraints/knapsack/delaysepa (boolean) should separation method be delayed, if other separators found cuts?	FALSE
constraints/knapsack/disaggregation (boolean) should disaggregation of knapsack constraints be allowed in preprocessing?	TRUE
constraints/knapsack/dualpresolving (boolean) should dual presolving steps be performed?	TRUE
constraints/knapsack/eagerfreq ( $-1 \leq \text{integer}$ ) frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	100
constraints/knapsack/maxcardbounddist ( $0 \leq \text{real} \leq 1$ ) maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for separating knapsack cuts	0
constraints/knapsack/maxprerounds ( $-1 \leq \text{integer}$ )	-1



maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/knapsack/negatedclique (boolean) should negated clique information be used in solving process	TRUE
constraints/knapsack/presolpairwise (boolean) should pairwise constraint comparison be performed in presolving?	TRUE
constraints/knapsack/presolusehashing (boolean) should hash table be used for detecting redundant constraints in advance	TRUE
constraints/knapsack/sepacardfreq ( $-1 \leq \text{integer}$ ) multiplier on separation frequency, how often knapsack cuts are separated (-1: never, 0: only at root)	1
constraints/knapsack/simplifyinequalities (boolean) should presolving try to simplify knapsacks	TRUE
constraints/knapsack/timingmask ( $1 \leq \text{integer} \leq 15$ ) timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	1
constraints/knapsack/usegubs (boolean) should GUB information be used for separation?	FALSE
constraints/linear/aggregatevariables (boolean) should presolving search for aggregations in equations	TRUE
constraints/linear/delaypresol (boolean) should presolving method be delayed, if other presolvers found reductions?	FALSE
constraints/linear/delayprop (boolean) should propagation method be delayed, if other propagators found reductions?	FALSE
constraints/linear/delaysepa (boolean) should separation method be delayed, if other separators found cuts?	FALSE
constraints/linear/dualpresolving (boolean) should dual presolving steps be performed?	TRUE
constraints/linear/eagerfreq ( $-1 \leq \text{integer}$ ) frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	100
constraints/linear/maxaggrnormscale ( $0 \leq \text{real}$ ) maximal allowed relative gain in maximum norm for constraint aggregation (0.0: disable constraint aggregation)	0
constraints/linear/maxcardbounddist ( $0 \leq \text{real} \leq 1$ ) maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for separating knapsack cardinality cuts	0
constraints/linear/maxprerounds ( $-1 \leq \text{integer}$ ) maximal number of presolving rounds the constraint handler participates in (-1: no limit)	-1
constraints/linear/mingainpernmincomparisons ( $0 \leq \text{real}$ ) minimal gain per minimal pairwise presolve comparisons to repeat pairwise comparison round	$10^{-6}$
constraints/linear/nmincomparisons ( $1 \leq \text{integer}$ ) number for minimal pairwise presolve comparisons	200000
constraints/linear/presolpairwise (boolean) should pairwise constraint comparison be performed in presolving?	TRUE
constraints/linear/presolusehashing (boolean) should hash table be used for detecting redundant constraints in advance	TRUE
constraints/linear/simplifyinequalities (boolean) should presolving try to simplify inequalities	TRUE
constraints/linear/sortvars (boolean)	TRUE

apply binaries sorting in decr. order of coeff abs value?

constraints/linear/tightenboundsfreq ( $-1 \leq \text{integer}$ )	1
multiplier on propagation frequency, how often the bounds are tightened (-1: never, 0: only at root)	
constraints/linear/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/logicor/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/logicor/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/logicor/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/logicor/dualpresolving (boolean)	TRUE
should dual presolving steps be performed?	
constraints/logicor/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/logicor/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/logicor/negatedclique (boolean)	TRUE
should negated clique information be used in presolving	
constraints/logicor/presolpairwise (boolean)	TRUE
should pairwise constraint comparison be performed in presolving?	
constraints/logicor/presolusehashing (boolean)	TRUE
should hash table be used for detecting redundant constraints in advance	
constraints/logicor/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/nonlinear/assumeconvex (boolean)	FALSE
whether to assume that nonlinear functions in inequalities ( $\leq$ ) are convex (disables reformulation)	
constraints/nonlinear/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/nonlinear/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/nonlinear/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/nonlinear/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/nonlinear/maxexpansionexponent ( $1 \leq \text{integer}$ )	2
maximal exponent where still expanding non-monomial polynomials in expression simplification	
constraints/nonlinear/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/nonlinear/minefficacyenfofac ( $1 \leq \text{real}$ )	2
minimal target efficacy of a cut in order to add it to relaxation during enforcement as a factor of the feasibility tolerance (may be ignored)	
constraints/nonlinear/minefficacysepa ( $0 \leq \text{real}$ )	0.0001

minimal efficacy for a cut to be added to the LP during separation; overwrites separating/efficacy	
constraints/nonlinear/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/obsoleteage ( $-1 \leq \text{integer}$ )	-1
age of a constraint after which it is marked obsolete (0: dynamic, -1 do not mark constraints obsolete)	
constraints/quadratic/binreforminitial (boolean)	FALSE
whether to make non-varbound linear constraints added due to replacing products with binary variables initial	
constraints/quadratic/checkfactorable (boolean)	TRUE
whether constraint functions should be checked to be factorable	
constraints/quadratic/cutmaxrange ( $0 \leq \text{real}$ )	10 <sup>7</sup>
maximal coef range of a cut (maximal coefficient divided by minimal coefficient) in order to be added to LP relaxation	
constraints/quadratic/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/quadratic/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/quadratic/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/quadratic/disaggregate (boolean)	FALSE
whether to disaggregate quadratic parts that decompose into a sum of non-overlapping quadratic terms	
constraints/quadratic/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/quadratic/linearizeheursol (boolean)	TRUE
whether linearizations of convex quadratic constraints should be added to cutpool in a solution found by some heuristic	
constraints/quadratic/linfeasshift (boolean)	TRUE
whether to try to make solutions in check function feasible by shifting a linear variable (esp. useful if constraint was actually objective function)	
constraints/quadratic/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/quadratic/maxpropounds ( $0 \leq \text{integer}$ )	1
limit on number of propagation rounds for a single constraint within one round of SCIP propagation during solve	
constraints/quadratic/maxpropoundspresolve ( $0 \leq \text{integer}$ )	10
limit on number of propagation rounds for a single constraint within one round of SCIP presolve	
constraints/quadratic/minefficacyenfofac ( $1 \leq \text{real}$ )	2
minimal target efficacy of a cut in order to add it to relaxation during enforcement as a factor of the feasibility tolerance (may be ignored)	
constraints/quadratic/minefficacysepa ( $0 \leq \text{real}$ )	0.0001
minimal efficacy for a cut to be added to the LP during separation; overwrites separating/efficacy	
constraints/quadratic/scaling (boolean)	TRUE
whether a quadratic constraint should be scaled w.r.t. the current gradient norm when checking for feasibility	
constraints/quadratic/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/setppc/addvariablesascliques (boolean)	FALSE
should we try to generate extra cliques out of all binary variables to maybe fasten redundant constraint detection	
constraints/setppc/cliquelifting (boolean)	FALSE

should we try to lift variables into other clique constraints, fix variables, aggregate them, and also shrink the amount of variables in clique constraints

constraints/setppc/cliquesthrinking (boolean) TRUE

should we try to shrink the number of variables in a clique constraints, by replacing more than one variable by only one

constraints/setppc/delaypresol (boolean) FALSE

should presolving method be delayed, if other presolvers found reductions?

constraints/setppc/delayprop (boolean) FALSE

should propagation method be delayed, if other propagators found reductions?

constraints/setppc/delaysepa (boolean) FALSE

should separation method be delayed, if other separators found cuts?

constraints/setppc/dualpresolving (boolean) TRUE

should dual presolving steps be performed?

constraints/setppc/eagerfreq ( $-1 \leq \text{integer}$ ) 100

frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)

constraints/setppc/maxprerounds ( $-1 \leq \text{integer}$ ) -1

maximal number of presolving rounds the constraint handler participates in (-1: no limit)

constraints/setppc/npseudobranches ( $0 \leq \text{integer}$ ) 2

number of children created in pseudo branching (0: disable pseudo branching)

constraints/setppc/presolpairwise (boolean) TRUE

should pairwise constraint comparison be performed in presolving?

constraints/setppc/presolusehashing (boolean) TRUE

should hash table be used for detecting redundant constraints in advance

constraints/setppc/timingmask ( $1 \leq \text{integer} \leq 15$ ) 1

timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALLWAYS))

constraints/soc/delaypresol (boolean) FALSE

should presolving method be delayed, if other presolvers found reductions?

constraints/soc/delayprop (boolean) FALSE

should propagation method be delayed, if other propagators found reductions?

constraints/soc/delaysepa (boolean) FALSE

should separation method be delayed, if other separators found cuts?

constraints/soc/eagerfreq ( $-1 \leq \text{integer}$ ) 100

frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)

constraints/soc/maxprerounds ( $-1 \leq \text{integer}$ ) -1

maximal number of presolving rounds the constraint handler participates in (-1: no limit)

constraints/soc/projectpoint (boolean) FALSE

whether the reference point of a cut should be projected onto the feasible set of the SOC constraint

constraints/soc/scaling (boolean) TRUE

whether a constraint should be scaled w.r.t. the current gradient norm when checking for feasibility

constraints/soc/sparsify (boolean) FALSE

whether to sparsify cuts

constraints/soc/sparsifymaxloss ( $0 \leq \text{real} \leq 1$ ) 0.2

maximal loss in cut efficacy by sparsification

constraints/soc/sparsifynzgrowth ( $1 \leq \text{real}$ ) 1.3

growth rate of maximal allowed nonzeros in cuts in sparsification

constraints/soc/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/SOS1/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/SOS1/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/SOS1/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/SOS1/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/SOS1/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/SOS1/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/SOS2/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/SOS2/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/SOS2/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/SOS2/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/SOS2/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/SOS2/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
constraints/superindicator/checkslacktype (boolean)	TRUE
should type of slack constraint be checked when creating superindicator constraint?	
constraints/superindicator/delaypresol (boolean)	FALSE
should presolving method be delayed, if other presolvers found reductions?	
constraints/superindicator/delayprop (boolean)	FALSE
should propagation method be delayed, if other propagators found reductions?	
constraints/superindicator/delaysepa (boolean)	FALSE
should separation method be delayed, if other separators found cuts?	
constraints/superindicator/eagerfreq ( $-1 \leq \text{integer}$ )	100
frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	
constraints/superindicator/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the constraint handler participates in (-1: no limit)	
constraints/superindicator/maxupgdcoeflinear ( $0 \leq \text{real} \leq 10^{15}$ )	10000
maximum big-M coefficient of binary variable in upgrade to a linear constraint (relative to smallest coefficient)	
constraints/superindicator/timingmask ( $1 \leq \text{integer} \leq 15$ )	1
timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:AL-	

WAYS))

constraints/superindicator/upgdprioindicator ( $-1 \leq \text{integer}$ ) priority for upgrading to an indicator constraint (-1: never)	1
constraints/superindicator/upgdprioinear ( $-1 \leq \text{integer}$ ) priority for upgrading to an indicator constraint (-1: never)	2
constraints/varbound/delaypresol (boolean) should presolving method be delayed, if other presolvers found reductions?	FALSE
constraints/varbound/delayprop (boolean) should propagation method be delayed, if other propagators found reductions?	FALSE
constraints/varbound/delaysepa (boolean) should separation method be delayed, if other separators found cuts?	FALSE
constraints/varbound/eagerfreq ( $-1 \leq \text{integer}$ ) frequency for using all instead of only the useful constraints in separation, propagation and enforcement (-1: never, 0: only in first evaluation)	100
constraints/varbound/maxlpcoef ( $0 \leq \text{real}$ ) maximum coefficient in varbound constraint to be added as a row into LP	$10^6$
constraints/varbound/maxprerounds ( $-1 \leq \text{integer}$ ) maximal number of presolving rounds the constraint handler participates in (-1: no limit)	-1
constraints/varbound/presolpairwise (boolean) should pairwise constraint comparison be performed in presolving?	TRUE
constraints/varbound/timingmask ( $1 \leq \text{integer} \leq 15$ ) timing when constraint propagation should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	1

## Output

display/avgdualbound/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <avgdualbound> (0: off, 1: auto, 2:on)	1
display/conflicts/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <conflicts> (0: off, 1: auto, 2:on)	1
display/conss/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <conss> (0: off, 1: auto, 2:on)	1
display/curcols/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <curcols> (0: off, 1: auto, 2:on)	1
display/curconss/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <curconss> (0: off, 1: auto, 2:on)	1
display/curdualbound/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <curdualbound> (0: off, 1: auto, 2:on)	1
display/currows/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <currows> (0: off, 1: auto, 2:on)	1
display/cutoffbound/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <cutoffbound> (0: off, 1: auto, 2:on)	1
display/cuts/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <cuts> (0: off, 1: auto, 2:on)	1
display/depth/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <depth> (0: off, 1: auto, 2:on)	1
display/dualbound/active ( $0 \leq \text{integer} \leq 2$ ) display activation status of display column <dualbound> (0: off, 1: auto, 2:on)	1

display/estimate/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <estimate> (0: off, 1: auto, 2:on)	
display/feasST/active ( $0 \leq \text{integer} \leq 2$ )	0
display activation status of display column <feasST> (0: off, 1: auto, 2:on)	
display/freq ( $-1 \leq \text{integer}$ )	100
frequency for displaying node information lines	
display/gap/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <gap> (0: off, 1: auto, 2:on)	
display/headerfreq ( $-1 \leq \text{integer}$ )	15
frequency for displaying header lines (every n'th node information line)	
display/lpavgiterations/active ( $0 \leq \text{integer} \leq 2$ )	1 (0 under Windows)
display activation status of display column <lpavgiterations> (0: off, 1: auto, 2:on)	
display/lpcond/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <lpcond> (0: off, 1: auto, 2:on)	
display/lpinfo (boolean)	FALSE
should the LP solver display status messages?	
display/lpiterations/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <lpiterations> (0: off, 1: auto, 2:on)	
display/lpobj/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <lpobj> (0: off, 1: auto, 2:on)	
display/maxdepth/active ( $0 \leq \text{integer} \leq 2$ )	1 (0 under Windows)
display activation status of display column <maxdepth> (0: off, 1: auto, 2:on)	
display/memused/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <memused> (0: off, 1: auto, 2:on)	
display/nexternbranchcands/active ( $0 \leq \text{integer} \leq 2$ )	1 (2 for nonlinear instances)
display activation status of display column <nexternbranchcands> (0: off, 1: auto, 2:on)	
display/nfrac/active ( $0 \leq \text{integer} \leq 2$ )	1 (2 if discrete variables)
display activation status of display column <nfrac> (0: off, 1: auto, 2:on)	
display/nnodes/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <nnodes> (0: off, 1: auto, 2:on)	
display/nodesleft/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <nodesleft> (0: off, 1: auto, 2:on)	
display/nsols/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <nsols> (0: off, 1: auto, 2:on)	
display/plungedepth/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <plungedepth> (0: off, 1: auto, 2:on)	
display/poolsize/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <poolsize> (0: off, 1: auto, 2:on)	
display/primalbound/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <primalbound> (0: off, 1: auto, 2:on)	
display/primalgap/active ( $0 \leq \text{integer} \leq 2$ )	0
display activation status of display column <primalgap> (0: off, 1: auto, 2:on)	
display/pseudoobj/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <pseudoobj> (0: off, 1: auto, 2:on)	
display/separounds/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <separounds> (0: off, 1: auto, 2:on)	

display/solfound/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <solfound> (0: off, 1: auto, 2:on)	
display/sols/active ( $0 \leq \text{integer} \leq 2$ )	0
display activation status of display column <sols> (0: off, 1: auto, 2:on)	
display/statistics (boolean)	FALSE
whether to print statistics on a solve	
display/strongbranches/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <strongbranches> (0: off, 1: auto, 2:on)	
display/time/active ( $0 \leq \text{integer} \leq 2$ )	1 (2 under Windows)
display activation status of display column <time> (0: off, 1: auto, 2:on)	
display/vars/active ( $0 \leq \text{integer} \leq 2$ )	1
display activation status of display column <vars> (0: off, 1: auto, 2:on)	
display/verblevel ( $0 \leq \text{integer} \leq 5$ )	4
verbosity level of output	
display/width ( $0 \leq \text{integer}$ )	139 (80 under Windows)
maximal number of characters in a node information line	

### Heuristics

heuristics/actconsdiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/actconsdiving/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <actconsdiving> (-1: never, 0: only at depth freqofs)	
heuristics/actconsdiving/freqofs ( $0 \leq \text{integer}$ )	5
frequency offset for calling primal heuristic <actconsdiving>	
heuristics/actconsdiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/actconsdiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/cliquote/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <cliquote> (-1: never, 0: only at depth freqofs)	
heuristics/cliquote/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <cliquote>	
heuristics/cliquote/minfixingrate ( $0 \leq \text{real} \leq 1$ )	0.5
minimum percentage of integer variables that have to be fixable	
heuristics/cliquote/nodesofs ( $0 \leq \text{integer}$ )	500
number of nodes added to the contingent of the total nodes	
heuristics/cliquote/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/coefdiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/coefdiving/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <coefdiving> (-1: never, 0: only at depth freqofs)	
heuristics/coefdiving/freqofs ( $0 \leq \text{integer}$ )	1
frequency offset for calling primal heuristic <coefdiving>	
heuristics/coefdiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/coefdiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05



maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/crossover/freq ( $-1 \leq \text{integer}$ )	30
frequency for calling primal heuristic <crossover> (-1: never, 0: only at depth freqofs)	
heuristics/crossover/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <crossover>	
heuristics/crossover/minfixingrate ( $0 \leq \text{real} \leq 1$ )	0.666
minimum percentage of integer variables that have to be fixed	
heuristics/crossover/nodesofs ( $0 \leq \text{integer}$ )	500
number of nodes added to the contingent of the total nodes	
heuristics/crossover/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/crossover/nusedsols ( $2 \leq \text{integer}$ )	3
number of solutions to be taken into account	
heuristics/dins/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <dins> (-1: never, 0: only at depth freqofs)	
heuristics/dins/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <dins>	
heuristics/dins/minnodes ( $0 \leq \text{integer}$ )	500
minimum number of nodes required to start the subproblem	
heuristics/dins/neighborhoodsize ( $1 \leq \text{integer}$ )	18
radius (using Manhattan metric) of the incumbent's neighborhood to be searched	
heuristics/dins/nodesofs ( $0 \leq \text{integer}$ )	5000
number of nodes added to the contingent of the total nodes	
heuristics/dins/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.05
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/dins/solnum ( $1 \leq \text{integer}$ )	5
number of pool-solutions to be checked for flag array update (for hard fixing of binary variables)	
heuristics/feaspump/alphadiff ( $0 \leq \text{real} \leq 1$ )	1
threshold difference for the convex parameter to perform perturbation	
heuristics/feaspump/beforecuts (boolean)	TRUE
should the feasibility pump be called at root node before cut separation?	
heuristics/feaspump/freq ( $-1 \leq \text{integer}$ )	20
frequency for calling primal heuristic <feaspump> (-1: never, 0: only at depth freqofs)	
heuristics/feaspump/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <feaspump>	
heuristics/feaspump/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/feaspump/maxlpiterquot ( $0 \leq \text{real}$ )	0.01
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/feaspump/neighborhoodsize ( $1 \leq \text{integer}$ )	18
radius (using Manhattan metric) of the neighborhood to be searched in stage 3	
heuristics/feaspump/objfactor ( $0 \leq \text{real} \leq 1$ )	1
factor by which the regard of the objective is decreased in each round, 1.0 for dynamic	
heuristics/feaspump/pertsolfound (boolean)	TRUE
should a random perturbation be performed if a feasible solution was found?	
heuristics/feaspump/stage3 (boolean)	FALSE

should we solve a local branching sub-MIP if no solution could be found?	
heuristics/feaspump/usefp20 (boolean)	FALSE
should an iterative round-and-propagate scheme be used to find the integral points?	
heuristics/fixandinfer/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <fixandinfer> (-1: never, 0: only at depth freqofs)	
heuristics/fixandinfer/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <fixandinfer>	
heuristics/fracdiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/fracdiving/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <fracdiving> (-1: never, 0: only at depth freqofs)	
heuristics/fracdiving/freqofs ( $0 \leq \text{integer}$ )	3
frequency offset for calling primal heuristic <fracdiving>	
heuristics/fracdiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/fracdiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/guideddiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/guideddiving/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <guideddiving> (-1: never, 0: only at depth freqofs)	
heuristics/guideddiving/freqofs ( $0 \leq \text{integer}$ )	7
frequency offset for calling primal heuristic <guideddiving>	
heuristics/guideddiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/guideddiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/intdiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/intdiving/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <intdiving> (-1: never, 0: only at depth freqofs)	
heuristics/intdiving/freqofs ( $0 \leq \text{integer}$ )	9
frequency offset for calling primal heuristic <intdiving>	
heuristics/intdiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/intdiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/intshifting/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <intshifting> (-1: never, 0: only at depth freqofs)	
heuristics/intshifting/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <intshifting>	
heuristics/linesearchdiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/linesearchdiving/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <linesearchdiving> (-1: never, 0: only at depth freqofs)	
heuristics/linesearchdiving/freqofs ( $0 \leq \text{integer}$ )	6

frequency offset for calling primal heuristic <linesearchdiving>	
heuristics/linesearchdiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/linesearchdiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/localbranching/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <localbranching> (-1: never, 0: only at depth freqofs)	
heuristics/localbranching/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <localbranching>	
heuristics/localbranching/neighborhoodsize ( $1 \leq \text{integer}$ )	18
radius (using Manhattan metric) of the incumbent's neighborhood to be searched	
heuristics/localbranching/nodesofs ( $0 \leq \text{integer}$ )	1000
number of nodes added to the contingent of the total nodes	
heuristics/localbranching/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.05
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/mutation/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <mutation> (-1: never, 0: only at depth freqofs)	
heuristics/mutation/freqofs ( $0 \leq \text{integer}$ )	8
frequency offset for calling primal heuristic <mutation>	
heuristics/mutation/minfixingrate ( $10^{-6} \leq \text{real} \leq 0.999999$ )	0.8
percentage of integer variables that have to be fixed	
heuristics/mutation/nodesofs ( $0 \leq \text{integer}$ )	500
number of nodes added to the contingent of the total nodes	
heuristics/mutation/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/nlpdiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/nlpdiving/fixquot ( $0 \leq \text{real} \leq 1$ )	0.2
percentage of fractional variables that should be fixed before the next NLP solve	
heuristics/nlpdiving/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <nlpdiving> (-1: never, 0: only at depth freqofs)	
heuristics/nlpdiving/freqofs ( $0 \leq \text{integer}$ )	3
frequency offset for calling primal heuristic <nlpdiving>	
heuristics/nlpdiving/maxfeasnlps ( $1 \leq \text{integer}$ )	10
maximal number of NLPs with feasible solution to solve during one dive	
heuristics/nlpdiving/maxnlpiterabs ( $0 \leq \text{integer}$ )	200
minimal absolute number of allowed NLP iterations	
heuristics/nlpdiving/maxnlpiterrel ( $0 \leq \text{integer}$ )	10
additional allowed number of NLP iterations relative to successfully found solutions	
heuristics/nlpdiving/minsuccquot ( $0 \leq \text{real} \leq 1$ )	0.1
heuristic will not run if less then this percentage of calls succeeded (0.0: no limit)	
heuristics/nlpdiving/nlpfastfail (boolean)	TRUE
should the NLP solver stop early if it converges slow?	
heuristics/nlpdiving/prefercover (boolean)	TRUE
should variables in a minimal cover be preferred?	
heuristics/nlpdiving/solvesubmip (boolean)	FALSE

should a sub-MIP be solved if all cover variables are fixed?

heuristics/nlpdiving/varselrule (character)	d
which variable selection should be used? ('f'ractionality, 'c'oefficient, 'p'seudocost, 'g'uided, 'd'ouble, 'v'eclen)	
heuristics/objpscostdiving/freq ( $-1 \leq \text{integer}$ )	20
frequency for calling primal heuristic <objpscostdiving> (-1: never, 0: only at depth freqofs)	
heuristics/objpscostdiving/freqofs ( $0 \leq \text{integer}$ )	4
frequency offset for calling primal heuristic <objpscostdiving>	
heuristics/objpscostdiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/objpscostdiving/maxlpiterquot ( $0 \leq \text{real} \leq 1$ )	0.01
maximal fraction of diving LP iterations compared to total iteration number	
heuristics/octane/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <octane> (-1: never, 0: only at depth freqofs)	
heuristics/octane/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <octane>	
heuristics/oneopt/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling primal heuristic <oneopt> (-1: never, 0: only at depth freqofs)	
heuristics/oneopt/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <oneopt>	
heuristics/pscostdiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/pscostdiving/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <pscostdiving> (-1: never, 0: only at depth freqofs)	
heuristics/pscostdiving/freqofs ( $0 \leq \text{integer}$ )	2
frequency offset for calling primal heuristic <pscostdiving>	
heuristics/pscostdiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/pscostdiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/rens/freq ( $-1 \leq \text{integer}$ )	0
frequency for calling primal heuristic <rens> (-1: never, 0: only at depth freqofs)	
heuristics/rens/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <rens>	
heuristics/rens/minfixingrate ( $0 \leq \text{real} \leq 1$ )	0.5
minimum percentage of integer variables that have to be fixable	
heuristics/rens/nodesofs ( $0 \leq \text{integer}$ )	500
number of nodes added to the contingent of the total nodes	
heuristics/rens/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/rens/startsol (character)	1
solution that is used for fixing values ('l'p relaxation, 'n'lp relaxation)	
heuristics/rins/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <rins> (-1: never, 0: only at depth freqofs)	
heuristics/rins/freqofs ( $0 \leq \text{integer}$ )	5
frequency offset for calling primal heuristic <rins>	
heuristics/rins/minfixingrate ( $0 \leq \text{real} \leq 1$ )	0

minimum percentage of integer variables that have to be fixed	
heuristics/rins/nodesofs ( $0 \leq \text{integer}$ )	500
number of nodes added to the contingent of the total nodes	
heuristics/rins/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/rootsoldiving/freq ( $-1 \leq \text{integer}$ )	20
frequency for calling primal heuristic <rootsoldiving> (-1: never, 0: only at depth freqofs)	
heuristics/rootsoldiving/freqofs ( $0 \leq \text{integer}$ )	5
frequency offset for calling primal heuristic <rootsoldiving>	
heuristics/rootsoldiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000
additional number of allowed LP iterations	
heuristics/rootsoldiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.01
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/rounding/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling primal heuristic <rounding> (-1: never, 0: only at depth freqofs)	
heuristics/rounding/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <rounding>	
heuristics/shiftandpropagate/freq ( $-1 \leq \text{integer}$ )	0
frequency for calling primal heuristic <shiftandpropagate> (-1: never, 0: only at depth freqofs)	
heuristics/shiftandpropagate/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <shiftandpropagate>	
heuristics/shifting/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <shifting> (-1: never, 0: only at depth freqofs)	
heuristics/shifting/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <shifting>	
heuristics/simplerounding/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling primal heuristic <simplerounding> (-1: never, 0: only at depth freqofs)	
heuristics/simplerounding/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <simplerounding>	
heuristics/subnlp/forbidfixings (boolean)	TRUE
whether to add constraints that forbid specific fixings that turned out to be infeasible	
heuristics/subnlp/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling primal heuristic <subnlp> (-1: never, 0: only at depth freqofs)	
heuristics/subnlp/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <subnlp>	
heuristics/subnlp/itermin ( $0 \leq \text{integer}$ )	300
contingent of NLP iterations in relation to the number of nodes in SCIP	
heuristics/subnlp/iteroffset ( $0 \leq \text{integer}$ )	500
number of iterations added to the contingent of the total number of iterations	
heuristics/subnlp/iterquotient ( $0 \leq \text{real}$ )	0.1
contingent of NLP iterations in relation to the number of nodes in SCIP	
heuristics/subnlp/nlpiterlimit ( $0 \leq \text{integer}$ )	0
iteration limit of NLP solver; 0 to use solver default	
heuristics/subnlp/nlptimelimit ( $0 \leq \text{real}$ )	0
time limit of NLP solver; 0 to use solver default	
heuristics/subnlp/nlpverblevel ( $0 \leq \text{integer}$ )	0

verbosity level of NLP solver

heuristics/subnlp/runalways (boolean)	FALSE
whether to run NLP heuristic always if starting point available (does not use iteroffset,iterquot,itermin)	
heuristics/trivial/freq ( $-1 \leq \text{integer}$ )	0
frequency for calling primal heuristic <trivial> (-1: never, 0: only at depth freqofs)	
heuristics/trivial/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <trivial>	
heuristics/trysol/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling primal heuristic <trysol> (-1: never, 0: only at depth freqofs)	
heuristics/trysol/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <trysol>	
heuristics/twoopt/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <twoopt> (-1: never, 0: only at depth freqofs)	
heuristics/twoopt/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <twoopt>	
heuristics/undercover/fixingalts (string)	1i
prioritized sequence of fixing values used ('l'p relaxation, 'n'lp relaxation, 'i'ncumbent solution)	
heuristics/undercover/freq ( $-1 \leq \text{integer}$ )	0
frequency for calling primal heuristic <undercover> (-1: never, 0: only at depth freqofs)	
heuristics/undercover/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <undercover>	
heuristics/undercover/nodesofs ( $0 \leq \text{integer}$ )	500
number of nodes added to the contingent of the total nodes	
heuristics/undercover/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/undercover/onlyconvexify (boolean)	FALSE
should we only fix variables in order to obtain a convex problem?	
heuristics/undercover/postnlp (boolean)	TRUE
should the NLP heuristic be called to polish a feasible solution?	
heuristics/vbounds/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <vbounds> (-1: never, 0: only at depth freqofs)	
heuristics/vbounds/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <vbounds>	
heuristics/vbounds/minfixingrate ( $0 \leq \text{real} \leq 1$ )	0.5
minimum percentage of integer variables that have to be fixable	
heuristics/vbounds/nodesofs ( $0 \leq \text{integer}$ )	500
number of nodes added to the contingent of the total nodes	
heuristics/vbounds/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/veclendiving/backtrack (boolean)	TRUE
use one level of backtracking if infeasibility is encountered?	
heuristics/veclendiving/freq ( $-1 \leq \text{integer}$ )	10
frequency for calling primal heuristic <veclendiving> (-1: never, 0: only at depth freqofs)	
heuristics/veclendiving/freqofs ( $0 \leq \text{integer}$ )	4
frequency offset for calling primal heuristic <veclendiving>	
heuristics/veclendiving/maxlpiterofs ( $0 \leq \text{integer}$ )	1000

additional number of allowed LP iterations	
heuristics/veclndiving/maxlpiterquot ( $0 \leq \text{real}$ )	0.05
maximal fraction of diving LP iterations compared to node LP iterations	
heuristics/zeroobj/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling primal heuristic <zeroobj> (-1: never, 0: only at depth freqofs)	
heuristics/zeroobj/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <zeroobj>	
heuristics/zeroobj/nodesofs ( $0 \leq \text{integer}$ )	100
number of nodes added to the contingent of the total nodes	
heuristics/zeroobj/nodesquot ( $0 \leq \text{real} \leq 1$ )	0.1
contingent of sub problem nodes in relation to the number of nodes of the original problem	
heuristics/zirounding/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling primal heuristic <zirounding> (-1: never, 0: only at depth freqofs)	
heuristics/zirounding/freqofs ( $0 \leq \text{integer}$ )	0
frequency offset for calling primal heuristic <zirounding>	

### Heuristics (advanced options)

heuristics/actconsdiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <actconsdiving> (-1: no limit)	
heuristics/actconsdiving/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/actconsdiving/maxdiveavgquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/actconsdiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/actconsdiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ )	0.1
maximal UBQUOT when no solution was found yet (0.0: no limit)	
heuristics/actconsdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/actconsdiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/actconsdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1003700
priority of heuristic <actconsdiving>	
heuristics/cliue/copycuts (boolean)	TRUE
should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/cliue/initseed ( $0 \leq \text{integer}$ )	0
initial random seed value to permute variables	
heuristics/cliue/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <cliue> (-1: no limit)	
heuristics/cliue/maxnodes ( $0 \leq \text{integer}$ )	5000
maximum number of nodes to regard in the subproblem	
heuristics/cliue/maxpropounds ( $-1 \leq \text{integer} \leq 536870911$ )	2
maximum number of propagation rounds during probing (-1 infinity)	
heuristics/cliue/minimprove ( $0 \leq \text{real} \leq 1$ )	0.01
factor by which cliue heuristic should at least improve the incumbent	
heuristics/cliue/minnodes ( $0 \leq \text{integer}$ )	500
minimum number of nodes required to start the subproblem	

heuristics/clique/multiplier ( $0 \leq \text{real}$ )	1.1
value to increase nodenumber to determine the next run	
heuristics/clique/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1000500
priority of heuristic <clique>	
heuristics/coefdiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <coefdiving> (-1: no limit)	
heuristics/coefdiving/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/coefdiving/maxdiveavgquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/coefdiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/coefdiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ )	0.1
maximal UBQUOT when no solution was found yet (0.0: no limit)	
heuristics/coefdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/coefdiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/coefdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1001000
priority of heuristic <coefdiving>	
heuristics/crossover/copycuts (boolean)	TRUE
if uselprows == FALSE, should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/crossover/dontwaitatroot (boolean)	FALSE
should the nwaitingnodes parameter be ignored at the root node?	
heuristics/crossover/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <crossover> (-1: no limit)	
heuristics/crossover/maxnodes ( $0 \leq \text{integer}$ )	5000
maximum number of nodes to regard in the subproblem	
heuristics/crossover/minimprove ( $0 \leq \text{real} \leq 1$ )	0.01
factor by which Crossover should at least improve the incumbent	
heuristics/crossover/minnodes ( $0 \leq \text{integer}$ )	500
minimum number of nodes required to start the subproblem	
heuristics/crossover/nwaitingnodes ( $0 \leq \text{integer}$ )	200
number of nodes without incumbent change that heuristic should wait	
heuristics/crossover/permute (boolean)	FALSE
should the subproblem be permuted to increase diversification?	
heuristics/crossover/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1104000
priority of heuristic <crossover>	
heuristics/crossover/randomization (boolean)	TRUE
should the choice which sols to take be randomized?	
heuristics/crossover/uselprows (boolean)	FALSE
should subproblem be created out of the rows in the LP rows?	
heuristics/dins/copycuts (boolean)	TRUE
if uselprows == FALSE, should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/dins/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <dins> (-1: no limit)	



heuristics/dins/maxnodes ( $0 \leq \text{integer}$ )	5000
maximum number of nodes to regard in the subproblem	
heuristics/dins/minimprove ( $0 \leq \text{real} \leq 1$ )	0.01
factor by which dins should at least improve the incumbent	
heuristics/dins/nwaitingnodes ( $0 \leq \text{integer}$ )	0
number of nodes without incumbent change that heuristic should wait	
heuristics/dins/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1105000
priority of heuristic <dins>	
heuristics/dins/uselprows (boolean)	FALSE
should subproblem be created out of the rows in the LP rows?	
heuristics/feaspump/copycuts (boolean)	TRUE
should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/feaspump/cyclelength ( $1 \leq \text{integer} \leq 100$ )	3
maximum length of cycles to be checked explicitly in each round	
heuristics/feaspump/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <feaspump> (-1: no limit)	
heuristics/feaspump/maxloops ( $-1 \leq \text{integer}$ )	10000
maximal number of pumping loops (-1: no limit)	
heuristics/feaspump/maxsols ( $-1 \leq \text{integer}$ )	10
total number of feasible solutions found up to which heuristic is called (-1: no limit)	
heuristics/feaspump/maxstallloops ( $-1 \leq \text{integer}$ )	10
maximal number of pumping rounds without fractionality improvement (-1: no limit)	
heuristics/feaspump/minflips ( $1 \leq \text{integer}$ )	10
minimum number of random variables to flip, if a 1-cycle is encountered	
heuristics/feaspump/perturbfreq ( $1 \leq \text{integer}$ )	100
number of iterations until a random perturbation is forced	
heuristics/feaspump/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1000000
priority of heuristic <feaspump>	
heuristics/fixandinfer/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <fixandinfer> (-1: no limit)	
heuristics/fixandinfer/minfixings ( $0 \leq \text{integer}$ )	100
minimal number of fixings to apply before dive may be aborted	
heuristics/fixandinfer/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-500000
priority of heuristic <fixandinfer>	
heuristics/fixandinfer/proprounds ( $-1 \leq \text{integer}$ )	0
maximal number of propagation rounds in probing subproblems (-1: no limit, 0: auto)	
heuristics/fracdiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <fracdiving> (-1: no limit)	
heuristics/fracdiving/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/fracdiving/maxdiveavgquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/fracdiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/fracdiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ )	0.1
maximal UBQUOT when no solution was found yet (0.0: no limit)	

heuristics/fracdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/fracdiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/fracdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1003000
priority of heuristic <fracdiving>	
heuristics/guideddiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <guideddiving> (-1: no limit)	
heuristics/guideddiving/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/guideddiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/guideddiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/guideddiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/guideddiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1007000
priority of heuristic <guideddiving>	
heuristics/intdiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <intdiving> (-1: no limit)	
heuristics/intdiving/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/intdiving/maxdiveavgquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/intdiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/intdiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ )	0.1
maximal UBQUOT when no solution was found yet (0.0: no limit)	
heuristics/intdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/intdiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/intdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1003500
priority of heuristic <intdiving>	
heuristics/intshifting/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <intshifting> (-1: no limit)	
heuristics/intshifting/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-10000
priority of heuristic <intshifting>	
heuristics/linesearchdiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <linesearchdiving> (-1: no limit)	
heuristics/linesearchdiving/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/linesearchdiving/maxdiveavgquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/linesearchdiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	

heuristics/linesearchdiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ ) maximal UBQUOT when no solution was found yet (0.0: no limit)	0.1
heuristics/linesearchdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ ) maximal relative depth to start diving	1
heuristics/linesearchdiving/minreldepth ( $0 \leq \text{real} \leq 1$ ) minimal relative depth to start diving	0
heuristics/linesearchdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of heuristic <linesearchdiving>	-1006000
heuristics/localbranching/copycuts (boolean) if uselprows == FALSE, should all active cuts from cutpool be copied to constraints in subproblem?	TRUE
heuristics/localbranching/maxdepth ( $-1 \leq \text{integer}$ ) maximal depth level to call primal heuristic <localbranching> (-1: no limit)	-1
heuristics/localbranching/maxnodes ( $0 \leq \text{integer}$ ) maximum number of nodes to regard in the subproblem	10000
heuristics/localbranching/minimprove ( $0 \leq \text{real} \leq 1$ ) factor by which localbranching should at least improve the incumbent	0.01
heuristics/localbranching/minnodes ( $0 \leq \text{integer}$ ) minimum number of nodes required to start the subproblem	1000
heuristics/localbranching/nwaitingnodes ( $0 \leq \text{integer}$ ) number of nodes without incumbent change that heuristic should wait	200
heuristics/localbranching/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of heuristic <localbranching>	-1102000
heuristics/localbranching/uselprows (boolean) should subproblem be created out of the rows in the LP rows?	FALSE
heuristics/mutation/copycuts (boolean) if uselprows == FALSE, should all active cuts from cutpool be copied to constraints in subproblem?	TRUE
heuristics/mutation/maxdepth ( $-1 \leq \text{integer}$ ) maximal depth level to call primal heuristic <mutation> (-1: no limit)	-1
heuristics/mutation/maxnodes ( $0 \leq \text{integer}$ ) maximum number of nodes to regard in the subproblem	5000
heuristics/mutation/minimprove ( $0 \leq \text{real} \leq 1$ ) factor by which mutation should at least improve the incumbent	0.01
heuristics/mutation/minnodes ( $0 \leq \text{integer}$ ) minimum number of nodes required to start the subproblem	500
heuristics/mutation/nwaitingnodes ( $0 \leq \text{integer}$ ) number of nodes without incumbent change that heuristic should wait	200
heuristics/mutation/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of heuristic <mutation>	-1103000
heuristics/mutation/uselprows (boolean) should subproblem be created out of the rows in the LP rows?	FALSE
heuristics/nlpdiving/lp (boolean) should the LP relaxation be solved before the NLP relaxation?	FALSE
heuristics/nlpdiving/maxdepth ( $-1 \leq \text{integer}$ ) maximal depth level to call primal heuristic <nlpdiving> (-1: no limit)	-1
heuristics/nlpdiving/maxdiveavgquot ( $0 \leq \text{real}$ ) maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	0

heuristics/nlpdiving/maxdiveavquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/nlpdiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/nlpdiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ )	0.1
maximal UBUOT when no solution was found yet (0.0: no limit)	
heuristics/nlpdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/nlpdiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/nlpdiving/nlpstart (character)	s
which point should be used as starting point for the NLP solver? ('n'one, last 'f'easible, from dive's'tart)	
heuristics/nlpdiving/preferlpfracs (boolean)	FALSE
prefer variables that are also fractional in LP solution?	
heuristics/nlpdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1003000
priority of heuristic <nlpdiving>	
heuristics/objpscostdiving/depthfac ( $0 \leq \text{real}$ )	0.5
maximal diving depth: number of binary/integer variables times depthfac	
heuristics/objpscostdiving/depthfacnosol ( $0 \leq \text{real}$ )	2
maximal diving depth factor if no feasible solution was found yet	
heuristics/objpscostdiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <objpscostdiving> (-1: no limit)	
heuristics/objpscostdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/objpscostdiving/maxsols ( $-1 \leq \text{integer}$ )	-1
total number of feasible solutions found up to which heuristic is called (-1: no limit)	
heuristics/objpscostdiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/objpscostdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1004000
priority of heuristic <objpscostdiving>	
heuristics/octane/ffirst ( $1 \leq \text{integer}$ )	10
number of 0-1-points to be tested at first whether they violate a common row	
heuristics/octane/fmax ( $1 \leq \text{integer}$ )	100
number of 0-1-points to be tested as possible solutions by OCTANE	
heuristics/octane/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <octane> (-1: no limit)	
heuristics/octane/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1008000
priority of heuristic <octane>	
heuristics/octane/useavgnbray (boolean)	TRUE
should the weighted average of the nonbasic cone be used as one ray direction?	
heuristics/octane/useavgray (boolean)	TRUE
should the average of the basic cone be used as one ray direction?	
heuristics/octane/useavgwgray (boolean)	TRUE
should the weighted average of the basic cone be used as one ray direction?	
heuristics/octane/usediffray (boolean)	FALSE
should the difference between the root solution and the current LP solution be used as one ray direction?	

heuristics/octane/usefracspace (boolean)	TRUE
execute OCTANE only in the space of fractional variables (TRUE) or in the full space?	
heuristics/octane/useobjray (boolean)	TRUE
should the inner normal of the objective be used as one ray direction?	
heuristics/oneopt/beforepresol (boolean)	FALSE
should the heuristic be called before presolving?	
heuristics/oneopt/duringroot (boolean)	TRUE
should the heuristic be called before and during the root node?	
heuristics/oneopt/forcelpconstruction (boolean)	FALSE
should the construction of the LP be forced even if LP solving is deactivated?	
heuristics/oneopt/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <oneopt> (-1: no limit)	
heuristics/oneopt/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-20000
priority of heuristic <oneopt>	
heuristics/oneopt/weightedobj (boolean)	TRUE
should the objective be weighted with the potential shifting value when sorting the shifting candidates?	
heuristics/pscostdiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <pscostdiving> (-1: no limit)	
heuristics/pscostdiving/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/pscostdiving/maxdiveavgquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/pscostdiving/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/pscostdiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ )	0.1
maximal UBQUOT when no solution was found yet (0.0: no limit)	
heuristics/pscostdiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/pscostdiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/pscostdiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1002000
priority of heuristic <pscostdiving>	
heuristics/rens/addallsols (boolean)	FALSE
should all subproblem solutions be added to the original SCIP?	
heuristics/rens/binarybounds (boolean)	TRUE
should general integers get binary bounds [floor(.),ceil(.)] ?	
heuristics/rens/copycuts (boolean)	TRUE
if useprows == FALSE, should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/rens/extratime (boolean)	FALSE
should the RENS sub-CIP get its own full time limit? This is only for tesing and not recommended!	
heuristics/rens/fullscale (boolean)	FALSE
should the RENS sub-CIP be solved with cuts, conflicts, strong branching,... This is only for tesing and not recommended!	
heuristics/rens/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <rens> (-1: no limit)	
heuristics/rens/maxnodes ( $0 \leq \text{integer}$ )	5000
maximum number of nodes to regard in the subproblem	

heuristics/rens/minimprove ( $0 \leq \text{real} \leq 1$ )	0.01
factor by which RENS should at least improve the incumbent	
heuristics/rens/minnodes ( $0 \leq \text{integer}$ )	500
minimum number of nodes required to start the subproblem	
heuristics/rens/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1100000
priority of heuristic <rens>	
heuristics/rens/uselprows (boolean)	FALSE
should subproblem be created out of the rows in the LP rows?	
heuristics/rins/copycuts (boolean)	TRUE
if uselprows == FALSE, should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/rins/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <rins> (-1: no limit)	
heuristics/rins/maxnodes ( $0 \leq \text{integer}$ )	5000
maximum number of nodes to regard in the subproblem	
heuristics/rins/minimprove ( $0 \leq \text{real} \leq 1$ )	0.01
factor by which rins should at least improve the incumbent	
heuristics/rins/minnodes ( $0 \leq \text{integer}$ )	500
minimum number of nodes required to start the subproblem	
heuristics/rins/nwaitingnodes ( $0 \leq \text{integer}$ )	200
number of nodes without incumbent change that heuristic should wait	
heuristics/rins/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1101000
priority of heuristic <rins>	
heuristics/rins/uselprows (boolean)	FALSE
should subproblem be created out of the rows in the LP rows?	
heuristics/rootsoldiving/alpha ( $0 \leq \text{real} \leq 1$ )	0.9
soft rounding factor to fade out objective coefficients	
heuristics/rootsoldiving/depthfac ( $0 \leq \text{real}$ )	0.5
maximal diving depth: number of binary/integer variables times depthfac	
heuristics/rootsoldiving/depthfacnosol ( $0 \leq \text{real}$ )	2
maximal diving depth factor if no feasible solution was found yet	
heuristics/rootsoldiving/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <rootsoldiving> (-1: no limit)	
heuristics/rootsoldiving/maxreldepth ( $0 \leq \text{real} \leq 1$ )	1
maximal relative depth to start diving	
heuristics/rootsoldiving/maxsols ( $-1 \leq \text{integer}$ )	-1
total number of feasible solutions found up to which heuristic is called (-1: no limit)	
heuristics/rootsoldiving/minreldepth ( $0 \leq \text{real} \leq 1$ )	0
minimal relative depth to start diving	
heuristics/rootsoldiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1005000
priority of heuristic <rootsoldiving>	
heuristics/rounding/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <rounding> (-1: no limit)	
heuristics/rounding/oncepernode (boolean)	FALSE
should the heuristic only be called once per node?	
heuristics/rounding/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1000
priority of heuristic <rounding>	

heuristics/rounding/successfactor ( $-1 \leq \text{integer}$ )	100
number of calls per found solution that are considered as standard success, a higher factor causes the heuristic to be called more often	
heuristics/shiftandpropagate/cutoffbreaker ( $-1 \leq \text{integer} \leq 1000000$ )	15
The number of cutoffs before heuristic stops	
heuristics/shiftandpropagate/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <shiftandpropagate> (-1: no limit)	
heuristics/shiftandpropagate/npropounds ( $-1 \leq \text{integer} \leq 1000$ )	10
The number of propagation rounds used for each propagation	
heuristics/shiftandpropagate/onlywithoutsol (boolean)	TRUE
Should heuristic only be executed if no primal solution was found, yet?	
heuristics/shiftandpropagate/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	1000
priority of heuristic <shiftandpropagate>	
heuristics/shiftandpropagate/probing (boolean)	TRUE
Should domains be reduced by probing?	
heuristics/shiftandpropagate/relax (boolean)	TRUE
Should continuous variables be relaxed?	
heuristics/shiftandpropagate/sortkey (character)	u
the key for variable sorting: (n)orms or (r)andom	
heuristics/shiftandpropagate/sortvars (boolean)	TRUE
Should variables be sorted for the heuristic?	
heuristics/shifting/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <shifting> (-1: no limit)	
heuristics/shifting/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-5000
priority of heuristic <shifting>	
heuristics/simplerounding/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <simplerounding> (-1: no limit)	
heuristics/simplerounding/oncepernode (boolean)	FALSE
should the heuristic only be called once per node?	
heuristics/simplerounding/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	0
priority of heuristic <simplerounding>	
heuristics/subnlp/keepcopy (boolean)	TRUE
whether to keep SCIP copy or to create new copy each time heuristic is applied	
heuristics/subnlp/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <subnlp> (-1: no limit)	
heuristics/subnlp/maxpresolverounds ( $-1 \leq \text{integer}$ )	-1
limit on number of presolve rounds in sub-SCIP (-1 for unlimited, 0 for no presolve)	
heuristics/subnlp/minimprove ( $0 \leq \text{real} \leq 1$ )	0.01
factor by which NLP heuristic should at least improve the incumbent	
heuristics/subnlp/nlpoptfile (string)	
name of an NLP solver specific options file	
heuristics/subnlp/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-2000000
priority of heuristic <subnlp>	
heuristics/subnlp/resolvefromscratch (boolean)	TRUE
should the NLP resolve be started from the original starting point or the infeasible solution?	
heuristics/subnlp/resolvetolfactor ( $0 \leq \text{real} \leq 1$ )	0.001
if SCIP does not accept a NLP feasible solution, resolve NLP with feas. tolerance reduced by this factor (set to 1.0 to turn	

off resolve)	
heuristics/trivial/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <trivial> (-1: no limit)	
heuristics/trivial/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	10000
priority of heuristic <trivial>	
heuristics/trysol/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <trysol> (-1: no limit)	
heuristics/trysol/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-3000000
priority of heuristic <trysol>	
heuristics/twoopt/intopt (boolean)	FALSE
Should Integer-2-Optimization be applied or not?	
heuristics/twoopt/matchingrate ( $0 \leq \text{real} \leq 1$ )	0.5
parameter to determine the percentage of rows two variables have to share before they are considered equal	
heuristics/twoopt/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <twoopt> (-1: no limit)	
heuristics/twoopt/maxnslaves ( $-1 \leq \text{integer} \leq 1000000$ )	199
maximum number of slaves for one master variable	
heuristics/twoopt/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-20100
priority of heuristic <twoopt>	
heuristics/twoopt/waitingnodes ( $0 \leq \text{integer} \leq 10000$ )	0
user parameter to determine number of nodes to wait after last best solution before calling heuristic	
heuristics/undercover/beforecuts (boolean)	TRUE
should the heuristic be called at root node before cut separation?	
heuristics/undercover/conflictweight (real)	1000
weight for conflict score in fixing order	
heuristics/undercover/copycuts (boolean)	TRUE
should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/undercover/coverbd (boolean)	FALSE
should bounddisjunction constraints be covered (or just copied)?	
heuristics/undercover/coveringobj (character)	u
objective function of the covering problem (influenced nonlinear 'c'onstraints/'t'erns, 'd'omain size, 'l'ocks, 'm'in of up/down locks, 'u'nit penalties)	
heuristics/undercover/cutoffweight ( $0 \leq \text{real}$ )	1
weight for cutoff score in fixing order	
heuristics/undercover/fixingorder (character)	v
order in which variables should be fixed (increasing 'C'onflict score, decreasing 'c'onflict score, increasing 'V'ariable index, decreasing 'v'ariable index)	
heuristics/undercover/fixintfirst (boolean)	FALSE
should integer variables in the cover be fixed first?	
heuristics/undercover/inferenceweight (real)	1
weight for inference score in fixing order	
heuristics/undercover/locksrounding (boolean)	TRUE
shall LP values for integer vars be rounded according to locks?	
heuristics/undercover/maxbacktracks ( $0 \leq \text{integer}$ )	6
maximum number of backtracks in fix-and-propagate	
heuristics/undercover/maxcoversizeconss ( $0 \leq \text{real}$ )	$\infty$
maximum coversize maximum coversize (as ratio to the percentage of non-affected constraints)	



heuristics/undercover/maxcoversizevars ( $0 \leq \text{real} \leq 1$ )	1
maximum coversize (as fraction of total number of variables)	
heuristics/undercover/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <undercover> (-1: no limit)	
heuristics/undercover/maxnodes ( $0 \leq \text{integer}$ )	500
maximum number of nodes to regard in the subproblem	
heuristics/undercover/maxrecovers ( $0 \leq \text{integer}$ )	0
maximum number of recoverings	
heuristics/undercover/maxreorders ( $0 \leq \text{integer}$ )	1
maximum number of reorderings of the fixing order	
heuristics/undercover/mincoveredabs ( $0 \leq \text{integer}$ )	5
minimum number of nonlinear constraints in the original problem	
heuristics/undercover/mincoveredrel ( $0 \leq \text{real} \leq 1$ )	0.15
minimum percentage of nonlinear constraints in the original problem	
heuristics/undercover/minimprove ( $-1 \leq \text{real} \leq 1$ )	0
factor by which the heuristic should at least improve the incumbent	
heuristics/undercover/minnodes ( $0 \leq \text{integer}$ )	500
minimum number of nodes required to start the subproblem	
heuristics/undercover/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1110000
priority of heuristic <undercover>	
heuristics/undercover/recoverdiv ( $0 \leq \text{real} \leq 1$ )	0.9
fraction of covering variables in the last cover which need to change their value when recovering	
heuristics/undercover/reusecover (boolean)	FALSE
shall the cover be reused if a conflict was added after an infeasible subproblem?	
heuristics/vbounds/copycuts (boolean)	TRUE
should all active cuts from cutpool be copied to constraints in subproblem?	
heuristics/vbounds/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <vbounds> (-1: no limit)	
heuristics/vbounds/maxnodes ( $0 \leq \text{integer}$ )	5000
maximum number of nodes to regard in the subproblem	
heuristics/vbounds/maxpropounds ( $-1 \leq \text{integer} \leq 536870911$ )	2
maximum number of propagation rounds during probing (-1 infinity)	
heuristics/vbounds/minimprove ( $0 \leq \text{real} \leq 1$ )	0.01
factor by which vbounds heuristic should at least improve the incumbent	
heuristics/vbounds/minnodes ( $0 \leq \text{integer}$ )	500
minimum number of nodes required to start the subproblem	
heuristics/vbounds/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1106000
priority of heuristic <vbounds>	
heuristics/veclending/maxdepth ( $-1 \leq \text{integer}$ )	-1
maximal depth level to call primal heuristic <veclending> (-1: no limit)	
heuristics/veclending/maxdiveavgquot ( $0 \leq \text{real}$ )	0
maximal quotient (curlowerbound - lowerbound)/(avglowerbound - lowerbound) where diving is performed (0.0: no limit)	
heuristics/veclending/maxdiveavgquotnosol ( $0 \leq \text{real}$ )	0
maximal AVGQUOT when no solution was found yet (0.0: no limit)	
heuristics/veclending/maxdiveubquot ( $0 \leq \text{real} \leq 1$ )	0.8
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where diving is performed (0.0: no limit)	

heuristics/veclendingiving/maxdiveubquotnosol ( $0 \leq \text{real} \leq 1$ ) maximal UBQUOT when no solution was found yet (0.0: no limit)	0.1
heuristics/veclendingiving/maxreldepth ( $0 \leq \text{real} \leq 1$ ) maximal relative depth to start diving	1
heuristics/veclendingiving/minreldepth ( $0 \leq \text{real} \leq 1$ ) minimal relative depth to start diving	0
heuristics/veclendingiving/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of heuristic <veclendingiving>	-1003100
heuristics/zeroobj/addallsols (boolean) should all subproblem solutions be added to the original SCIP?	FALSE
heuristics/zeroobj/maxdepth ( $-1 \leq \text{integer}$ ) maximal depth level to call primal heuristic <zeroobj> (-1: no limit)	0
heuristics/zeroobj/maxlpiters ( $-1 \leq \text{integer}$ ) maximum number of LP iterations to be performed in the subproblem	5000
heuristics/zeroobj/maxnodes ( $0 \leq \text{integer}$ ) maximum number of nodes to regard in the subproblem	1000
heuristics/zeroobj/minimprove ( $0 \leq \text{real} \leq 1$ ) factor by which zeroobj should at least improve the incumbent	0.01
heuristics/zeroobj/minnodes ( $0 \leq \text{integer}$ ) minimum number of nodes required to start the subproblem	100
heuristics/zeroobj/onlywithoutsol (boolean) should heuristic only be executed if no primal solution was found, yet?	TRUE
heuristics/zeroobj/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of heuristic <zeroobj>	100
heuristics/zirounding/maxdepth ( $-1 \leq \text{integer}$ ) maximal depth level to call primal heuristic <zirounding> (-1: no limit)	-1
heuristics/zirounding/maxroundingloops ( $-1 \leq \text{integer}$ ) determines maximum number of rounding loops	2
heuristics/zirounding/minstopncalls ( $1 \leq \text{integer}$ ) determines the minimum number of calls before percentage-based deactivation of Zirounding is applied	1000
heuristics/zirounding/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of heuristic <zirounding>	-500
heuristics/zirounding/stoppercentage ( $0 \leq \text{real} \leq 1$ ) if percentage of found solutions falls below this parameter, Zirounding will be deactivated	0.02
heuristics/zirounding/stopziround (boolean) flag to determine if Zirounding is deactivated after a certain percentage of unsuccessful calls	TRUE

### Limits

limits/absgap ( $0 \leq \text{real}$ ) solving stops, if the absolute gap = $ \text{primalbound} - \text{dualbound} $ is below the given value	GAMS optca
limits/bestsol ( $-1 \leq \text{integer}$ ) solving stops, if the given number of solution improvements were found (-1: no limit)	-1
limits/gap ( $0 \leq \text{real}$ ) solving stops, if the relative gap = $ \text{primal} - \text{dual}  / \text{MIN}( \text{dual} ,  \text{primal} )$ is below the given value	GAMS optcr
limits/maxorigsol ( $0 \leq \text{integer}$ ) maximal number of solutions candidates to store in the solution storage of the original problem	10
limits/maxsol ( $1 \leq \text{integer}$ )	100

maximal number of solutions to store in the solution storage

limits/memory ( $0 \leq \text{real}$ ) GAMS workspace  
 maximal memory usage in MB; reported memory usage is lower than real memory usage!

limits/nodes ( $-1 \leq \text{integer}$ ) GAMS nodlim, if set, otherwise -1  
 maximal number of nodes to process (-1: no limit)

limits/restarts ( $-1 \leq \text{integer}$ ) -1  
 solving stops, if the given number of restarts was triggered (-1: no limit)

limits/solutions ( $-1 \leq \text{integer}$ ) -1  
 solving stops, if the given number of solutions were found (-1: no limit)

limits/stallnodes ( $-1 \leq \text{integer}$ ) -1  
 solving stops, if the given number of nodes was processed since the last improvement of the primal solution value (-1: no limit)

limits/time ( $0 \leq \text{real}$ ) GAMS reslim  
 maximal time in seconds to run

limits/totalnodes ( $-1 \leq \text{integer}$ ) -1  
 maximal number of total nodes (incl. restarts) to process (-1: no limit)

## LP

lp/initialalgorithm (character) s  
 LP algorithm for solving initial LP relaxations (automatic 's'implex, 'p'rimal simplex, 'd'ual simplex, 'b'arrier, barrier with 'c'rossover)

lp/pricing (character) 1  
 LP pricing strategy ('l'pi default, 'a'uto, 'f'ull pricing, 'p'artial, 's'teepest edge pricing, 'q'uickstart steepest edge pricing, 'd'evex pricing)

lp/resolvealgorithm (character) s  
 LP algorithm for resolving LP relaxations if a starting basis exists (automatic 's'implex, 'p'rimal simplex, 'd'ual simplex, 'b'arrier, barrier with 'c'rossover)

lp/solvedepth ( $-1 \leq \text{integer}$ ) -1  
 maximal depth for solving LP at the nodes (-1: no depth limit)

lp/solvefreq ( $-1 \leq \text{integer}$ ) 1  
 frequency for solving LP at the nodes (-1: never; 0: only root LP)

lp/solver (string) cplex, if licensed, otherwise soplex  
 LP solver to use (clp, cplex, soplex)

## LP (advanced options)

lp/checkfeas (boolean) TRUE  
 should LP solutions be checked, resolving LP when numerical troubles occur?

lp/checkstability (boolean) TRUE  
 should LP solver's return status be checked for stability?

lp/cleanupcols (boolean) FALSE  
 should new non-basic columns be removed after LP solving?

lp/cleanupcolsroot (boolean) FALSE  
 should new non-basic columns be removed after root LP solving?

lp/cleanuprows (boolean) TRUE  
 should new basic rows be removed after LP solving?

lp/cleanuprowsroot (boolean) TRUE  
 should new basic rows be removed after root LP solving?

lp/clearinitialprobinglp (boolean) TRUE

should lp state be cleared at the end of probing mode when lp was initially unsolved, e.g., when called right after presolving?

lp/colagelimit ( $-1 \leq \text{integer}$ ) 10  
maximum age a dynamic column can reach before it is deleted from the LP (-1: don't delete columns due to aging)

lp/fastmip ( $0 \leq \text{integer} \leq 1$ ) 1  
which FASTMIP setting of LP solver should be used? 0: off, 1: low

lp/freesolvalbuffers (boolean) FALSE  
should the buffers for storing LP solution values during diving be freed at end of diving?

lp/iterlim ( $-1 \leq \text{integer}$ ) -1  
iteration limit for each single LP solve (-1: no limit)

lp/lexdualalgo (boolean) FALSE  
should the lexicographic dual algorithm be used?

lp/lexdualbasic (boolean) FALSE  
choose fractional basic variables in lexicographic dual algorithm?

lp/lexdualmaxrounds ( $-1 \leq \text{integer}$ ) 2  
maximum number of rounds in the lexicographic dual algorithm (-1: unbounded)

lp/lexdualrootonly (boolean) TRUE  
should the lexicographic dual algorithm be applied only at the root node

lp/lexdualstalling (boolean) TRUE  
turn on the lex dual algorithm only when stalling?

lp/presolving (boolean) TRUE  
should presolving of LP solver be used?

lp/resolveiterfac ( $-1 \leq \text{real}$ ) -1  
factor of average LP iterations that is used as LP iteration limit for LP resolve (-1: unlimited)

lp/resolveitermin ( $1 \leq \text{integer}$ ) 1000  
minimum number of iterations that are allowed for LP resolve

lp/resolverestore (boolean) FALSE  
should the LP be resolved to restore the state at start of diving (if FALSE we buffer the solution values)?

lp/rootiterlim ( $-1 \leq \text{integer}$ ) -1  
iteration limit for initial root LP solve (-1: no limit)

lp/rowagelimit ( $-1 \leq \text{integer}$ ) 10  
maximum age a dynamic row can reach before it is deleted from the LP (-1: don't delete rows due to aging)

lp/rowrepswitch ( $-1 \leq \text{real}$ ) -1  
simplex algorithm shall use row representation of the basis if number of rows divided by number of columns exceeds this value (-1.0 to disable row representation)

lp/scaling (boolean) TRUE  
should scaling of LP solver be used?

lp/threads ( $0 \leq \text{integer} \leq 64$ ) GAMS threads option  
number of threads used for solving the LP (0: automatic)

## Memory

memory/savefac ( $0 \leq \text{real} \leq 1$ ) 0.8  
fraction of maximal memory usage resulting in switch to memory saving mode

## Memory (advanced options)

memory/arraygrowfac ( $1 \leq \text{real} \leq 10$ ) 1.2  
memory growing factor for dynamically allocated arrays

memory/arraygrowinit ( $0 \leq \text{integer}$ ) 4

initial size of dynamically allocated arrays	
memory/pathgrowfac ( $1 \leq \text{real} \leq 10$ )	2
memory growing factor for path array	
memory/pathgrowinit ( $0 \leq \text{integer}$ )	256
initial size of path array	
memory/treegrowfac ( $1 \leq \text{real} \leq 10$ )	2
memory growing factor for tree array	
memory/treegrowinit ( $0 \leq \text{integer}$ )	65536
initial size of tree array	

### Micellaneous

misc/catchctrlc (boolean)	TRUE
should the CTRL-C interrupt be caught by SCIP?	
misc/estimexternmem (boolean)	TRUE
should the usage of external memory be estimated?	
misc/improvingols (boolean)	FALSE
should only solutions be checked which improve the primal bound	
misc/permutationseed ( $-1 \leq \text{integer}$ )	-1
seed value for permuting the problem after the problem was transformed (-1: no permutation)	
misc/printreason (boolean)	FALSE
should the reason be printed if a given start solution is infeasible	
misc/resetstat (boolean)	TRUE
should the statistics be reseted if the transformed problem is freed (in case of a benders decomposition this parameter should be set to FALSE)	
misc/transferols (boolean)	TRUE
should SCIP try to transfer original solutions to the extended space (after presolving)?	
misc/useconstable (boolean)	TRUE
should a hashtable be used to map from constraint names to constraints?	
misc/usesmalltables (boolean)	FALSE
should smaller hashtables be used? yields better performance for small problems with about 100 variables	
misc/usevarstable (boolean)	TRUE
should a hashtable be used to map from variable names to variables?	

### Node Selection

nodeselection/bfs/stdpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	100000
priority of node selection rule <bfs> in standard mode	
nodeselection/childsel (character)	h
child selection rule ('d'own, 'u'p, 'p'seudo costs, 'i'nference, 'l'p value, 'r'oot LP value difference, 'h'ybrid inference/root LP value difference)	
nodeselection/dfs/stdpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	0
priority of node selection rule <dfs> in standard mode	
nodeselection/estimate/bestnodefreq ( $0 \leq \text{integer}$ )	10
frequency at which the best node instead of the best estimate is selected (0: never)	
nodeselection/estimate/breadthfirstdepth ( $-1 \leq \text{integer}$ )	-1
depth until breadth-first search is applied	
nodeselection/estimate/stdpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	200000
priority of node selection rule <estimate> in standard mode	
nodeselection/hybridestim/bestnodefreq ( $0 \leq \text{integer}$ )	1000

frequency at which the best node instead of the hybrid best estimate / best bound is selected (0: never)	
nodeselection/hybridestim/stdpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	50000
priority of node selection rule <hybridestim> in standard mode	
nodeselection/restartdfs/countonlyleaves (boolean)	TRUE
count only leaf nodes (otherwise all nodes)?	
nodeselection/restartdfs/selectbestfreq ( $0 \leq \text{integer}$ )	100
frequency for selecting the best node instead of the deepest one	
nodeselection/restartdfs/stdpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	10000
priority of node selection rule <restartdfs> in standard mode	

### Node Selection (advanced options)

nodeselection/bfs/maxplungedepth ( $-1 \leq \text{integer}$ )	-1
maximal plunging depth, before new best node is forced to be selected (-1 for dynamic setting)	
nodeselection/bfs/maxplungequot ( $0 \leq \text{real}$ )	0.25
maximal quotient (curlowerbound - lowerbound)/(cutoffbound - lowerbound) where plunging is performed	
nodeselection/bfs/memsavepriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	0
priority of node selection rule <bfs> in memory saving mode	
nodeselection/bfs/minplungedepth ( $-1 \leq \text{integer}$ )	-1
minimal plunging depth, before new best node may be selected (-1 for dynamic setting)	
nodeselection/dfs/memsavepriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	100000
priority of node selection rule <dfs> in memory saving mode	
nodeselection/estimate/maxplungedepth ( $-1 \leq \text{integer}$ )	-1
maximal plunging depth, before new best node is forced to be selected (-1 for dynamic setting)	
nodeselection/estimate/maxplungequot ( $0 \leq \text{real}$ )	0.25
maximal quotient (estimate - lowerbound)/(cutoffbound - lowerbound) where plunging is performed	
nodeselection/estimate/memsavepriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	100
priority of node selection rule <estimate> in memory saving mode	
nodeselection/estimate/minplungedepth ( $-1 \leq \text{integer}$ )	-1
minimal plunging depth, before new best node may be selected (-1 for dynamic setting)	
nodeselection/hybridestim/estimweight ( $0 \leq \text{real} \leq 1$ )	0.1
weight of estimate value in node selection score (0: pure best bound search, 1: pure best estimate search)	
nodeselection/hybridestim/maxplungedepth ( $-1 \leq \text{integer}$ )	-1
maximal plunging depth, before new best node is forced to be selected (-1 for dynamic setting)	
nodeselection/hybridestim/maxplungequot ( $0 \leq \text{real}$ )	0.25
maximal quotient (estimate - lowerbound)/(cutoffbound - lowerbound) where plunging is performed	
nodeselection/hybridestim/memsavepriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	50
priority of node selection rule <hybridestim> in memory saving mode	
nodeselection/hybridestim/minplungedepth ( $-1 \leq \text{integer}$ )	-1
minimal plunging depth, before new best node may be selected (-1 for dynamic setting)	
nodeselection/restartdfs/memsavepriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	50000
priority of node selection rule <restartdfs> in memory saving mode	

### Tolerances

numerics/dualfeastol ( $10^{-17} \leq \text{real} \leq 0.001$ )	$10^{-6}$
feasibility tolerance for reduced costs in LP solution	
numerics/epsilon ( $10^{-20} \leq \text{real} \leq 0.001$ )	$10^{-9}$
absolute values smaller than this are considered zero	
numerics/feastol ( $10^{-17} \leq \text{real} \leq 0.001$ )	$10^{-6}$

feasibility tolerance for constraints

numerics/lpfeastol ( $10^{-17} \leq \text{real} \leq 0.001$ ) 10<sup>-6</sup>  
 primal feasibility tolerance of LP solver

numerics/sumepsilon ( $10^{-17} \leq \text{real} \leq 0.001$ ) 10<sup>-6</sup>  
 absolute values of sums smaller than this are considered zero

### Tolerances (advanced options)

numerics/barrierconvtol ( $10^{-17} \leq \text{real} \leq 0.001$ ) 10<sup>-10</sup>  
 LP convergence tolerance used in barrier algorithm

numerics/boundstreps ( $10^{-17} \leq \text{real}$ ) 0.05  
 minimal relative improve for strengthening bounds

numerics/hugeval ( $0 \leq \text{real}$ ) 10<sup>15</sup>  
 values larger than this are considered huge and should be handled separately (e.g., in activity computation)

numerics/pseudocostdelta ( $0 \leq \text{real}$ ) 0.0001  
 minimal objective distance value to use for branching pseudo cost updates

numerics/pseudocosteps ( $10^{-17} \leq \text{real} \leq 1$ ) 0.1  
 minimal variable distance value to use for branching pseudo cost updates

numerics/recompute fac ( $0 \leq \text{real}$ ) 10<sup>7</sup>  
 minimal decrease factor that causes the recomputation of a value (e.g., pseudo objective) instead of an update

### Presolving

presolving/boundshift/maxrounds ( $-1 \leq \text{integer}$ ) 0  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/components/intfactor ( $0 \leq \text{real}$ ) 1  
 the weight of an integer variable compared to binary variables

presolving/components/maxintvars ( $-1 \leq \text{integer}$ ) 500  
 maximum number of integer (or binary) variables to solve a subproblem directly (-1: unlimited)

presolving/components/maxrounds ( $-1 \leq \text{integer}$ ) -1  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/components/nodelimit ( $-1 \leq \text{integer}$ ) 10000  
 maximum number of nodes to be solved in subproblems

presolving/components/reldecrease ( $0 \leq \text{real} \leq 1$ ) 0.2  
 percentage by which the number of variables has to be decreased after the last component solving to allow running again (1.0: do not run again)

presolving/components/writeproblems (boolean) FALSE  
 should the single components be written as an .lp-file?

presolving/convertinttobin/maxrounds ( $-1 \leq \text{integer}$ ) 0  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/domcol/maxrounds ( $-1 \leq \text{integer}$ ) -1  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/dualfix/maxrounds ( $-1 \leq \text{integer}$ ) -1  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/gateextraction/maxrounds ( $-1 \leq \text{integer}$ ) -1  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/implics/maxrounds ( $-1 \leq \text{integer}$ ) -1  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/inttobinary/maxrounds ( $-1 \leq \text{integer}$ ) -1  
 maximal number of presolving rounds the presolver participates in (-1: no limit)

presolving/maxrestarts ( $-1 \leq \text{integer}$ ) maximal number of restarts (-1: unlimited)	-1
presolving/maxrounds ( $-1 \leq \text{integer}$ ) maximal number of presolving rounds (-1: unlimited, 0: off)	-1
presolving/trivial/maxrounds ( $-1 \leq \text{integer}$ ) maximal number of presolving rounds the presolver participates in (-1: no limit)	-1

### Presolving (advanced options)

presolving/abortfac ( $0 \leq \text{real} \leq 1$ ) abort presolve, if at most this fraction of the problem was changed in last presolve round	0.0001
presolving/boundshift/delay (boolean) should presolver be delayed, if other presolvers found reductions?	FALSE
presolving/boundshift/flipping (boolean) is flipping allowed (multiplying with -1)?	TRUE
presolving/boundshift/integer (boolean) shift only integer ranges?	TRUE
presolving/boundshift/maxshift ( $0 \leq \text{integer}$ ) absolute value of maximum shift	$\infty$
presolving/boundshift/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of presolver <boundshift>	7900000
presolving/components/delay (boolean) should presolver be delayed, if other presolvers found reductions?	TRUE
presolving/components/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of presolver <components>	-9200000
presolving/convertinttobin/delay (boolean) should presolver be delayed, if other presolvers found reductions?	FALSE
presolving/convertinttobin/maxdomainsize ( $0 \leq \text{integer}$ ) absolute value of maximum domain size for converting an integer variable to binaries variables	$\infty$
presolving/convertinttobin/onlypoweroftwo (boolean) should only integer variables with a domain size of $2^p - 1$ be converted(, there we don't need an knapsack-constraint for restricting the sum of the binaries)	FALSE
presolving/convertinttobin/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of presolver <convertinttobin>	6000000
presolving/convertinttobin/samelocksinbothdirections (boolean) should only integer variables with uplocks equals downlocks be converted	FALSE
presolving/domcol/delay (boolean) should presolver be delayed, if other presolvers found reductions?	TRUE
presolving/domcol/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of presolver <domcol>	20000000
presolving/donotaggr (boolean) should aggregation of variables be forbidden?	FALSE
presolving/donotmultaggr (boolean) should multi-aggregation of variables be forbidden?	FALSE
presolving/dualfix/delay (boolean) should presolver be delayed, if other presolvers found reductions?	FALSE
presolving/dualfix/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of presolver <dualfix>	8000000



presolving/gateextraction/delay (boolean)	TRUE
should presolver be delayed, if other presolvers found reductions?	
presolving/gateextraction/onlysetpart (boolean)	FALSE
should we only try to extract set-partitioning constraints and no and-constraints	
presolving/gateextraction/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	1000000
priority of presolver <gateextraction>	
presolving/gateextraction/searchequations (boolean)	TRUE
should we try to extract set-partitioning constraint out of one logicor and one corresponding set-packing constraint	
presolving/gateextraction/sorting ( $-1 \leq \text{integer} \leq 1$ )	1
order logicor constraints to extract big-gates before smaller ones (-1), do not order them (0) or order them to extract smaller gates at first (1)	
presolving/immrestartfac ( $0 \leq \text{real} \leq 1$ )	0.2
fraction of integer variables that were fixed in the root node triggering an immediate restart with preprocessing	
presolving/implics/delay (boolean)	FALSE
should presolver be delayed, if other presolvers found reductions?	
presolving/implics/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-10000
priority of presolver <implics>	
presolving/inttobinary/delay (boolean)	FALSE
should presolver be delayed, if other presolvers found reductions?	
presolving/inttobinary/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	7000000
priority of presolver <inttobinary>	
presolving/restartfac ( $0 \leq \text{real} \leq 1$ )	0.05
fraction of integer variables that were fixed in the root node triggering a restart with preprocessing after root node evaluation	
presolving/restartminred ( $0 \leq \text{real} \leq 1$ )	0.1
minimal fraction of integer variables removed after restart to allow for an additional restart	
presolving/subrestartfac ( $0 \leq \text{real} \leq 1$ )	1
fraction of integer variables that were globally fixed during the solving process triggering a restart with preprocessing	
presolving/trivial/delay (boolean)	FALSE
should presolver be delayed, if other presolvers found reductions?	
presolving/trivial/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	9000000
priority of presolver <trivial>	

### Domain Propagation

propagating/abortoncutoff (boolean)	TRUE
should propagation be aborted immediately? setting this to FALSE could help conflict analysis to produce more conflict constraints	
propagating/genvbounds/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling propagator <genvbounds> (-1: never, 0: only in root node)	
propagating/genvbounds/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the propagator participates in (-1: no limit)	
propagating/maxrounds ( $-1 \leq \text{integer}$ )	100
maximal number of propagation rounds per node (-1: unlimited)	
propagating/maxroundsroot ( $-1 \leq \text{integer}$ )	1000
maximal number of propagation rounds in the root node (-1: unlimited)	
propagating/obbt/dualfeastol ( $0 \leq \text{real}$ )	$10^{-9}$
feasibility tolerance for reduced costs used in obbt; this value is used if SCIP's dual feastol is greater	
propagating/obbt/freq ( $-1 \leq \text{integer}$ )	0
frequency for calling propagator <obbt> (-1: never, 0: only in root node)	

propagating/obbt/itlimitfactor (real)	5
multiple of root node LP iterations used as total LP iteration limit for obbt ( $\leq 0$ : no limit)	
propagating/obbt/maxlookahead ( $-1 \leq \text{integer}$ )	3
maximal number of bounds evaluated without success per group (-1: no limit)	
propagating/obbt/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the propagator participates in (-1: no limit)	
propagating/probing/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling propagator <probing> (-1: never, 0: only in root node)	
propagating/probing/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the propagator participates in (-1: no limit)	
propagating/probing/maxruns ( $-1 \leq \text{integer}$ )	1
maximal number of runs, probing participates in (-1: no limit)	
propagating/pseudoobj/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling propagator <pseudoobj> (-1: never, 0: only in root node)	
propagating/pseudoobj/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the propagator participates in (-1: no limit)	
propagating/redcost/continuous (boolean)	FALSE
should reduced cost fixing be also applied to continuous variables?	
propagating/redcost/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling propagator <redcost> (-1: never, 0: only in root node)	
propagating/redcost/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the propagator participates in (-1: no limit)	
propagating/redcost/useimplics (boolean)	TRUE
should implications be used to strength the reduced cost for binary variables?	
propagating/rootredcost/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling propagator <rootredcost> (-1: never, 0: only in root node)	
propagating/rootredcost/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the propagator participates in (-1: no limit)	
propagating/vbounds/dotoposort (boolean)	TRUE
should the bounds be topologically sorted in advance?	
propagating/vbounds/freq ( $-1 \leq \text{integer}$ )	1
frequency for calling propagator <vbounds> (-1: never, 0: only in root node)	
propagating/vbounds/maxprerounds ( $-1 \leq \text{integer}$ )	-1
maximal number of presolving rounds the propagator participates in (-1: no limit)	
propagating/vbounds/sortcliques (boolean)	FALSE
should cliques be regarded for the topological sort?	
propagating/vbounds/usebdwidening (boolean)	TRUE
should bound widening be used to initialize conflict analysis?	
propagating/vbounds/usecliques (boolean)	FALSE
should cliques be propagated?	
propagating/vbounds/useimplics (boolean)	FALSE
should implications be propagated?	
propagating/vbounds/usevbounds (boolean)	TRUE
should vbounds be propagated?	

#### Domain Propagation (advanced options)

propagating/genvbounds/delay (boolean)	FALSE
--	-------

should propagator be delayed, if other propagators found reductions?	
propagating/genvbounds/global (boolean) apply global propagation?	TRUE
propagating/genvbounds/presoldelay (boolean) should presolving be delayed, if other presolvers found reductions?	FALSE
propagating/genvbounds/presolpriority ( $-536870912 \leq \text{integer} \leq 536870911$ ) presolving priority of propagator <genvbounds>	-2000000
propagating/genvbounds/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of propagator <genvbounds>	3000000
propagating/genvbounds/propinrootnode (boolean) apply genvbounds in root node if no new incumbent was found?	TRUE
propagating/genvbounds/sort (boolean) sort genvbounds and wait for bound change events?	TRUE
propagating/genvbounds/timingmask ( $1 \leq \text{integer} \leq 15$ ) timing when propagator should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	15
propagating/obbt/applyfilterrounds (boolean) try to filter bounds in so-called filter rounds by solving auxiliary LPs?	FALSE
propagating/obbt/creategenvbounds (boolean) should obbt try to provide genvbounds if possible?	TRUE
propagating/obbt/delay (boolean) should propagator be delayed, if other propagators found reductions?	TRUE
propagating/obbt/minfilter ( $1 \leq \text{integer}$ ) minimal number of filtered bounds to apply another filter round	2
propagating/obbt/normalize (boolean) should coefficients in filtering be normalized w.r.t. the domains sizes?	TRUE
propagating/obbt/presoldelay (boolean) should presolving be delayed, if other presolvers found reductions?	FALSE
propagating/obbt/presolpriority ( $-536870912 \leq \text{integer} \leq 536870911$ ) presolving priority of propagator <obbt>	0
propagating/obbt/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of propagator <obbt>	-1000000
propagating/obbt/timingmask ( $1 \leq \text{integer} \leq 15$ ) timing when propagator should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	4
propagating/probing/delay (boolean) should propagator be delayed, if other propagators found reductions?	TRUE
propagating/probing/maxdepth ( $-1 \leq \text{integer}$ ) maximal depth until propagation is executed (-1: no limit)	-1
propagating/probing/maxfixings ( $0 \leq \text{integer}$ ) maximal number of fixings found, until probing is interrupted (0: don't interrupt)	25
propagating/probing/maxsumuseless ( $0 \leq \text{integer}$ ) maximal number of probings without fixings, until probing is aborted (0: don't abort)	0
propagating/probing/maxtotaluseless ( $0 \leq \text{integer}$ ) maximal number of successive probings without fixings, bound changes, and implications, until probing is aborted (0: don't abort)	50
propagating/probing/maxuseless ( $0 \leq \text{integer}$ ) maximal number of successive probings without fixings, until probing is aborted (0: don't abort)	1000

propagating/probing/presolddelay (boolean)	TRUE
should presolving be delayed, if other presolvers found reductions?	
propagating/probing/presolpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-100000
presolving priority of propagator <probing>	
propagating/probing/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-100000
priority of propagator <probing>	
propagating/probing/proprounds ( $-1 \leq \text{integer}$ )	-1
maximal number of propagation rounds in probing subproblems (-1: no limit, 0: auto)	
propagating/probing/timingmask ( $1 \leq \text{integer} \leq 15$ )	4
timing when propagator should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
propagating/pseudoobj/delay (boolean)	FALSE
should propagator be delayed, if other propagators found reductions?	
propagating/pseudoobj/force (boolean)	FALSE
should the propagator be forced even active pricer are present?	
propagating/pseudoobj/maximplvars ( $-1 \leq \text{integer}$ )	50000
maximum number of binary variables the implications are used if turned on (-1: unlimited)?	
propagating/pseudoobj/maxnewvars ( $0 \leq \text{integer}$ )	1000
number of variable added after the propagator is reinitialized?	
propagating/pseudoobj/maxvarsfrac ( $0 \leq \text{real} \leq 1$ )	0.1
maximal fraction of none binary variables with non-zero objective without a bound reduction before aborted	
propagating/pseudoobj/minuseless ( $0 \leq \text{integer}$ )	100
minimal number of successive none binary variable propagator without a bound reduction before aborted	
propagating/pseudoobj/presolddelay (boolean)	TRUE
should presolving be delayed, if other presolvers found reductions?	
propagating/pseudoobj/presolpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	6000000
presolving priority of propagator <pseudoobj>	
propagating/pseudoobj/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	3000000
priority of propagator <pseudoobj>	
propagating/pseudoobj/propcutoffbound (boolean)	TRUE
propagate new cutoff bound directly globally	
propagating/pseudoobj/propfullinroot (boolean)	TRUE
do we want to propagate all none binary variables if we are propagating the root node	
propagating/pseudoobj/propuseimplics (boolean)	TRUE
use implications to strengthen the propagation of binary variable (increasing the objective change)?	
propagating/pseudoobj/respropuseimplics (boolean)	TRUE
use implications to strengthen the resolve propagation of binary variable (increasing the objective change)?	
propagating/pseudoobj/timingmask ( $1 \leq \text{integer} \leq 15$ )	5
timing when propagator should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
propagating/redcost/delay (boolean)	FALSE
should propagator be delayed, if other propagators found reductions?	
propagating/redcost/presolddelay (boolean)	FALSE
should presolving be delayed, if other presolvers found reductions?	
propagating/redcost/presolpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	0
presolving priority of propagator <redcost>	
propagating/redcost/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	1000000
priority of propagator <redcost>	

propagating/redcost/timingmask ( $1 \leq \text{integer} \leq 15$ )	2
timing when propagator should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
propagating/rootredcost/delay (boolean)	FALSE
should propagator be delayed, if other propagators found reductions?	
propagating/rootredcost/presoldelay (boolean)	FALSE
should presolving be delayed, if other presolvers found reductions?	
propagating/rootredcost/presolpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	0
presolving priority of propagator <rootredcost>	
propagating/rootredcost/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	10000000
priority of propagator <rootredcost>	
propagating/rootredcost/timingmask ( $1 \leq \text{integer} \leq 15$ )	5
timing when propagator should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	
propagating/vbounds/delay (boolean)	FALSE
should propagator be delayed, if other propagators found reductions?	
propagating/vbounds/presoldelay (boolean)	FALSE
should presolving be delayed, if other presolvers found reductions?	
propagating/vbounds/presolpriority ( $-536870912 \leq \text{integer} \leq 536870911$ )	0
presolving priority of propagator <vbounds>	
propagating/vbounds/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	3000000
priority of propagator <vbounds>	
propagating/vbounds/timingmask ( $1 \leq \text{integer} \leq 15$ )	5
timing when propagator should be called (1:BEFORELP, 2:DURINGLPLOOP, 4:AFTERLPLOOP, 15:ALWAYS))	

### Separation

separating/cgmip/addviolationcons (boolean)	FALSE
add constraint to subscip that only allows violated cuts (otherwise add obj. limit)?	
separating/cgmip/addviolconshdlr (boolean)	FALSE
add constraint handler to filter out violated cuts?	
separating/cgmip/allowlocal (boolean)	FALSE
Allow to generate local cuts?	
separating/cgmip/cmirownbounds (boolean)	FALSE
tell CMIR-generator which bounds to used in rounding?	
separating/cgmip/conshdlrusenorm (boolean)	TRUE
should the violation constraint handler use the norm of a cut to check for feasibility?	
separating/cgmip/contconvert (boolean)	FALSE
Convert some integral variables to be continuous to reduce the size of the sub-MIP?	
separating/cgmip/contconvfrac ( $0 \leq \text{real} \leq 1$ )	0.1
fraction of integral variables converted to be continuous (if contconvert)	
separating/cgmip/contconvmin ( $-1 \leq \text{integer}$ )	100
minimum number of integral variables before some are converted to be continuous	
separating/cgmip/decisiontree (boolean)	FALSE
Use decision tree to turn separation on/off?	
separating/cgmip/dynamiccuts (boolean)	TRUE
should generated cuts be removed from the LP if they are no longer tight?	
separating/cgmip/earlyterm (boolean)	TRUE
terminate separation if a violated (but possibly sub-optimal) cut has been found?	
separating/cgmip/freq ( $-1 \leq \text{integer}$ )	-1

frequency for calling separator <cg mip> (-1: never, 0: only in root node)	
separating/cgmip/intconvert (boolean) Convert some integral variables attaining fractional values to have integral value?	FALSE
separating/cgmip/intconvfrac ( $0 \leq \text{real} \leq 1$ ) fraction of frac. integral variables converted to have integral value (if intconvert)	0.1
separating/cgmip/intconvmin ( $-1 \leq \text{integer}$ ) minimum number of integral variables before some are converted to have integral value	100
separating/cgmip/maxdepth ( $-1 \leq \text{integer}$ ) maximal depth at which the separator is applied (-1: unlimited)	-1
separating/cgmip/maxodelimit ( $-1 \leq \text{integer}$ ) maximum number of nodes considered for sub-MIP (-1: unlimited)	5000
separating/cgmip/maxrounds ( $-1 \leq \text{integer}$ ) maximal number of cg mip separation rounds per node (-1: unlimited)	5
separating/cgmip/maxroundsroot ( $-1 \leq \text{integer}$ ) maximal number of cg mip separation rounds in the root node (-1: unlimited)	50
separating/cgmip/maxrowage ( $-1 \leq \text{integer}$ ) maximal age of rows to consider if onlyactiverows is false	-1
separating/cgmip/minodelimit ( $-1 \leq \text{integer}$ ) minimum number of nodes considered for sub-MIP (-1: unlimited)	500
separating/cgmip/objlone (boolean) Should the objective of the sub-MIP minimize the l1-norm of the multipliers?	FALSE
separating/cgmip/objweighsize (boolean) Weigh each row by its size?	FALSE
separating/cgmip/onlyactiverows (boolean) Use only active rows to generate cuts?	FALSE
separating/cgmip/onlyintvars (boolean) Generate cuts for problems with only integer variables?	FALSE
separating/cgmip/onlyrankone (boolean) Separate only rank 1 inequalities?	FALSE
separating/cgmip/primalseparation (boolean) only separate cuts that are tight for the best feasible solution?	TRUE
separating/cgmip/skipmultbounds (boolean) Skip the upper bounds on the multipliers in the sub-MIP?	TRUE
separating/cgmip/usecmir (boolean) use CMIR-generator (otherwise add cut directly)?	TRUE
separating/cgmip/usecutpool (boolean) use cutpool to store CG-cuts even if the are not efficient?	TRUE
separating/cgmip/usestrongcg (boolean) use strong CG-function to strengthen cut?	FALSE
separating/clique/freq ( $-1 \leq \text{integer}$ ) frequency for calling separator <clique> (-1: never, 0: only in root node)	0
separating/clique/maxsepacuts ( $-1 \leq \text{integer}$ ) maximal number of clique cuts separated per separation round (-1: no limit)	10
separating/closecuts/freq ( $-1 \leq \text{integer}$ ) frequency for calling separator <closecuts> (-1: never, 0: only in root node)	-1
separating/cmir/dynamiccuts (boolean)	TRUE

should generated cuts be removed from the LP if they are no longer tight?	
<code>separating/cmir/freq</code> ( $-1 \leq \text{integer}$ )	0
frequency for calling separator <code>&lt;cmir&gt;</code> (-1: never, 0: only in root node)	
<code>separating/cmir/maxrounds</code> ( $-1 \leq \text{integer}$ )	3
maximal number of cmir separation rounds per node (-1: unlimited)	
<code>separating/cmir/maxroundsroot</code> ( $-1 \leq \text{integer}$ )	10
maximal number of cmir separation rounds in the root node (-1: unlimited)	
<code>separating/cmir/maxsepacuts</code> ( $0 \leq \text{integer}$ )	100
maximal number of cmir cuts separated per separation round	
<code>separating/cmir/maxsepacutsroot</code> ( $0 \leq \text{integer}$ )	500
maximal number of cmir cuts separated per separation round in the root node	
<code>separating/flowcover/dynamiccuts</code> (boolean)	TRUE
should generated cuts be removed from the LP if they are no longer tight?	
<code>separating/flowcover/freq</code> ( $-1 \leq \text{integer}$ )	0
frequency for calling separator <code>&lt;flowcover&gt;</code> (-1: never, 0: only in root node)	
<code>separating/flowcover/maxrounds</code> ( $-1 \leq \text{integer}$ )	5
maximal number of separation rounds per node (-1: unlimited)	
<code>separating/flowcover/maxroundsroot</code> ( $-1 \leq \text{integer}$ )	15
maximal number of separation rounds in the root node (-1: unlimited)	
<code>separating/flowcover/maxsepacuts</code> ( $0 \leq \text{integer}$ )	100
maximal number of flow cover cuts separated per separation round	
<code>separating/flowcover/maxsepacutsroot</code> ( $0 \leq \text{integer}$ )	200
maximal number of flow cover cuts separated per separation round in the root	
<code>separating/gomory/away</code> ( $0 \leq \text{real} \leq 0.5$ )	0.01
minimal integrality violation of a basis variable in order to try Gomory cut	
<code>separating/gomory/dynamiccuts</code> (boolean)	TRUE
should generated cuts be removed from the LP if they are no longer tight?	
<code>separating/gomory/freq</code> ( $-1 \leq \text{integer}$ )	0
frequency for calling separator <code>&lt;gomory&gt;</code> (-1: never, 0: only in root node)	
<code>separating/gomory/maxrounds</code> ( $-1 \leq \text{integer}$ )	5
maximal number of gomory separation rounds per node (-1: unlimited)	
<code>separating/gomory/maxroundsroot</code> ( $-1 \leq \text{integer}$ )	10
maximal number of gomory separation rounds in the root node (-1: unlimited)	
<code>separating/gomory/maxsepacuts</code> ( $0 \leq \text{integer}$ )	50
maximal number of gomory cuts separated per separation round	
<code>separating/gomory/maxsepacutsroot</code> ( $0 \leq \text{integer}$ )	200
maximal number of gomory cuts separated per separation round in the root node	
<code>separating/IMPLIEDBOUNDS/freq</code> ( $-1 \leq \text{integer}$ )	0
frequency for calling separator <code>&lt;IMPLIEDBOUNDS&gt;</code> (-1: never, 0: only in root node)	
<code>separating/intobj/freq</code> ( $-1 \leq \text{integer}$ )	-1
frequency for calling separator <code>&lt;intobj&gt;</code> (-1: never, 0: only in root node)	
<code>separating/maxbounddist</code> ( $0 \leq \text{real} \leq 1$ )	1
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separation (0.0: only on current best node, 1.0: on all nodes)	
<code>separating/maxcuts</code> ( $0 \leq \text{integer}$ )	100
maximal number of cuts separated per separation round (0: disable local separation)	

separating/maxcutsroot ( $0 \leq \text{integer}$ )	2000
maximal number of separated cuts at the root node (0: disable root node separation)	
separating/maxrounds ( $-1 \leq \text{integer}$ )	5
maximal number of separation rounds per node (-1: unlimited)	
separating/maxroundsroot ( $-1 \leq \text{integer}$ )	-1
maximal number of separation rounds in the root node (-1: unlimited)	
separating/maxstallrounds ( $-1 \leq \text{integer}$ )	5
maximal number of consecutive separation rounds without objective or integrality improvement (-1: no additional restriction)	
separating/mcf/dynamiccuts (boolean)	TRUE
should generated cuts be removed from the LP if they are no longer tight?	
separating/mcf/freq ( $-1 \leq \text{integer}$ )	0
frequency for calling separator <mcf> (-1: never, 0: only in root node)	
separating/mcf/maxsepacuts ( $-1 \leq \text{integer}$ )	100
maximal number of mcf cuts separated per separation round	
separating/mcf/maxsepacutsroot ( $-1 \leq \text{integer}$ )	200
maximal number of mcf cuts separated per separation round in the root node – default separation	
separating/minefficacy ( $0 \leq \text{real}$ )	0.05
minimal efficacy for a cut to enter the LP	
separating/minefficacyroot ( $0 \leq \text{real}$ )	0.01
minimal efficacy for a cut to enter the LP in the root node	
separating/minortho ( $0 \leq \text{real} \leq 1$ )	0.5
minimal orthogonality for a cut to enter the LP	
separating/minorthoroot ( $0 \leq \text{real} \leq 1$ )	0.5
minimal orthogonality for a cut to enter the LP in the root node	
separating/oddcycle/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling separator <oddcycle> (-1: never, 0: only in root node)	
separating/oddcycle/liftoddcycles (boolean)	FALSE
should odd cycle cuts be lifted?	
separating/oddcycle/maxrounds ( $-1 \leq \text{integer}$ )	10
maximal number of oddcycle separation rounds per node (-1: unlimited)	
separating/oddcycle/maxroundsroot ( $-1 \leq \text{integer}$ )	10
maximal number of oddcycle separation rounds in the root node (-1: unlimited)	
separating/oddcycle/maxsepacuts ( $0 \leq \text{integer}$ )	5000
maximal number of oddcycle cuts separated per separation round	
separating/oddcycle/maxsepacutsroot ( $0 \leq \text{integer}$ )	5000
maximal number of oddcycle cuts separated per separation round in the root node	
separating/oddcycle/usegls (boolean)	TRUE
should the search method by Groetschel, Lovasz, Schrijver be used? Otherwise use levelgraph method by Hoffman, Padberg.	
separating/poolfreq ( $-1 \leq \text{integer}$ )	0
separation frequency for the global cut pool (-1: disable global cut pool, 0: only separate pool at the root)	
separating/rapidlearning/freq ( $-1 \leq \text{integer}$ )	-1
frequency for calling separator <rapidlearning> (-1: never, 0: only in root node)	
separating/strongcg/dynamiccuts (boolean)	TRUE
should generated cuts be removed from the LP if they are no longer tight?	
separating/strongcg/freq ( $-1 \leq \text{integer}$ )	0



frequency for calling separator <strongcg> (-1: never, 0: only in root node)	
separating/strongcg/maxrounds (-1 ≤ integer)	5
maximal number of strong CG separation rounds per node (-1: unlimited)	
separating/strongcg/maxroundsroot (-1 ≤ integer)	20
maximal number of strong CG separation rounds in the root node (-1: unlimited)	
separating/strongcg/maxsepacuts (0 ≤ integer)	50
maximal number of strong CG cuts separated per separation round	
separating/strongcg/maxsepacutsroot (0 ≤ integer)	500
maximal number of strong CG cuts separated per separation round in the root node	
separating/zerohalf/dynamiccuts (boolean)	TRUE
should generated cuts be removed from the LP if they are no longer tight?	
separating/zerohalf/freq (-1 ≤ integer)	-1
frequency for calling separator <zerohalf> (-1: never, 0: only in root node)	
separating/zerohalf/maxrounds (-1 ≤ integer)	5
maximal number of zerohalf separation rounds per node (-1: unlimited)	
separating/zerohalf/maxroundsroot (-1 ≤ integer)	10
maximal number of zerohalf separation rounds in the root node (-1: unlimited)	
separating/zerohalf/maxsepacuts (0 ≤ integer)	50
maximal number of 0,1/2-cuts separated per separation round	
separating/zerohalf/maxsepacutsroot (0 ≤ integer)	500
maximal number of 0,1/2-cuts separated per separation round in the root node	
separating/zerohalf/preprocessing/decomposeproblem (boolean)	FALSE
should problem be decomposed into subproblems (if possible) before applying preprocessing?	
separating/zerohalf/preprocessing/delta (0 ≤ real ≤ 1)	0.5
value of delta parameter used in preprocessing method 'd'	
separating/zerohalf/preprocessing/ppmethods (string)	CXGXIM
preprocessing methods and ordering:	
# 'd' columns with small LP solution,	
# 'G' modified Gaussian elimination,	
# 'i' identical columns,	
# 'I' identical rows,	
# 'L' large slack rows,	
# 'M' large slack rows (minslack),	
# 's' column singletons,	
# 'X' add trivial zerohalf cuts,	
# 'z' zero columns,	
# 'Z' zero rows,	
# 'C' fast 'z','s',	
# 'R' fast 'Z','L','I'	
## '-' no preprocessing	
#	
separating/zerohalf/separating/auxip/objective (character)	v
auxiliary IP objective:	
# 'v' maximize cut violation,	
# 'u' minimize number of aggregated rows in cut,	
# 'w' minimize number of aggregated rows in cut	
# weighted by the number of rows in the aggregation,	
# 'p' maximize cut violation and penalize a high number	
# of aggregated rows in the cut weighted by the number	
# of rows in the aggregation and the penalty factor p	

#	
separating/zerohalf/separating/auxip/penaltyfactor ( $0 \leq \text{real} \leq 1$ ) penalty factor used with objective function 'p' of auxiliary IP	0.001
separating/zerohalf/separating/auxip/settingsfile (string) optional settings file of the auxiliary IP (-: none)	-
separating/zerohalf/separating/auxip/sollimit ( $-1 \leq \text{integer}$ ) limits/solutions setting of the auxiliary IP	-1
separating/zerohalf/separating/auxip/useallsols (boolean) should all (proper) solutions of the auxiliary IP be used to generate cuts instead of using only the best?	TRUE
separating/zerohalf/separating/forcecutstolp (boolean) should the cuts be forced to enter the LP?	FALSE
separating/zerohalf/separating/forcecutstosepastore (boolean) should the cuts be forced to enter SCIP's sepastore?	FALSE
separating/zerohalf/separating/minviolation ( $0.001 \leq \text{real} \leq 0.5$ ) minimal violation of a 0,1/2-cut to be separated	0.3
separating/zerohalf/separating/sepamethods (string) separating methods and ordering: # '!' stop further processing if a cut was found, # '2' exact polynomial time algorithm (only if matrix has max 2 odd entries per row), # 'e' enumeration heuristics (k=1: try all preprocessed rows), # 'E' enumeration heuristics (k=2: try all combinations of up to two preprocessed rows), # 'g' Extended Gaussian elimination heuristics, # 's' auxiliary IP heuristics (i.e. number of solved nodes is limited) # 'S' auxiliary IP exact (i.e. unlimited number of nodes) # # '-' no processing #	2g

### Separation (advanced options)

separating/cgmip/cutcoefbnd ( $0 \leq \text{real}$ ) bounds on the values of the coefficients in the CG-cut	1000
separating/cgmip/delay (boolean) should separator be delayed, if other separators found cuts?	FALSE
separating/cgmip/maxboundddist ( $0 \leq \text{real} \leq 1$ ) maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <cgmip> (0.0: only on current best node, 1.0: on all nodes)	0
separating/cgmip/memorylimit ( $0 \leq \text{real}$ ) memory limit for sub-MIP	$\infty$
separating/cgmip/objweight ( $0 \leq \text{real}$ ) weight used for the row combination coefficient in the sub-MIP objective	0.001
separating/cgmip/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of separator <cgmip>	-1000
separating/cgmip/timelimit ( $0 \leq \text{real}$ ) time limit for sub-MIP	$\infty$
separating/cliue/backtrackfreq ( $0 \leq \text{integer}$ ) frequency for premature backtracking up to tree level 1 (0: no backtracking)	1000
separating/cliue/cliuedensity ( $0 \leq \text{real} \leq 1$ ) minimal density of cliques to use a dense clique table	0.05
separating/cliue/cliquestablemem ( $0 \leq \text{real} \leq 2.09715 \cdot 10^6$ ) maximal memory size of dense clique table (in kb)	20000

<code>separating/clique/delay</code> (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
<code>separating/clique/maxbounddist</code> ( $0 \leq \text{real} \leq 1$ )	0
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <clique> (0.0: only on current best node, 1.0: on all nodes)	
<code>separating/clique/maxtreenodes</code> ( $-1 \leq \text{integer}$ )	10000
maximal number of nodes in branch and bound tree (-1: no limit)	
<code>separating/clique/maxzeroextensions</code> ( $-1 \leq \text{integer}$ )	1000
maximal number of zero-valued variables extending the clique (-1: no limit)	
<code>separating/clique/priority</code> ( $-536870912 \leq \text{integer} \leq 536870911$ )	-5000
priority of separator <clique>	
<code>separating/clique/scaleval</code> ( $1 \leq \text{real}$ )	1000
factor for scaling weights	
<code>separating/closecuts/closethres</code> ( $-1 \leq \text{integer}$ )	50
threshold on number of generated cuts below which the ordinary separation is started	
<code>separating/closecuts/delay</code> (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
<code>separating/closecuts/inclobjcutoff</code> (boolean)	FALSE
include an objective cutoff when computing the relative interior?	
<code>separating/closecuts/maxbounddist</code> ( $0 \leq \text{real} \leq 1$ )	1
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <closecuts> (0.0: only on current best node, 1.0: on all nodes)	
<code>separating/closecuts/maxlpiterfactor</code> ( $-1 \leq \text{real}$ )	2
factor for maximal LP iterations in relative interior computation compared to node LP iterations (negative for no limit)	
<code>separating/closecuts/maxunsuccessful</code> ( $-1 \leq \text{integer}$ )	0
turn off separation in current node after unsuccessful calls (-1 never turn off)	
<code>separating/closecuts/priority</code> ( $-536870912 \leq \text{integer} \leq 536870911$ )	1000000
priority of separator <closecuts>	
<code>separating/closecuts/recomputerelint</code> (boolean)	FALSE
recompute relative interior point in each separation call?	
<code>separating/closecuts/relintnormtype</code> (character)	o
type of norm to use when computing relative interior: 'o'ne norm, 's'upremum norm	
<code>separating/closecuts/sepacombvalue</code> ( $0 \leq \text{real} \leq 1$ )	0.3
convex combination value for close cuts	
<code>separating/closecuts/separelint</code> (boolean)	TRUE
generate close cuts w.r.t. relative interior point (best solution otherwise)?	
<code>separating/cmirtol/aggrtol</code> ( $0 \leq \text{real}$ )	0.1
tolerance for bound distances used to select continuous variable in current aggregated constraint to be eliminated	
<code>separating/cmirtol/delay</code> (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
<code>separating/cmirtol/densityoffset</code> ( $0 \leq \text{integer}$ )	100
additional number of variables allowed in row on top of density	
<code>separating/cmirtol/densityscore</code> ( $0 \leq \text{real}$ )	0.0001
weight of row density in the aggregation scoring of the rows	
<code>separating/cmirtol/fixintegralrhs</code> (boolean)	TRUE
should an additional variable be complemented if $f_0 = 0$ ?	
<code>separating/cmirtol/maxaggdensity</code> ( $0 \leq \text{real} \leq 1$ )	0.2

maximal density of aggregated row	
separating/cmir/maxaggrs ( $0 \leq \text{integer}$ )	3
maximal number of aggregations for each row per separation round	
separating/cmir/maxaggrsroot ( $0 \leq \text{integer}$ )	6
maximal number of aggregations for each row per separation round in the root node	
separating/cmir/maxbounddist ( $0 \leq \text{real} \leq 1$ )	0
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <cmir> (0.0: only on current best node, 1.0: on all nodes)	
separating/cmir/maxconts ( $0 \leq \text{integer}$ )	10
maximal number of active continuous variables in aggregated row	
separating/cmir/maxcontsroot ( $0 \leq \text{integer}$ )	10
maximal number of active continuous variables in aggregated row in the root node	
separating/cmir/maxfails ( $-1 \leq \text{integer}$ )	20
maximal number of consecutive unsuccessful aggregation tries (-1: unlimited)	
separating/cmir/maxfailsroot ( $-1 \leq \text{integer}$ )	100
maximal number of consecutive unsuccessful aggregation tries in the root node (-1: unlimited)	
separating/cmir/maxrowdensity ( $0 \leq \text{real} \leq 1$ )	0.05
maximal density of row to be used in aggregation	
separating/cmir/maxrowfac ( $0 \leq \text{real}$ )	10000
maximal row aggregation factor	
separating/cmir/maxslack ( $0 \leq \text{real}$ )	0
maximal slack of rows to be used in aggregation	
separating/cmir/maxslackroot ( $0 \leq \text{real}$ )	0.1
maximal slack of rows to be used in aggregation in the root node	
separating/cmir/maxtestdelta ( $-1 \leq \text{integer}$ )	-1
maximal number of different deltas to try (-1: unlimited)	
separating/cmir/maxtries ( $-1 \leq \text{integer}$ )	100
maximal number of rows to start aggregation with per separation round (-1: unlimited)	
separating/cmir/maxtriesroot ( $-1 \leq \text{integer}$ )	-1
maximal number of rows to start aggregation with per separation round in the root node (-1: unlimited)	
separating/cmir/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-3000
priority of separator <cmir>	
separating/cmir/slackscore ( $0 \leq \text{real}$ )	0.001
weight of slack in the aggregation scoring of the rows	
separating/cmir/trynegscaling (boolean)	TRUE
should negative values also be tested in scaling?	
separating/cutagelimit ( $-1 \leq \text{integer}$ )	100
maximum age a cut can reach before it is deleted from the global cut pool, or -1 to keep all cuts	
separating/efficacynorm (character)	e
row norm to use for efficacy calculation ('e':uclidean, 'm':aximum, 's':um, 'd':iscrete)	
separating/flowcover/delay (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
separating/flowcover/maxbounddist ( $0 \leq \text{real} \leq 1$ )	0
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <flowcover> (0.0: only on current best node, 1.0: on all nodes)	
separating/flowcover/maxfails ( $-1 \leq \text{integer}$ )	50
maximal number of consecutive fails to generate a cut per separation round (-1: unlimited)	

separating/flowcover/maxfailsroot ( $-1 \leq \text{integer}$ )	100
maximal number of consecutive fails to generate a cut per separation round in the root (-1: unlimited)	
separating/flowcover/maxrowdensity ( $0 \leq \text{real} \leq 1$ )	1
maximal density of row to separate flow cover cuts for	
separating/flowcover/maxslack ( $0 \leq \text{real}$ )	$\infty$
maximal slack of rows to separate flow cover cuts for	
separating/flowcover/maxslackroot ( $0 \leq \text{real}$ )	$\infty$
maximal slack of rows to separate flow cover cuts for in the root	
separating/flowcover/maxtestdelta ( $0 \leq \text{integer}$ )	10
cut generation heuristic: maximal number of different deltas to try	
separating/flowcover/maxtries ( $-1 \leq \text{integer}$ )	100
maximal number of rows to separate flow cover cuts for per separation round (-1: unlimited)	
separating/flowcover/maxtriesroot ( $-1 \leq \text{integer}$ )	-1
maximal number of rows to separate flow cover cuts for per separation round in the root (-1: unlimited)	
separating/flowcover/multbyminusone (boolean)	TRUE
should flow cover cuts be separated for 0-1 single node flow set with reversed arcs in addition?	
separating/flowcover/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-4000
priority of separator <flowcover>	
separating/flowcover/slackscore ( $0 \leq \text{real}$ )	0.001
weight of slack in the scoring of the rows	
separating/gomory/delay (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
separating/gomory/delayedcuts (boolean)	TRUE
should cuts be added to the delayed cut pool?	
separating/gomory/forcecuts (boolean)	TRUE
if conversion to integral coefficients failed still use the cut	
separating/gomory/makeintegral (boolean)	TRUE
try to scale cuts to integral coefficients	
separating/gomory/maxbounddist ( $0 \leq \text{real} \leq 1$ )	0
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <gomory> (0.0: only on current best node, 1.0: on all nodes)	
separating/gomory/maxweightrange ( $1 \leq \text{real}$ )	10000
maximal valid range $\max( \text{weights} )/\min( \text{weights} )$ of row weights	
separating/gomory/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-1000
priority of separator <gomory>	
separating/gomory/separaterows (boolean)	TRUE
separate rows with integral slack	
separating/impliedbounds/delay (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
separating/impliedbounds/maxbounddist ( $0 \leq \text{real} \leq 1$ )	0
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <impliedbounds> (0.0: only on current best node, 1.0: on all nodes)	
separating/impliedbounds/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-50
priority of separator <impliedbounds>	
separating/intobj/delay (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
separating/intobj/maxbounddist ( $0 \leq \text{real} \leq 1$ )	0

maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <intobj> (0.0: only on current best node, 1.0: on all nodes)	
separating/intobj/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-100
priority of separator <intobj>	
separating/maxaddrounds ( $-1 \leq \text{integer}$ )	1
maximal additional number of separation rounds in subsequent price-and-cut loops (-1: no additional restriction)	
separating/maxroundsrootsubrun ( $-1 \leq \text{integer}$ )	1
maximal number of separation rounds in the root node of a subsequent run (-1: unlimited)	
separating/maxruns ( $-1 \leq \text{integer}$ )	-1
maximal number of runs for which separation is enabled (-1: unlimited)	
separating/mcf/checkcutshoreconnectivity (boolean)	TRUE
should we separate only if the cuts shores are connected?	
separating/mcf/delay (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
separating/mcf/fixintegralrhs (boolean)	TRUE
should an additional variable be complemented if $f_0 = 0$ ?	
separating/mcf/maxarcinconsistencyratio ( $0 \leq \text{real}$ )	0.5
maximum inconsistency ratio of arcs not to be deleted	
separating/mcf/maxbounddist ( $0 \leq \text{real} \leq 1$ )	0
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <mcf> (0.0: only on current best node, 1.0: on all nodes)	
separating/mcf/maxinconsistencyratio ( $0 \leq \text{real}$ )	0.02
maximum inconsistency ratio for separation at all	
separating/mcf/maxtestdelta ( $-1 \leq \text{integer}$ )	20
maximal number of different deltas to try (-1: unlimited) – default separation	
separating/mcf/maxweightrange ( $1 \leq \text{real}$ )	$10^6$
maximal valid range $\max( \text{weights} )/\min( \text{weights} )$ of row weights	
separating/mcf/modeltype ( $0 \leq \text{integer} \leq 2$ )	0
model type of network (0: auto, 1:directed, 2:undirected)	
separating/mcf/nclusters ( $2 \leq \text{integer} \leq 32$ )	5
number of clusters to generate in the shrunken network – default separation	
separating/mcf/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-10000
priority of separator <mcf>	
separating/mcf/separateflowcutset (boolean)	TRUE
should we separate flowcutset inequalities on the network cuts?	
separating/mcf/separateknapsack (boolean)	TRUE
should we separate knapsack cover inequalities on the network cuts?	
separating/mcf/separatesinglenodecuts (boolean)	TRUE
should we separate inequalities based on single-node cuts?	
separating/mcf/trynegscaling (boolean)	FALSE
should negative values also be tested in scaling?	
separating/objparalfac ( $0 \leq \text{real}$ )	0.0001
factor to scale objective parallelism of cut in separation score calculation	
separating/oddcycle/addselfarcs (boolean)	TRUE
add links between a variable and its negated	
separating/oddcycle/allowmultiplecuts (boolean)	TRUE
even if a variable is already covered by a cut, still allow another cut to cover it too	

separating/oddcycle/delay (boolean)	FALSE
should separator be delayed, if other separators found cuts?	
separating/oddcycle/includetriangles (boolean)	TRUE
separate triangles found as 3-cycles or repaired larger cycles	
separating/oddcycle/lpliftcoef (boolean)	FALSE
choose lifting candidate by coef*lpvalue or only by coef	
separating/oddcycle/maxbounddist ( $0 \leq \text{real} \leq 1$ )	1
maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <oddcycle> (0.0: only on current best node, 1.0: on all nodes)	
separating/oddcycle/maxcutslevel ( $0 \leq \text{integer}$ )	50
maximal number of oddcycle cuts generated in every level of the level graph	
separating/oddcycle/maxcutsroot ( $0 \leq \text{integer}$ )	1
maximal number of oddcycle cuts generated per chosen variable as root of the level graph	
separating/oddcycle/maxnlevels ( $0 \leq \text{integer}$ )	20
maximal number of levels in level graph	
separating/oddcycle/maxpernodeslevel ( $0 \leq \text{integer} \leq 100$ )	100
percentage of nodes allowed in the same level of the level graph [0-100]	
separating/oddcycle/maxreference ( $0 \leq \text{integer}$ )	0
minimal weight on an edge (in level graph or bipartite graph)	
separating/oddcycle/maxunsucessfull ( $0 \leq \text{integer}$ )	3
number of unsuccessful calls at current node	
separating/oddcycle/multiplecuts (boolean)	FALSE
even if a variable is already covered by a cut, still try it as start node for a cycle search	
separating/oddcycle/offsetnodeslevel ( $0 \leq \text{integer}$ )	10
offset of nodes allowed in the same level of the level graph (additional to the percentage of levelnodes)	
separating/oddcycle/offsettestvars ( $0 \leq \text{integer}$ )	100
offset of variables to try the chosen method on (additional to the percentage of testvars)	
separating/oddcycle/percenttestvars ( $0 \leq \text{integer} \leq 100$ )	0
percentage of variables to try the chosen method on [0-100]	
separating/oddcycle/priority ( $-536870912 \leq \text{integer} \leq 536870911$ )	-15000
priority of separator <oddcycle>	
separating/oddcycle/recalccliftcoef (boolean)	TRUE
calculate lifting coefficient of every candidate in every step (or only if its chosen)	
separating/oddcycle/repaircycles (boolean)	TRUE
try to repair violated cycles with double appearance of a variable	
separating/oddcycle/scalingfactor ( $1 \leq \text{integer}$ )	1000
factor for scaling of the arc-weights	
separating/oddcycle/sortrootneighbors (boolean)	TRUE
sort level of the root neighbors by fractionality (maxfrac)	
separating/oddcycle/sortswitch ( $0 \leq \text{integer} \leq 4$ )	3
use sorted variable array (unsorted(0),maxlp(1),minlp(2),maxfrac(3),minfrac(4))	
separating/orthofac ( $0 \leq \text{real}$ )	1
factor to scale orthogonality of cut in separation score calculation (0.0 to disable orthogonality calculation)	
separating/orthofunc (character)	e
function used for calc. scalar prod. in orthogonality test ('e' uclidean, 'd' iscrete)	
separating/rapidlearning/applybdchgs (boolean)	TRUE
should the found global bound deductions be applied in the original SCIP?	

separating/rapidlearning/applyconflicts (boolean) should the found conflicts be applied in the original SCIP?	TRUE
separating/rapidlearning/applyinfevals (boolean) should the inference values be used as initialization in the original SCIP?	TRUE
separating/rapidlearning/applyprimalsol (boolean) should the incumbent solution be copied to the original SCIP?	TRUE
separating/rapidlearning/applysolved (boolean) should a solved status be copied to the original SCIP?	TRUE
separating/rapidlearning/contvars (boolean) should rapid learning be applied when there are continuous variables?	FALSE
separating/rapidlearning/contvarsquot ( $0 \leq \text{real} \leq 1$ ) maximal portion of continuous variables to apply rapid learning	0.3
separating/rapidlearning/copycuts (boolean) should all active cuts from cutpool be copied to constraints in subproblem?	TRUE
separating/rapidlearning/delay (boolean) should separator be delayed, if other separators found cuts?	FALSE
separating/rapidlearning/lpiterquot ( $0 \leq \text{real}$ ) maximal fraction of LP iterations compared to node LP iterations	0.2
separating/rapidlearning/maxbounddist ( $0 \leq \text{real} \leq 1$ ) maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <rapidlearning> (0.0: only on current best node, 1.0: on all nodes)	1
separating/rapidlearning/maxnconss ( $0 \leq \text{integer}$ ) maximum problem size (constraints) for which rapid learning will be called	10000
separating/rapidlearning/maxnodes ( $0 \leq \text{integer}$ ) maximum number of nodes considered in rapid learning run	5000
separating/rapidlearning/maxnvars ( $0 \leq \text{integer}$ ) maximum problem size (variables) for which rapid learning will be called	10000
separating/rapidlearning/minnodes ( $0 \leq \text{integer}$ ) minimum number of nodes considered in rapid learning run	500
separating/rapidlearning/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of separator <rapidlearning>	-1200000
separating/rapidlearning/reducedinfer (boolean) should the inference values only be used when rapidlearning found other reductions?	FALSE
separating/strongcg/delay (boolean) should separator be delayed, if other separators found cuts?	FALSE
separating/strongcg/maxbounddist ( $0 \leq \text{real} \leq 1$ ) maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <strongcg> (0.0: only on current best node, 1.0: on all nodes)	0
separating/strongcg/maxweightrange ( $1 \leq \text{real}$ ) maximal valid range $\max( \text{weights} )/\min( \text{weights} )$ of row weights	10000
separating/strongcg/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) priority of separator <strongcg>	-2000
separating/zerohalf/delay (boolean) should separator be delayed, if other separators found cuts?	FALSE
separating/zerohalf/ignoreprevzhcuts (boolean) should zerohalf cuts found in previous callbacks ignored?	FALSE
separating/zerohalf/maxbounddist ( $0 \leq \text{real} \leq 1$ )	0



maximal relative distance from current node's dual bound to primal bound compared to best node's dual bound for applying separator <zerohalf> (0.0: only on current best node, 1.0: on all nodes)

separating/zerohalf/maxcutsfound ( $0 \leq \text{integer}$ ) 100

maximal number of 0,1/2-cuts determined per separation round

# (this includes separated but inefficacious cuts)

separating/zerohalf/maxcutsfoundroot ( $0 \leq \text{integer}$ ) 1000

maximal number of 0,1/2-cuts determined per separation round in the root node

# (this includes separated but inefficacious cuts)

separating/zerohalf/maxdepth ( $-1 \leq \text{integer}$ ) -1

separating cuts only if depth  $\leq$  maxdepth (-1: unlimited)

separating/zerohalf/maxncalls ( $-1 \leq \text{integer}$ ) -1

maximal number of calls (-1: unlimited)

separating/zerohalf/maxtestdelta ( $-1 \leq \text{integer}$ ) 10

maximal number of different deltas to try for cmir (-1: unlimited, 0: delta=1)

separating/zerohalf/onlyorigrows (boolean) FALSE

should only original LP rows be considered (i.e. ignore previously added LP rows)?

separating/zerohalf/priority ( $-536870912 \leq \text{integer} \leq 536870911$ ) -6000

priority of separator <zerohalf>

separating/zerohalf/relaxcontvars (boolean) FALSE

should continuous variables be relaxed by adding variable bounds?

separating/zerohalf/scalefraccoeffs (boolean) TRUE

should rows be scaled to make fractional coefficients integer?

separating/zerohalf/trynegscaling (boolean) TRUE

should negative values also be tested in scaling for cmir?

separating/zerohalf/usezhcutpool (boolean) TRUE

should zerohalf cuts be filtered using a cutpool?

### Timing

timing/clocktype ( $1 \leq \text{integer} \leq 2$ ) 1

default clock type (1: CPU user seconds, 2: wall clock time)

timing/enabled (boolean) TRUE

is timing enabled?

timing/reading (boolean) FALSE

belongs reading time to solving time?

# SCIP References

- [1] Tobias Achterberg. *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2007.
- [2] Tobias Achterberg. SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computations*, 1(1):1–41, 2009.
- [3] Tobias Achterberg, Timo Berthold, Thorsten Koch, and Kati Wolter. Constraint integer programming: A new approach to integrate CP and MIP. In L. Perron and M.A. Trick, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 5th International Conference, CPAIOR 2008*, volume 5015 of *LNCS*, pages 6–20. Springer, 2008.
- [4] Timo Berthold. Primal heuristics for mixed integer programs. Diploma thesis, Technische Universität Berlin, 2006.
- [5] Timo Berthold, Stefan Heinz, and Stefan Vigerske. Extending a CIP framework to solve MIQCPs. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 427–444. Springer, 2012.
- [6] Stefan Vigerske. *Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD thesis, Humboldt Universität zu Berlin, 2012.
- [7] Andreas Wächter and Lorenz T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. <http://projects.coin-or.org/Ipopt>.
- [8] Kati Wolter. Implementation of cutting plane separators for mixed integer programs. Diploma thesis, Technische Universität Berlin, 2006.
- [9] Roland Wunderling. *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin, 1996. <http://www.zib.de/Publications/abstracts/TR-96-09>, <http://soplex.zib.de>.