

# R documentation

of  
'/Users/vaibhavthakkar/Desktop/CS335/randomized\_lp\_solver.rd'

March 18, 2020

---

randomized\_lp\_solver    *Solving a LP problem using Randomized algorithms.*

---

## Description

Currently, only 2 algorithms are implemented - Randomized Cutting plane and Simulated Annealing.

## Usage

```
randomized_lp_solver(P, obj, bounds = NULL, algo = 0L, verbose = FALSE)
```

## Arguments

bounds	optional. A list that contains the bound of the variables (the default is complete Real space), as follows: <ul style="list-style-type: none"><li>• indices A vector containing the variable indices(0 indexed) for which the bounds have to be set.</li><li>• lower A vector containing the value of lower bounds for all the variables specified in indices.</li><li>• upper A vector containing the value of upper bounds for all the variables specified in indices.</li></ul>
algo	Optional. An unsigned integer that declares which algorithm, as follows: <ul style="list-style-type: none"><li>• 0 Use the Randomized Cutting Plane algorithm (RCP).</li><li>• 1 Use the Simulated Annealing algorithm (SIM_ANN).</li></ul>
verbose	Optional. A boolean parameter for printing out the LP program formed.
P.	A convex H Polytope, it is the feasible region of the LP problem ( $Ax \leq b$ ).
obj.	A vector for the coefficients of the objective function ( $min c^T x$ ).

## Value

A list containing the value of the objective function and value of all variables.

## References

*Dabbene, Fabrizio, Pavel S. Shcherbakov, and Boris T. Polyak., “ A randomized cutting plane method with probabilistic geometric convergence,” SIAM Journal on Optimization 20.6, (2010): 3185-3207.,*

*Adam Tauman Kalai, Santosh Vempala, “Simulated Annealing for Convex Optimization,” Mathematics of Operations Research Vol. 31, No. 2, 2006.*

## Examples

```
# computing Chebychev ball for a H-polytope (3d cube)
P <- gen_cube(3, 'H')
row_norm <- sqrt(rowSums((P$A)^2))
P$A <- cbind(P$A, row_norm)
var_bounds <- list("indices"=c(3), "lower"=c(0), "upper"=c(1000))
randomized_lp_solver(P, obj=c(0,0,0,-1), bounds=var_bounds, algo=1, verbose=TRUE)
```

# Index

randomized\_lp\_solver, [1](#)