YALMIP Wiki

Nonlinear Operators

Nonlinear programming

Operators and functions, such as exp or abs, can be equipped with various properties, which allows YALMIP to, e.g., analyze the optimization problem with respect to convexity, select a suitable way to model the problem, or automatically compute gradients and jacobians. If the problem is convex, YALMIP can sometimes use a graph representation, while a nonconvex problem might require the introduction of mixed-integer representations or a callback approach.

Graph-based representations

Graph-based implementations model the operators by using additional variables and constraints, so called epi- and hypo-graphs. The benefit of this modeling strategy is that YALMIP can derive simple optimization models, such as linear programs, instead of treating the operators as general nonlinearities and use a nonlinear solver.

These graph representations are only valid if the expressions satisfy certain convexity and concavity conditions. Hence, YALMIP has to analyze the problem to ensure that the representation actually can be used.

Graph-based implementations are available for a large number of operators, such as min, max, abs, sqrt, norm and geomean. Adding new operators is easy, and can be done almost entirely from template code.

More information can be found here.

Mixed-integer representations

Mixed-integer representations are models that encode an exact representation of an operator by using integer and binary decision variables. The benefit of using such a representation is that the resulting nonconvex problem typically is a well structured problem, such as a mixed-integer linear program. Many of the operators that are implemented using linear programming representable graphs, are also available in a mixed integer representation. If YALMIP fails to propagate convexity, it will switch from graph-based modelling to mixed-integer modelling (unless the option 'allownonconvex' is set to 0). Mixed-integer models are available for, e.g., min, max, abs, and combinatorial and logical operators such as or[?], and[?], min, max, ne[?], iff, implies, nnz, alldifferent, sort and ismember, and on a higher level, piecewise affine and quadratic functions in connection with MPT.

More information can be found here.

Evaluation-based representations

A third way to model operators in YALMIP is by using simple callback evaluations. Operators modelled this way can also be equipped with convexity information, and thus fit into YALMIPs convexity propagations. However, since they are based on callbacks, they can only be used together with general purpose optimization solvers, such as fmincon.

Most of MATLABs built-in nonlinear functions, such as exp and log, are available as evaluation-based representations in YALMIP, and you can easily add your own.

More information can be found here.