# AIMMS Language Reference - Additional Separation Procedures for Benders Decomposition

AIMMS 4

AIMMS B.V.
Diakenhuisweg 29-35
2033 AP Haarlem
The Netherlands
Tel.: +31 23 5511512

AIMMS Inc.
11711 SE 8th Street
Suite 303
Bellevue, WA 98005
USA
Tel.: +1 425 458 4024

AIMMS Pte. Ltd.
55 Market Street #10-00
Singapore 048941
Tel.: +65 6521 2827

AIMMS
SOHO Fuxing Plaza No.388
Building D-71, Level 3
Madang Road, Huangpu District
Shanghai 200025
China
Tel.: ++86 21 5309 8733

Email: info@aimms.com
WWW: www.aimms.com

This documentation was typeset by AIMMS B.V. using LATEX and the LUCIDA font family.

# Chapter 2

# Additional Separation Procedures for Benders' Decomposition

*This chapter*

In Section 21.4 we showed the implementation of the procedure SeparationOptimalityAndFeasibilityDual as used by the textbook algorithm. The Benders' module implements also three other separation procedure that can be used by the Benders' decomposition algorithm depending on the setting of the control parameters UseDual and FeasibilityOnly. In this chapter we explain the implementation of these procedures.

*The procedure Separation-FeasibilityOnly*

The procedure SeparationFeasibilityOnly is called by the Benders' decomposition algorithm in case the primal of the Benders subproblem is used (parameter UseDual equals 0) and if only feasibility cuts can be generated by the algorithm (parameter FeasibilityOnly equals 1). This procedure creates the feasibility problem for the (primal) subproblem if it does not exist yet. The feasibility problem is updated and solved. If its optimal objective value equals 0 (or is negative) then we have found an optimal solution for the original problem and the algorithm terminates. If the optimal objective value is larger than 0, indicating that the subproblem would have been infeasible, we add a feasibility cut to the master problem. The feasibility cut is created using the dual solution of the feasibility problem. By the dual solution we mean the shadow prices of the constraints and the reduced costs of the variables in the feasibility subproblem.

```
return when ( BendersAlgorithmFinished );

! Create feasibility problem corresponding to Subproblem (if it does not exist yet).
if ( not FeasibilityProblemCreated ) then
    gmpF := GMP::Instance::CreateFeasibility( gmpS, "FeasProb",
                              useMinMax : UseMinMaxForFeasibilityProblem );
    solsesF := GMP::Instance::CreateSolverSession( gmpF ) ;
    FeasibilityProblemCreated := 1;
endif;

! Update feasibility problem corresponding to Subproblem and solve it.
GMP::Benders::UpdateSubProblem( gmpF, gmpM, 1, round : 1 );

GMP::SolverSession::Execute( solsesF ) ;
GMP::Solution::RetrieveFromSolverSession( solsesF, 1 ) ;
```

```
! Check whether objective is 0 in which case optimality condition is satisfied.
ObjectiveFeasProblem := GMP::SolverSession::GetObjective( solsesF );

if ( ObjectiveFeasProblem <= BendersOptimalityTolerance ) then
    if ( MasterHasBeenSolved ) then
        return AlgorithmTerminate( 'Optimal' );
    endif;
endif;

! Add feasibility cut to the Master problem.
NumberOfFeasibilityCuts += 1;
GMP::Benders::AddFeasibilityCut( gmpM, gmpF, 1, NumberOfFeasibilityCuts );
```

*The procedure Separation-OptimalityAnd-Feasibility*

The procedure `SeparationOptimalityAndFeasibility` is called by the Benders' decomposition algorithm in case the primal of the Benders subproblem is used (parameter `UseDual` equals 0) and if both optimality and feasibility cuts can be generated by the algorithm (parameter `FeasibilityOnly` equals 0). This procedure updates the primal subproblem and solves it. If the primal subproblem is infeasible then this procedure creates a feasibility problem for the subproblem if it does not exist yet. The feasibility problem is updated and solved, and its dual solution is used to created a feasibility cut which is added to the master problem. If the primal subproblem is bounded and optimal then the objective value of the subproblem is compared to the objective value of the master problem to check whether the algorithm has found an optimal solution for the original problem. If the solution is not optimal yet then an optimality cut is added to the master problem, using the dual solution of the primal subproblem.

```
return when ( BendersAlgorithmFinished );

! Update Subproblem and solve it.
GMP::Benders::UpdateSubProblem( gmpS, gmpM, 1, round : 1 );

GMP::SolverSession::Execute( solsesS ) ;
GMP::Solution::RetrieveFromSolverSession( solsesS, 1 ) ;

ProgramStatus := GMP::Solution::GetProgramStatus( gmpS, 1 ) ;

if ( ProgramStatus = 'Unbounded' ) then
    return AlgorithmTerminate( 'Unbounded' );
endif;

if ( ProgramStatus = 'Infeasible' ) then

    ! Create (if it does not exist yet) and update feasibility problem corresponding to
    ! Subproblem, and solve it to create feasibility cut for the Master problem.
    if ( not FeasibilityProblemCreated ) then
        gmpF := GMP::Instance::CreateFeasibility( gmpS, "FeasProb",
                                  useMinMax : UseMinMaxForFeasibilityProblem );
        solsesF := GMP::Instance::CreateSolverSession( gmpF ) ;
        FeasibilityProblemCreated := 1;
    endif;
```

```
        GMP::Benders::UpdateSubProblem( gmpF, gmpM, 1, round : 1 );

        GMP::SolverSession::Execute( solsesF ) ;
        GMP::Solution::RetrieveFromSolverSession( solsesF, 1 ) ;

        ! Add feasibility cut to the Master problem.
        NumberOfFeasibilityCuts += 1;
        GMP::Benders::AddFeasibilityCut( gmpM, gmpF, 1, NumberOfFeasibilityCuts );

    else

        ! Check whether optimality condition is satisfied.
        ObjectiveSubProblem := GMP::SolverSession::GetObjective( solsesS );

        if ( SolutionImprovement( ObjectiveSubProblem, BestObjective ) ) then
            BestObjective := ObjectiveSubProblem;
        endif;

        if ( SolutionIsOptimal( ObjectiveSubProblem, ObjectiveMaster ) ) then
            return AlgorithmTerminate( 'Optimal' );
        endif;

        ! Add optimality cut to the Master problem.
        NumberOfOptimalityCuts += 1;
        GMP::Benders::AddOptimalityCut( gmpM, gmpS, 1, NumberOfOptimalityCuts );

    endif;
```

*The procedure Separation-Feasibility-OnlyDual*

The procedure SeparationFeasibilityOnlyDual is called by the Benders' decomposition algorithm in case the dual of the Benders subproblem is used (parameter UseDual equals 1) and if only feasibility cuts can be generated by the algorithm (parameter FeasibilityOnly equals 1). This procedure updates the dual subproblem and solves it. If its optimal objective value equals 0 (or is negative) then we have found an optimal solution for the original problem and the algorithm terminates. If the optimal objective value is larger than 0 then we create a feasibility cut using the level values of the variables in the solution of the dual subproblem. This feasibility cut is added to the master problem.

```
    return when ( BendersAlgorithmFinished );

    ! Update Subproblem and solve it.
    GMP::Benders::UpdateSubProblem( gmpS, gmpM, 1, round : 1 );

    GMP::SolverSession::Execute( solsesS ) ;
    GMP::Solution::RetrieveFromSolverSession( solsesS, 1 ) ;

    ! Check whether objective is 0 in which case optimality condition is satisfied.
    ObjectiveSubproblem := GMP::SolverSession::GetObjective( solsesS );

    if ( ObjectiveSubproblem <= BendersOptimalityTolerance ) then
        if ( MasterHasBeenSolved ) then
            return AlgorithmTerminate( 'Optimal' );
        endif;
    else
        ! Add feasibility cut to the Master problem.
        NumberOfFeasibilityCuts += 1;
        GMP::Benders::AddFeasibilityCut( gmpM, gmpS, 1, NumberOfFeasibilityCuts );
    endif;
```