Convex MINLP Algorithms

Algorithm	Basic Idea	Solvers
Branch and Bound	Solve continuous relaxation; branch on fractional x_j	**BONMIN, MINLPBB**, SBB
Extended Cutting Plane	Solve MILP iteratively; if solution is infeasible for MINLP then add a linearization of the most violated constraint to the next MILP. If solution is feasible then it is optimal	Alpha-ECP
Outer-Approximation	Based on OA cut: Divide into 2 subproblems: NLP in which the integer variables are fixed; MILP which uses the OA cuts generated from the NLP	**BONMIN**, DICOPT, MINOPT
Generalised Benders Decompostion	Similar to OA; 2 subproblems: NLP in which the integer variables are fixed; MILP master problem which is the relaxation of the projection of the MINLP on the x-space	MINOPT
LP/NLP B&B	Use BB to solve MILP of NLP relaxation. When an integer feasible solution is found, solve NLP subproblem and add OA cut to all open nodes	**BONMIN, FilMINT**
Bonami Hybrid	Like above, but NLP is solved at more times than just when integer solution is found. Also, local search is performed at some nodes by partial enumeration of MILP relaxations (to collect more OA cuts and improve bounds early)	**BONMIN**

NonConvex Adjustments

Reformulation	
Convex envelopes/Underestimators	
Factorisation	
:: Spatial Branch-and-Bound, branch and reduce	**COUENNE, BARON**

COUENNE Algorithm

Input: Problem P'
Output: The value z_{opt} of an optimal solution of P'

```
Define set L of subproblems; let L ← {P′};
Define z_u as an upper bound for P′; let z_u ← +∞
    while L <> 0
        choose P_k ∈ L
        L ← L \ {P_k}
        apply bounds tightening to P_k
        if bounds tightening did not prove P_k infeasible, then
            generate a linear relaxation LP_k of P_k
            repeat
                solve LP_k; let x'_k be an optimum and z'_k its obj value
                refine linearization LP_k
            until x'_k is feasible for P_k or z'_k does not improve sufficiently
        if x'_k is feasible for P_k, then let z_u ← min{z_u, z'_k}
        (optional) find a local optimum z*_k of P_k
            z_u ← min{z_u, z*_k}
        if z'_k ≤ z_u − ε then
            choose a variable x_i
            choose a branching point x_bi
            create subproblems:
                P_k− with x_i ≤ x_bi
                P_k+ with x_i ≥ x_bi
            L ← L ∪ {P_k−,P_k+}
    output z_opt := z_u
```

Select Sources:

C. D'Ambrosio, A. Lodi. Mixed integer nonlinear programming tools: a practical overview. 4OR-Q J Oper Res (2011) 9:329–349

S. Burer, A. Letchford. Non-convex mixed-integer nonlinear programming: A survey. Surveys in Operations Research and Management Science 17 (2012) 97–106

P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Waechter. Branching and bounds tightening techniques for non-convex MINLP. Optimization Methods and Software (2009), 24(4-5):597–634.