Cut-First Branch-and-Price-Second for the Capacitated Arc-Routing Problem

Stefan Irnich

(joint work with Claudia Bode)

Chair of Logistics Management, Mainz School of Management and Economics

Johannes Gutenberg University, D-55099 Mainz, Germany

Email: irnich@uni-mainz.de

This paper presents the first full-fledged branch-and-price (bap) algorithm for the capacitated arc-routing problem (CARP). Prior exact solution techniques either rely on cutting planes or the transformation of the CARP into a node-routing problem. The drawbacks are either models with inherent symmetry, dense underlying networks, or a formulation where edge flows in a potential solution do not allow the reconstruction of unique CARP tours. The proposed algorithm circumvents all these drawbacks by taking the beneficial ingredients from existing CARP methods and combining them in a new way. The first step is the solution of the one-index formulation of the CARP in order to produce strong cuts and an excellent lower bound. It is known that this bound is typically stronger than relaxations of a pure set-partitioning CARP model. Such a set-partitioning master program results from a Dantzig-Wolfe decomposition. In the second phase, the master program is initialized with the strong cuts, CARP tours are iteratively generated by a pricing procedure, and branching is required to produce integer solutions. This is a cut-first bap-second algorithm and its main function is, in fact, the splitting of edge flows into unique CARP tours.

Computational experiments show that the proposed cut-first branch-and-price-second algorithm gives considerable results in all four standard benchmark sets (`kshs`, `gdb`, `bccm`, and `egl`). Several earlier exact approaches proved optimality of known heuristic solutions by matching lower and upper bounds, but were not able to deliver optimal CARP routes. Our branching scheme however enables us to compute feasible integer solutions and optimal ones in many cases. As a result, all open `bccm` benchmark instances are solved now. For the `egl` benchmark set, optimality of one more instance was shown and many lower bounds were improved significantly.