Speeding up Logic-Based Benders' Decomposition by a Metaheuristic for a Bi-Level Capacitated Vehicle Routing Problem*

Günther R. Raidl[1], Thomas Baumhauer[1,2], and Bin Hu[1]

[1] Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
{raidl|hu}@ads.tuwien.ac.at
[2] Salzburg Research, Salzburg, Austria
thomas.baumhauer@gmail.com

Abstract. Benders' Decomposition (BD) is a prominent technique for tackling large mixed integer programming problems having a certain structure by iteratively solving a series of smaller master and subproblem instances. We apply a generalization of this technique called Logic-Based BD, which does not restrict the subproblems to have continuous variables only, to a bi-level vehicle routing problem originating in the timely distribution of printed newspapers to subscribers. When solving all master and subproblem instances exactly by CPLEX, it turns out that the scalability of the approach is quite limited. The situation can be dramatically improved when using a meaningful metaheuristic – in our case a variable neighborhood search – for approximately solving either only the subproblems or both, the master as well as the subproblem instances. More generally, it is shown that Logic-Based BD can be a highly promising framework also for hybrid metaheuristics.

1 Introduction

Benders' Decomposition (BD) [1] is a classical and frequently applied approach for solving large *Mixed Integer Linear Programming* (MIP) problems having a special block-diagonal structure with "complicating variables". It essentially reformulates a given problem by expressing it as a master problem on only a subset of all original variables – the complicating ones – and considering the contributions of all further variables by additional inequalities, so-called *Benders' cuts*. The optimization starts by solving a restricted form of the master problem without any or with only few of these inequalities. A new Benders' cut is then identified by solving a subproblem and its dual on the remaining variables with the master problem variables fixed to the current master solution. Obtained Benders' cuts are added to the master problem and the process is iterated until no further Benders' cuts can be derived. When the master problem and all subproblems are solved to optimality, the finally obtained solution also is optimal for the original problem.

* This work is supported by Salzburg Research, Austria, and the Austrian Science Fund (FWF) under grant P24660.

A major restriction of this original form of BD is the fact that the subproblem must be a *Linear Programming* (LP) problem with only continuous variables as its dual solution is required to derive the Benders' cuts. Some authors, however, have also generalized BD to other types of subproblems, such as certain kinds of continuous non-linear ones [2]. In particular, Hooker and Ottosson [3] proposed *logic-based BD*, which is applicable to a wide category of subproblems including discrete ones. This is achieved by generalizing the LP dual to an *inference dual*. Constraint programming techniques turned out to be especially useful in conjunction with logic-based BD, and this combination could be successfully applied to several problems, in particular in the planning and scheduling domain [4].

In other works it has been shown that metaheuristics can be very useful in conjunction with classical BD: While the LP subproblems are usually solved efficiently by an LP-solver, the master problem typically remains a MIP, although smaller than the original problem, and in general needs to be resolved with newly added Benders' cuts many times. It has therefore been suggested to solve the master problem only approximately but faster by means of metaheuristics, and possibly only in the end apply an exact method in order to obtain a guaranteed optimum. Poojari and Beasley [5] describe such an approach for solving general MIPs in which a genetic algorithm together with a feasibility pump heuristic are applied to the master problem. The authors argue that a population based metaheuristic like a genetic algorithm is particularly useful as it provides multiple solutions in each iteration giving rise to more Benders' cuts. Similarly in spirit, Lai et al. [6, 7] propose a genetic algorithm/BD hybrid for solving the capacitated plant location problem; results indicate a tremendous saving of computation time in comparison to classical BD. Lai et al. [8] further discuss such an approach for a *Capacitated Vehicle Routing Problem* (VRP). Rei et al. [9] suggest to use local branching for solving a MIP master problem in order to sooner find improved upper as well as lower bounds.

It has also been recognized that BD subproblems need not necessarily always to be solved to optimality in order to obtain useful Benders' cuts, even when completeness of the whole approach shall be retained [10]. Especially when considering difficult subproblems in logic-based BD, this aspect becomes increasingly interesting. However, we are not aware of any work so far where metaheuristics have been applied to discrete BD subproblems for deriving Benders' cuts. The major reason obviously lies in the difficulty that it is not sufficient to find a heuristic solution to the subproblem but dual solution information is also required for identifying Benders' cuts that are guaranteed to be valid for any master problem solution. In fact, suboptimal solutions to the subproblem may easily yield inequalities that cut away too large portions of the search space, possibly also a global optimum.

This work considers a bi-level vehicle routing problem motivated by the time-critical distribution of newspapers from a printing center via satellite depots to subscribers and demonstrates how a metaheuristic may effectively be applied to the master as well as the subproblem instances of a suitable logic-based BD. Experimental results indicate that high-quality solutions can be obtained much faster than when using CPLEX for solving the master and subproblem instances exactly, and the scalability of the BD to large instances is substantially improved.

The next Sections 2 to 4 introduce the considered bi-level vehicle routing problem, refer to related work, and present a basic MIP formulation, respectively. Section 5 describes the applied logic-based BD. All the metaheuristic enhancements are presented in Section 6. Experimental results of the basic MIP, classical logic-based BD where all subproblems are solved to optimality, and metaheuristic hybrid variants are discussed in Section 7. Finally, Section 8 concludes this article with remarks on future work.

2 The Bi-Level Capacitated Vehicle Routing Problem with Time Limits

We consider a two-level vehicle routing problem in which goods shall be transported from a main depot to satellite depots and from there further to customers. Homogeneous vehicle fleets exist at the main depot and each satellite depot. A global time limit is imposed on all deliveries, i.e., each customer has to receive its goods within this time. In contrast to the two-echelon vehicle routing problem known in the literature [11–13], the assignment of customers to the satellite depots is pre-specified in our case.

This problem is motivated by the real-world scenario at Mediaprint, a major Austrian newspaper print shop who has to distribute printed newspapers from each printing center to subscribers within a guaranteed time. A natural assignment of subscribers to satellite depots arises here from the fact that region-specific supplements such as advertisements are added to the newspapers, and each region-specific version is only distributed via a dedicated satellite depot. The real distribution scenario even comprises three levels, but it turns out that only the first two levels, up to certain delivery points we call customers here, can be meaningfully optimized as the lowest level corresponds to routes of delivery agents who do not need a more serious planning or do this on their own.

We define the *Bi-Level Capacitated Vehicle Routing Problem with Time Limits* (2L-VRP-TL) as follows. Given are

- a complete, directed graph $G_0 = (V_0, A_0)$ with node set $V_0 = \{0\} \cup V'_0$ and arc set $A_0 = V_0 \times V_0$, where the special node 0 represents a main depot and V'_0 the set of further satellite depots;
- for each satellite depot $s \in V'_0$ a complete, directed graph $G_s = (V_s, A_s)$ with node set $V_s = \{s\} \cup V'_s$, where V'_s represents a set of customers that receive their deliveries via satellite depot s;
- a demand $q_v \geq 0$ for each customer $v \in V'_s$ and a resulting total demand $q_s = \sum_{v \in V'_s} d_v$ for each satellite depot $s \in V'_0$;
- travel cost $c_{u,v} \geq 0$ and a travel time $t_{u,v} \geq 0$ for each arc $(u, v) \in \bigcup_{s \in V_0} A_s$ representing the fastest way to go from u to v;
- vehicle capacities $Q_s \geq 0$ for each vehicle starting at depot $s \in V_0$; thus, we assume a homogeneous vehicle fleet for each depot and the number of vehicles is not limited; in our practical application, larger vehicles are used for the first level and smaller ones for the second level;
- and a global time limit T (due time) within which all deliveries at customers have to be performed.

A solution R consists of a set of routes R_s in each subgraph $G_s,\ \forall s \in V_0$, with a route $r \in R_s$ being an ordered sequence of nodes $r = (r_i)_{i=1,\ldots,|r|}$ with $r_i \in V_s'$. Each vehicle starts its route at the depot s, visits the nodes as specified by r and finally has to return to the depot again. For convenience, we also define $r_0 = r_{|r|+1} = s$. Each node except the main depot 0 has to be visited exactly once, all satellite depots within the first-level routes R_0 and all customer nodes within the second-level routes $\bigcup_{s \in V_0'} R_s$. Thus, each set of routes R_s also defines a partitioning of V_s'.

The cost $c(r)$ of a route $r \in R_s,\ \forall s \in V_0$, is

$$c(r) = \sum_{i=1}^{|r|+1} c_{r_{i-1},r_i}, \tag{1}$$

the route's total demand is

$$q(r) = \sum_{i=1}^{|r|} q_{r_i}, \tag{2}$$

and the times needed to reach each node r_i from the route's depot s are

$$t(r_i) = \sum_{j=1}^{i} t_{r_{j-1},r_j} \quad \forall i = 1,\ldots,|r|. \tag{3}$$

A solution is feasible if the routes satisfy the capacity constraints

$$q(r) \le Q_s \quad \forall r \in R_s, s \in V_0, \tag{4}$$

and all deliveries are performed within the due time T. Since the second-level tours may only start after the goods have been delivered to the respective satellite depots by the first-level tours, the latter holds when

$$t(s) + t(v) \le T \quad \forall v \in V_s',\ s \in V_0'. \tag{5}$$

The objective is to minimize the total cost of a solution, which is the sum over all its routes' costs

$$c(R) = \sum_{s \in V_0} \sum_{r \in R_s} c(r). \tag{6}$$

3 Related Work

As already mentioned, 2L-VRP-TL is related to the *Two-Echelon Vehicle Routing Problem* (2E-VRP) [11], in which also a two-level distribution via satellite depots is considered. Major differences are, however, that in 2E-VRP no time constraints are considered and the assignments of customers to satellites are not fixed but shall also be optimized. This additional degree of flexibility makes 2E-VRP even harder to solve in practice. Perboli et al. [11] propose a flow-based MIP model, strengthening inequalities, and two matheuristics. Experimental results are shown for instances with up to 50 customers and four satellites.

Crainic et al. [14] describe for the same problem multi-start heuristics based on separating the depot-to-satellite transfer and the satellite-to-customer delivery by iteratively solving the two resulting routing problems. In its spirit, this concept comes close to our logic-based Benders' decomposition, although it is not an exact approach. Hemmelmayr et al. [12] further describe an adaptive large neighborhood search heuristic involving several neighborhood structures exploiting specificities of the 2E-VRP.

Already in 1989, Jacobsen and Madsen [15] addressed the *Two-Echelon Location-Routing Problem* in the context of newspaper delivery, which further generalizes 2E-VRP by the additional aspect of deciding at which locations to open facilities (corresponding to depots). The authors suggest and compare three rather simple construction heuristics. Later more sophisticated approaches include a tabu search [16], diverse MIP models [17], and a variable neighborhood search [13].

Concerning BD and more classical (single-level) VRPs, Fisher and Jaikumar [18] describe an approach where the master problem is a general assignment problem and the subproblem is a traveling salesman problem with time-windows for each vehicle. Lai et al. [8] propose the already mentioned hybrid of BD and a genetic algorithm. Here the VRP is expressed by a multi-commodity flow formulation, the subproblems are network flow problems that can be solved efficiently, and the remaining master problem is approximately solved by the genetic algorithm.

For a more general introduction that presents BD and Lagrangian relaxation from a metaheuristic design perspective see [19].

4 MIP Model for 2L-VRP-TL

The above introduced 2L-VRP-TL can be modeled by the following MIP using variables

- $x_{u,v} \in \{0,1\},\ \forall (u,v) \in A_s,\ s \in V_0$ indicating the arcs used for realizing the routes and
- $t_v \geq 0,\ \forall v \in V_s,\ s \in V_0$ corresponding to the above defined $t(v)$, i.e., the time needed make the delivery at v from starting at the respective depot s.

(2L-VRP-TL)

$$\text{minimize} \quad \sum_{s \in V_0} \sum_{(u,v) \in A_s} c_{u,v}\, x_{u,v} \tag{7}$$

$$\text{s.t.} \quad (x(A_s), t(V_s')) \in \mathrm{VRP}(G_s) \qquad \forall s \in V_0 \tag{8}$$

$$t_s + t_v \leq T \qquad \forall v \in V_s,\ s \in V_0' \tag{9}$$

$$0 \leq t_v \leq T \qquad \forall v \in V_s',\ s \in V_0 \tag{10}$$

$$x_{u,v} \in \{0,1\} \qquad \forall (u,v) \in A_s,\ s \in V_0 \tag{11}$$

In (8) $\mathrm{VRP}(G_s)$ represents a valid formulation for the classical capacitated vehicle routing problem including the calculation of the corresponding traveling times $t(v)$ on graph G_s expressed on the variables $x(A_s)$ and $t_v(V_s)$. Equations (9) limit the total times for the deliveries at all customers to T.

$\mathrm{VRP}(G_s)$, for $s \in V_0$, can be expressed in different ways, for simplicity we use here the following compact Miller-Tucker-Zemlin-based formulation, see e.g. [20], although significantly fore effective (but much more complex) approaches exist. Additionally used variables are

- $g_v \geq 0, v \in V_s$ corresponding to the total demand of the nodes in the tour starting from s up to (and including) v.

$(\mathrm{VRP}(G_s))$

$$\sum_{v \in V_s} x_{u,v} = 1 \qquad \forall u \in V_s' \tag{12}$$

$$\sum_{u \in V_s} x_{u,v} = 1 \qquad \forall v \in V_s' \tag{13}$$

$$\sum_{v \in V_s'} x_{s,v} = \sum_{u \in V_s'} x_{u,s} \tag{14}$$

$$g_v - g_u + Q_s(1 - x_{u,v}) \geq q_v \qquad \forall (u,v) \in A_s,\ u \neq s,\ v \neq s \tag{15}$$

$$g_v + q_v(1 - x_{s,v}) \geq q_v \qquad \forall (s,v) \in A_s \tag{16}$$

$$t_v - t_u + (T + t_{u,v})(1 - x_{u,v}) \geq t_{u,v} \qquad \forall (u,v) \in A_s,\ u \neq s,\ v \neq s \tag{17}$$

$$t_v + t_{u,v}(1 - x_{s,v}) \geq t_{u,v} \qquad \forall (s,v) \in A_s \tag{18}$$

$$0 \leq g_u \leq Q_s \qquad \forall u \in V_s' \tag{19}$$

Inequalities (12) and (13) state that any node other than s must have exactly one ingoing and one outgoing arc. Equality (14) ensures that every tour must finish at s or more precisely that s has the same number of ingoing and outgoing arcs. Inequalities (15) and (16) are the Miller-Tucker-Zemlin constraints that calculate the amounts of goods delivered up to node v. The domains of variables g_v (19) ensure that the capacity Q_s of a vehicles is not exceeded. Likewise inequalities (17) and (18) are used to calculate the traveling times up to each node v as defined above.

We can further strengthen $\mathrm{VRP}(G_s)$ by the following inequalities from [20]:

$$Q_s \sum_{u \in V_s'} x_{u,s} \geq \sum_{v \in V_s'} q_v \tag{20}$$

$$\sum_{u,v \in U,\ u \neq v} x_{u,v} \leq |U| - \left\lceil \frac{\sum_{u \in U} q_u}{Q_s} \right\rceil \qquad \forall U \subseteq V_s' \tag{21}$$

In our implementation we initially provide inequalities (21) for subsets U of cardinality two and three, but do not separate the more general ones as cuts.

5 Logic-Based Benders' Decomposition for 2L-VRP-TL

Hooker [3] generalized classical BD to logic-based BD by replacing the LP dual with a so-called *inference dual*. Benders' cuts need not to be linear inequalities anymore but are more general functions. Benders' subproblems may then involve discrete variables and nonlinear functions.

We apply this approach here and decompose the above MIP model for 2L-VRP-TL into a master problem corresponding to the first-level VRP augmented with Benders' cuts and a Benders' subproblem that decouples into a set of $|V_0'|$ independent second-level VRPs. More specifically, our *master problem* is

(MP)

$$\text{minimize} \quad \sum_{(u,v)\in A_0} c_{u,v}\, x_{u,v} + \sum_{s\in V_0'} c_s \tag{22}$$

$$\text{s.t.} \quad (x(A_0), t(V_0')) \in \mathrm{VRP}(G_0) \tag{23}$$

$$c_s \geq \beta_{t_s^k}(t_s) \qquad k \in K_s,\ s \in V_0' \tag{24}$$

$$0 \leq t_s \leq T \qquad \forall s \in V_0' \tag{25}$$

$$0 \leq c_s \qquad \forall s \in V_0' \tag{26}$$

$$x_{u,v} \in \{0,1\} \qquad \forall (u,v) \in A_0 \tag{27}$$

It only considers the first-level decision variables $x_{u,v}$ and t_s associated with G_0 and new variables c_s representing (upper bounds for) the total cost of the second-level tours in G_s for each satellite depot s. Inequalities (24) are the Benders' cuts relating c_s with t_s in order to ultimately ensure feasibility and optimality.

The associated *Benders' subproblem* to be solved for deriving Benders' cuts assumes the above master problem variables t_s to be fixed to some current values t_s^k and becomes for each $s \in V_0'$

$(\mathrm{SP}_s(t_s^k))$

$$\text{minimize} \quad \sum_{(u,v)\in A_s} c_{u,v}\, x_{u,v} \tag{28}$$

$$\text{s.t.} \quad (x(A_s), t(V_s')) \in \mathrm{VRP}(G_s) \tag{29}$$

$$0 \leq t_v \leq T - t_s^k \qquad \forall v \in V_s' \tag{30}$$

$$x_{u,v} \in \{0,1\} \qquad \forall (u,v) \in A_s \tag{31}$$

Thus, a minimum cost VRP-solution on G_s with delivery times at most $T - t_s^k$ shall be found for each $s \in V_0'$.

In general, Benders' algorithm starts by solving MP with none or only a small set of initial Benders' cuts. This yields values for the MP variables, i.e., t_s^k, for which the subproblem and its dual are solved in order to derive one or more cuts. These are added to the MP and the whole process is iterated until no further violated cuts exist. It has been shown that when the master problem as well as the duals and the associated inference duals are always solved to optimality, an optimal solution for the original problem will be obtained [3].

The inference dual of subproblem $(\mathrm{SP}_s(t_s^k))$ is

$(\mathrm{DSP}_s(t_s^k))$

$$\text{maximize} \quad \beta_s \tag{32}$$

$$\text{s.t.} \quad (x(A_s), t(V'_s)) \in \mathrm{VRP}(G_s) \wedge (t_v \leq T - t_s^k \; \forall v \in V'_s)$$
$$\xrightarrow{\{0,1\}^{|A_s|}, [0,T]^{|V'_s|}} \sum_{(u,v) \in A_s} c_{u,v}\, x_{u,v} \geq \beta_s \tag{33}$$

i.e., to find the best possible lower bound β_s on the cost $\sum_{(u,v)\in A_s} c_{u,v}\, x_{u,v}$ that can be inferred from the constraints, assuming the fixed t_s^k.

The heart of Logic-based Benders' decomposition is now to derive from this result a more general function $\beta_{t_s^k}(t_s)$ that gives a valid lower bound on the optimal value of the cost $\sum_{(u,v)\in A_s} c_{u,v}\, x_{u,v}$ for any given value of t_s, ideally with $\beta_{t_s^k}(t_s^k)$ corresponding to the optimal solution value of $\mathrm{SP}_s(t_s^k)$. This function $\beta_{t_s^k}(t_s)$ then directly yields a corresponding Benders' cut (24).

Fortunately, in our case the situation is relatively simple. We can observe that increasing or decreasing t_s^k results in stronger or weaker constraints for SP_s, respectively, and consequently the subproblem's cost will weakly monotonically increase with t_s^k. Consider a current $\mathrm{SP}_s(t_s^k)$ and assume it is bounded and non-empty and thus has an optimal solution. Let c_s^k be this solution's cost. From the previous observations we can define a Benders' cut

$$c_s \geq \beta_{t_s^k}(t_s) = \begin{cases} c_s^k & \text{if } t_s \geq t_s^k \\ 0 & \text{else.} \end{cases} \tag{34}$$

Intuitively this means that the costs are at least c_s^k or we need more than $T - t_s^k$ time for the subproblem, i.e., $c_s \geq c_s^k \;\vee\; t_s < t_s^k$. As we want to solve the MP by a MIP-solver, this logic-based Benders' cut is translated into the following pair of linear inequalities

$$c_s \geq c_s^k\, \chi_k \tag{35}$$

$$t_s \leq (t_s^k - \varepsilon)\,(1 - \chi_k) \tag{36}$$

with $\chi_k \in \{0,1\}$ being a new decision variable that is also added to MP and ε being a small constant to ensure $c_s \geq c_s^k$ in case of $t_s = t_s^k$.

In general, it might happen that $\mathrm{SP}_s(t_s^k)$ is infeasible. Then, $\mathrm{DSP}_s(t_s^k)$ is unbounded and a *feasibility cut* – in contrast to above *optimality cut* – needs to be derived, which is a condition that cuts away the current t_s^k from MP. In our case, however, we avoid infeasible subproblems by initially determining a minimum time required for each subproblem $\mathrm{SP}_s,\ s \in V'_0$ to be solvable and limiting t_s correspondingly. As we can safely assume that the triangle inequality holds for travel times, a minimum time solution is achieved by visiting each customer by an individual vehicle directly from the depot, i.e.,

$$t_s \leq T - \max_{v \in V'_s} t_{s,v} \quad \forall s \in V'_0. \tag{37}$$

To start with a more meaningful initial MP, general lower bounds for the subproblem costs c_s are determined by solving $\mathrm{SP}_s(0)$ and requiring $c_s \geq c_s^k \;\forall s \in V'_0$.

To avoid unnecessary recalculations, we store all solved subproblems with their optimal solutions $(t_s^k, x^k, c_s^k, \hat{t}_s^k),\ \forall k \in K_s,\ s \in V_0'$, with

$$\hat{t}_s^k = T - \sum_{(u,v)\in A_s,\ v\neq s} t_{u,v}\, x_{u,v} \tag{38}$$

being the latest possible time for t_s for which this subproblem solution x^k would still be feasible and optimal; note that $t_s^k \leq \hat{t}_s^k$. A new subproblem $\mathrm{SP}_s(t_s^l)$ only needs to be processed if there is no stored solution with $t_s^k \leq t_s^l \leq \hat{t}_s^k$.

When solving $\mathrm{SP}_s(t_s^l)$, a possibly existing record $(t_s^k, x^k, c_s^k, \hat{t}_s^k)$ with the largest t_s^k less than t_s^l yields a lower bound on the costs and a possibly existing record $(t_s^{k'}, x^{k'}, c_s^{k'}, \hat{t}_s^{k'})$ with the smallest t_s^k larger than t_s^l yields an upper bound, i.e.,

$$c_s^k \leq c_s^l \leq c_s^{k'} \tag{39}$$

can be added as strengthening inequalities, and $x^{k'}$ can be used as initial heuristic solution to speed up the optimization.

Finally, when the solution x^l to $\mathrm{SP}_s(t_s^l)$ has cost c^l that were already encountered at an earlier instance $\mathrm{SP}_s(t_s^k)$, i.e., $\exists k \in K_s \mid c_s^l = c_s^k$, the corresponding records can be merged to

$$(\min(t_s^k, t_s^l), x^k, c_s^k, \hat{t}_s^k) \quad \text{if } \hat{t}_s^k \geq \hat{t}_s^l \tag{40}$$

$$(\min(t_s^k, t_s^l), x^l, c_s^k, \hat{t}_s^l) \quad \text{else,} \tag{41}$$

and the already existing Benders' cut $c_s \geq \beta_{t_s^k}(t_s)$ is adapted (lifted) accordingly without introducing a new cut.

6 Metaheuristic Improvements

The subproblems as well as the master problem we obtain in above decomposition are much smaller than the original 2L-VRP-TL, and therefore there might be hope that they can be solved to proven optimality in practice. However, all these are still NP-hard problems, and we pay the decomposition by usually having to solve many instances of the master and subproblems.

For generally improving scalability to larger instances, we can turn the exact BD approach into a faster approximate one by solving the subproblems and/or the master problem only approximately. When we terminate the MIP-solver on each of these instances early after reaching a solution with costs that are guaranteed to not exceed a specified optimality gap of $p\%$ and we obtain a feasible final solution, we can be sure that this solution's cost also does not exceed an optimal value by more than $p\%$.

While this might be a practical approach in some cases, the MIP-solver will often still require too much time to obtain approximate solutions with reasonable quality guarantees. In fact, experiments indicated in our scenario that only very moderate speedups could be achieved when allowing a gap of 5%. Suitable metaheuristics appear to be a more promising alternative.

6.1 Heuristic BD

We might consider virtually any well-working metaheuristic for the VRP with time-windows which is not too slow to approach our master and subproblem. For our proof-of-concept experiments here, we decided to apply the following previous work for the periodic VRP with time-windows [21]:

- One initial solution is created by Clarke and Wright's savings algorithm [22], which is adapted in a straight-forward way to only merge feasible routes w.r.t. the time limits.
- A set of n_{init} further, diverse initial solutions is derived by applying a randomized variant of the savings algorithm. The savings of combining two tours is accepted as the currently best savings if its value multiplied by a uniformly distributed random value within $[0.7, 1.3]$ is greater than the previously best known savings.
- The best initial solution undergoes *variable neighborhood descent* [23] using the following neighborhood structures in this order: *intra-route 2-opt*, *intra or-opt* (sequences of one, two, or three stations are moved to another position), and *inter-route 2-opt** (exchange of all feasible end-segments among two routes); for details see [21]. A first-improvement strategy is applied and the procedure only stops after reaching a locally optimal solution w.r.t. all these neighborhoods. Each candidate solution is checked for feasibility concerning the time limits and only feasible solutions are accepted.

In the BD, this metaheuristic can directly replace the exact resolution of the master and subproblems by the MIP-solver. However, we must take care in the bookkeeping of already known solutions $(t_s^k, x^k, c_s^k, \hat{t}_s^k)$ as they are not necessarily optimal anymore. On the one hand, a later identified solution for a time t_s^l may dominate earlier solutions $t_s^k < t_s^l$ even with lower cost, i.e., $c_s^k < c_s^l$. Thus, existing entries need to be verified and must possibly be removed together with the corresponding cuts. On the other hand, it may also happen that a newly found solution $(t_s^l, x^l, c_s^l, \hat{t}_s^l)$ has higher cost than an already known solution $(t_s^k, x^k, c_s^k, \hat{t}_s^k)$ with $\hat{t}_s^k \geq t_s^l$. In this case, no new violated cut can be derived, we may just store $(t_s^k + \varepsilon, x^l, c_s^l, \hat{t}_s^l)$ and the corresponding cut if $\hat{t}_s^l \geq \hat{t}_s^k$ for possible future use.

7 Computational Experiments

We compare the performance of directly solving the MIP model (7)–(11) for 2L-VRP-TL, the MIP-based exact BD approach, and two variants of the heuristic BD on a set of synthetic Euclidean instances and instances based on the TSPlib[3].

All algorithms have been implemented with GCC 4.6. Each test run was performed on a single core of an Intel Xeon E5540 machine with 2.53 GHz. CPLEX version 12.1 was used for solving the MIPs.

[3] https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95

7.1 Instances

The synthetic Euclidean instances can be divided into four subtypes:

– instances where the first-level VRP and the second-level VRPs are equally large, i.e. $|V_0| = |V_s|,\ \forall s \in V_0'$,
– instances with a larger first-level VRP, i.e. $|V_0| > |V_s|,\ \forall s \in V_0'$,
– instances with larger second-level VRPs, i.e. $|V_0| < |V_s|,\ \forall s \in V_0'$, and
– instances where each of the second-level VRPs has different size, i.e. $|V_s| = \lceil 0.5|V_0| \rceil + 1, \ldots, \lfloor 1.5|V_0| \rfloor$.

For the first-level VRP, satellites are randomly placed on a 201×201 grid with the depot node being located at the center. Each second-level VRP is constructed essentially in the same way considering a separate grid of the same size: The satellite is assumed to be at the center, and all customers are placed randomly at the grid. Traveling times are rounded Euclidean distances, and traveling costs are derived from these times by adding uniform random perturbations of 20%. Demands are chosen randomly from $\{1, \ldots, 100\}$. The vehicle capacity and the global time limit were selected manually in a way that the instances are non-trivial.

For the TSPlib instances we applied a clustering that roughly simulates the process of opening satellite depots in real-world. Given the basic nodes which represent customers, we added satellites manually at plausible locations and then assigned each customer node to the closest satellite. As a result, the sizes of the second-level VRPs differ to a certain degree. Customer demands are chosen randomly from $\{1, \ldots, 10\}$ while traveling times and costs, vehicle capacity and the global time limit are determined in the same way as above. All instances are available online[4].

7.2 Results

Tables 1 and 2 compare the following algorithm variants: directly solving the MIP model (7)–(11) ("pure MIP"), the MIP-based exact BD, the BD variant where the subproblems are solved heuristically, and the fully heuristic BD variant where the master problem as well as the subproblems are solved heuristically. Synthetic instances are specified by the size of the master problem ($|V_0|$) and the size of the subproblems ($|V_s|$). For the TSPlib problems we only list $|V_0|$ since the subproblems have different sizes.

Table 1 shows the objective values of final solutions and the required CPU times. Best values are printed bold for each instance. For the exact BD variants, we list the gaps between lower and upper bounds after reaching the time limit of one hour. For the variants where we use heuristics, 30 independent runs were performed in order to obtain average objective values of final solutions and standard deviations. The time limit was set to 10min. Table 2 displays for the BD variants further information on the number of added Benders' cuts and the number of times the master problem is (re-)solved.

First of all, we observe that the pure MIP approach is only viable for small instances where the size of the sub-VRPs is at most 15. For larger instances the gaps are soon too large for the solutions to be meaningful. The exact MIP-based BD performs better on

[4] https://www.ads.tuwien.ac.at/w/Research/Problem_Instances

Table 1. Solution qualities and CPU-times of different algorithm variants.

Synthetic instances		pure MIP			BD using MIP			heur. BD for subp.			fully heuristic BD		
$\|V_0\|$	$\|V_s\|$	obj	gap[%]	time[s]	obj	gap[%]	time[s]	$\overline{\text{obj}}$	sd	time[s]	$\overline{\text{obj}}$	sd	time[s]
5	5	**2844.0**	0.0	0.0	**2844.0**	0.0	0.1	**2844.0**	0.0	0.0	**2844.0**	0.0	0.0
5	3	**1450.0**	0.0	0.0	**1450.0**	0.0	0.0	**1450.0**	0.0	0.0	**1450.0**	0.0	0.0
3	5	**1113.0**	0.0	0.0	**1113.0**	0.0	0.1	**1113.0**	0.0	0.0	**1113.0**	0.0	0.0
5	4...7	**2514.0**	0.0	0.0	**2514.0**	0.0	0.1	**2514.0**	0.0	0.0	**2514.0**	0.0	0.0
9	9	8250.0	0.8	0.2	**8235.0**	0.0	0.3	**8235.0**	0.0	0.1	**8235.0**	0.0	0.1
9	5	**5304.0**	0.0	0.0	**5304.0**	0.0	0.1	**5304.0**	0.0	0.0	**5304.0**	0.0	0.0
5	9	**5376.0**	0.0	0.1	**5376.0**	0.0	0.1	5400.0	0.0	0.0	5400.0	0.0	0.0
9	6...13	**9337.0**	0.2	1.9	**9337.0**	0.0	0.5	**9337.0**	0.0	0.2	9359.0	0.0	0.2
15	15	18367.0	4.7	600.0	**17518.0**	0.0	49.0	17704.8	34.4	7.7	17709.0	24.3	1.7
15	9	11270.0	0.5	5.2	**11268.0**	0.0	4.3	11271.2	12.0	5.4	11265.4	10.8	0.5
8	15	9307.0	0.6	600.0	**9093.0**	0.0	41.8	9096.7	6.5	0.6	9094.9	4.3	0.6
15	9...22	21506.0	2.5	900.0	22218.0	15.0	24.7	21148.5	49.4	2.3	**21147.6**	54.3	1.6
25	25	40708.0	22.1	3600.0	42299.0	15.0	3590.0	-	-	-	**40000.4**	63.2	26.0
25	14	23932.0	14.9	3600.0	25817.0	15.0	4.9	25203.2	24.9	577.0	**23384.0**	34.8	11.3
13	25	18880.0	21.6	3600.0	-	-	3600.0	**18788.4**	23.1	76.4	18793.9	29.6	10.5
25	14...37	38881.0	23.5	3600.0	-	-	3600.0	39138.2	332.4	600.0	**38173.0**	63.3	38.9
35	35	70467.0	36.0	3600.0	-	-	3600.0	-	-	600.0	**60630.8**	84.8	136.3
35	19	39536.0	19.2	3600.0	42343.0	15.0	236.4	**30474.6**	133.9	590.0	38517.2	57.2	40.9
18	35	31275.0	28.5	3600.0	-	-	3600.0	-	-	600.0	**30061.0**	63.0	56.7
35	19...52	69238.0	34.7	3600.0	-	-	3600.0	-	-	600.0	**61041.1**	87.9	236.8
50	50	234161.0	69.2	3600.0	-	-	3600.0	-	-	600.0	**100823.7**	356.3	600.0
50	27	67746.0	26.9	3600.0	-	-	3600.0	-	-	600.0	**62687.1**	81.0	316.3
26	50	76159.0	51.4	3600.0	-	-	3600.0	-	-	600.0	**52019.9**	60.8	331.1
50	27...75	226549.0	67.3	3600.0	-	-	3600.0	-	-	600.0	**105743.4**	702.4	600.0

TSPlib instances		pure MIP			BD using MIP			heur. BD for subp.			fully heuristic BD		
Name	$\|V_0\|$	obj	gap[%]	time[s]	obj	gap[%]	time[s]	$\overline{\text{obj}}$	sd	time[s]	$\overline{\text{obj}}$	sd	time[s]
a280	20	6004.0	23.3	3600.0	**5762.0**	0.0	1176.1	5798.2	9.6	33.6	5798.5	11.5	6.5
berlin52	6	**13784.0**	3.1	3600.0	**13784.0**	0.0	12.8	13784.8	2.0	0.2	13786.1	4.8	0.2
bier127	13	204856.0	7.1	3600.0	**204361.0**	0.0	180.5	204464.3	94.1	8.5	204407.8	163.9	1.0
ch130	11	13727.0	7.8	3600.0	**13704.0**	0.0	558.6	13747.8	19.1	1.1	13751.2	24.8	1.1
pr1002	16	2347399.0	80.3	3600.0	-	-	3600.0	18158.8	39.0	389.1	**18153.7**	41.1	389.9
rat783	13	64684.0	81.2	3600.0	-	-	3600.0	**715598.6**	1261.9	265.6	716024.0	1242.4	249.3

Table 2. Numbers of generated cuts and master problem resolves of the BD variants.

Synthetic instances							
		BD using MIP		heur. BD for subp.		fully heuristic BD	
$\|V_0\|$	$\|V_s\|$	#cuts	#resolves	#cuts	#resolves	#cuts	#resolves
5	5	4.0	3.0	4.0	3.0	4.0	3.0
5	3	1.0	2.0	1.0	2.0	1.0	2.0
3	5	3.0	3.0	3.0	3.0	3.0	3.0
5	4...7	3.0	2.0	3.0	2.0	3.0	2.0
9	9	10.0	3.0	3.0	2.0	7.0	3.0
9	5	5.0	3.0	5.0	3.0	4.0	3.0
5	9	3.0	2.0	3.0	2.0	3.0	2.0
9	6...13	10.0	4.0	9.0	3.0	9.4	3.3
15	15	32.0	7.0	30.3	7.1	27.6	6.6
15	9	34.0	6.0	34.9	7.0	22.0	4.1
8	15	12.0	5.0	10.2	5.0	10.2	5.0
15	9...22	13.0	2.0	28.0	6.8	25.8	6.3
25	25	16.0	2.0	-	-	87.9	11.5
25	14	12.0	2.0	31.7	1.0	104.9	14.5
13	25	-	-	41.9	8.5	39.6	7.9
25	14...37	-	-	47.4	2.1	105.4	12.9
35	35	-	-	-	-	125.9	12.3
35	19	17.0	2.0	29.0	3.2	165.4	13.2
18	35	-	-	-	-	56.2	11.1
35	19...52	-	-	-	-	182.8	16.9
50	50	-	-	-	-	138.4	8.4
50	27	-	-	-	-	251.4	22.3
26	50	-	-	-	-	76.9	12.3
50	27...75	-	-	-	-	113.4	4.1

instances with sub-VRPs with up to 15 nodes. However, on larger instances it is often not able to solve all subproblems of the first major iteration in time, and we therefore do not get any feasible solution for the master problem. For instances with sub-VRPs of size 25 or more, this even holds when terminating CPLEX early with an optimality gap limit of 15%. We remark that this rather bad behavior is particularly due to the relatively weak Miller-Tucker-Zemlin formulation, and one can expect to improve the situation by using a more state-of-the-art exact VRP solver. When using the heuristic approach for the BD subproblems, they are solved in a relatively short time to very reasonable quality so that a feasible solution to the master problem can be obtained most of the time. However, on the larger instances it proves to be difficult nonetheless to solve the master problem via MIP and in some cases it was not possible to solve it even once within the time limit. The fully heuristic BD works well on small instances where optimal solutions are reliably reached in short times. On larger instances it has excellent scalability and produces by far the best results.

Comparing the different subtypes of synthetic instances, we clearly see that the most challenging ones are those where the master problem and the subproblems are equally large. The pure MIP approach and the exact BD approach are able to solve instances with small subproblems in comparison faster since they can concentrate on the master problem. This is expressed by the low number of Benders' cuts that are added and thus the low number of times that the master problem has to be (re-)solved, see Table 2. Instances where subproblems have different sizes are usually also easier to solve. The reason here is that the master problem often is easier: In the first-level VRP, the satellites of larger subproblems typically need to be visited earlier than satellites of smaller subproblems. In Table 2 we also observe that for larger instances, the fully heuristic BD variant is able to perform much more iterations, i.e., more Bender's cuts are added and the master problem is more often resolved within the same time limit.

8 Conclusions

Logic-based BD is a promising extension of classical BD to tackle far more problems from practice because subproblems are not restricted to LPs anymore. Its application to our 2L-VRP-TL is relatively intuitive and Benders' cuts can be derived from primal subproblem solutions and its inference dual in a rather straight-forward way.

Applying the logic-based BD with CPLEX for exactly solving all master and subproblem instances turned out to be beneficial for some mid-size instances in comparison to solving the original MIP formulation directly. Problematic, however, are the long running times for solving the Benders' subproblems, preventing the approach from finding feasible solutions to larger 2L-VRP-TL instances at all. By just aiming for good approximate solutions and using a variant of variable neighborhood search to heuristically solve the master and subproblem instances, we could dramatically improve the scalability and obtain by far the best results on the considered instances.

More generally, it was shown that logic-based BD may be a fruitful framework also for metaheuristics. While previous work already documented the usefulness of metaheuristics for approximately solving the master problem in classic BD approaches, our work goes beyond and applies a metaheuristic especially to more complex subproblems (and their inference duals) as they appear in logic-based BD. The general technique seems to be promising also for other classes of problems and deserves further research.

Our implementation for 2L-VRP-TL only is a first proof-of-concept. It is obvious that it can be improved on the one hand by utilizing a tighter MIP-formulation for $\mathrm{VRP}(G_s)$, e.g., based on multi-commodity flows, or even a more sophisticated branch-and-cut. On the other hand a more advanced metaheuristic may also be chosen for $\mathrm{VRP}(G_s)$. The principles of the logic-based BD and the combination with the metaheuristic, however, stay the same.

Future work should in particular investigate a combined application of heuristic and exact methods for solving the Benders' subproblems. For example, one can first solve the subproblems heuristically yielding approximate Benders' cuts and a heuristic solution quickly. In a second phase, the existing Benders' cuts are iteratively validated by solving corresponding subproblems exactly, exploiting the already known heuristic solutions. Possibly found improved subproblem solutions yield new exact Benders' cuts that replace the dominated heuristic cuts. When resolving the master problem and validating all Benders' cuts in this way, an exact solution is obtained in the end.

References

1. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik **4** (1962) 238–252
2. Geoffrion, A.M.: Generalized Benders decomposition. Journal of Optimization Theory and Applications **10**(4) (1972) 237–260

3. Hooker, J.N., Ottosson, G.: Logic-based benders decomposition. Mathematical Programming **96** (2003) 33–60
4. Hooker, J.N.: Planning and scheduling by logic-based Benders decomposition. Operations Research **55**(3) (2007) 588–602
5. Poojari, C.A., Beasley, J.E.: Improving Benders decomposition using a genetic algorithm. European Journal of Operational Research **199**(1) (2009) 89–97
6. Lai, M.C., Sohn, H.s., Tseng, T.L., Chiang, C.: A hybrid algorithm for capacitated plant location problem. Expert Systems with Applications **37**(12) (2010) 8599–8605
7. Lai, M.C., Sohn, H.S.: Using a genetic algorithm to solve the Benders' master problem for capacitated plant location. In Gao, S., ed.: Bio-Inspired Computational Algorithms and Their Applications. InTech (2012) 405–420
8. Lai, M.C., Sohn, H.S., Tseng, T.L., Bricker, D.L.: A hybrid Benders/genetic algorithm for vehicle routing and scheduling problem. International Journal of Industrial Engineering **19**(1) (2012) 33–46
9. Rei, W., Cordeau, J.F., Gendreau, M., Soriano, P.: Accelerating Benders decomposition by local branching. INFORMS Journal on Computing **21**(2) (2008) 333–345
10. Zakeri, G., Philpott, A.B., Ryan, D.M.: Inexact cuts in Benders decomposition. SIAM Journal on Optimization **10**(3) (1999) 643–657
11. Perboli, G., Tadei, R., Vigo, D.: The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. Transportation Science **45**(3) (2011) 364–380
12. Hemmelmayr, V.C., Cordeau, J.F., Crainic, T.G.: An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Computers and Operations Research **39**(12) (2012) 3215–3228
13. Schwengerer, M., Pirkwieser, S., Raidl, G.R.: A variable neighborhood search approach for the two-echelon location-routing problem. In Hao, J.K., Middendorf, M., eds.: Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2012. Volume 7245 of LNCS., Springer (2012) 13–24
14. Crainic, T.G., Mancini, S., Perboli, G., Tadei, R.: Multi-start heuristics for the two-echelon vehicle routing problem. In Merz, P., Hao, J.K., eds.: Evolutionary Computation in Combinatorial Optimization – EvoCOP 2011. Number 6622 in LNCS, Springer (2011) 179–190
15. Jacobsen, S.K., Madsen, O.B.G.: A comparative study of heuristics for a two-level routing-location problem. European Journal of Operational Research **5**(6) (1980) 378–387
16. Boccia, M., Crainic, T., Sforza, A., Sterle, C.: A metaheuristic for a two echelon location-routing problem. In Festa, P., ed.: Experimental Algorithms. Volume 6049 of LNCS. Springer (2010) 288–301
17. Boccia, M., Crainic, T.G., Sforza, A., Sterle, C.: Location-routing models for designing a two-echelon freight distribution system. Technical Report CIRRELT-2011-06, University of Montreal (2011)
18. Fisher, M.L., Jaikumar, R.: A generalized assignment heuristic for vehicle routing. Networks **11**(2) (1981) 109–124
19. Boschetti, M., Maniezzo, V.: Benders decomposition, Lagrangian relaxation and metaheuristic design. Journal of Heuristics **15** (2009) 283–312
20. Yaman, H.: Formulations and valid inequalities for the heterogeneous vehicle routing problem. Mathematical Programming **106**(2) (2006) 365–390
21. Pirkwieser, S., Raidl, G.R.: A variable neighborhood search for the periodic vehicle routing problem with time windows. In Prodhon, C., et al., eds.: Proceedings of the 9th EU/MEeting on Metaheuristics for Logistics and Vehicle Routing, Troyes, France (2008)
22. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research **12**(4) (1964) 568–581
23. Mladenović, N., Hansen, P.: Variable neighborhood search. Computers and Operations Research **24**(11) (1997) 1097–1100