Two-Phase Branch-and-Cut for the Mixed Capacitated General Routing Problem

Claudia Bode[a], Stefan Irnich[a], Demetrio Laganà[b], Francesca Vocaturo[c]

[a] *Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.*
[b] *Department of Mechanical, Energy and Management Engineering, University of Calabria, 87036 Arcavacata di Rende (CS), Italy.*
[c] *Department of Economics, Statistics and Finance, University of Calabria, 87036 Arcavacata di Rende (CS), Italy.*

Abstract

The Mixed Capacitated General Routing Problem (MCGRP) is defined over a mixed graph, for which some vertices must be visited and some links must be traversed at least once. The problem consists of determining a set of least-cost vehicle routes that satisfy this requirement and respect the vehicle capacity. Few papers have been devoted to the MCGRP, in spite of interesting real-world applications, prevalent in school bus routing, mail delivery, and waste collection. This paper presents a new mathematical model for the MCGRP based on two-index variables. The approach proposed for the solution is a two-phase branch-and-cut algorithm, which uses an aggregate formulation to develop an effective lower bounding procedure. This procedure also provides strong valid inequalities for the two-index model. Extensive computational experiments over benchmark instances are presented.

Key words: general routing problem, mixed graph, integer programming, branch-and-cut algorithm

1. Introduction

This paper presents a new exact algorithm for the *Mixed Capacitated General Routing Problem* (MCGRP) based on branch-and-cut (B&C). The MCGRP generalizes the single-vehicle and multiple-vehicle *General Routing Problems* (GRPs) and the *Capacitated Arc Routing Problem* (CARP).

GRPs constitute a class of vehicle-routing problems, in which a single vehicle or a fleet of vehicles must serve both a subset of links and a subset of vertices of a given graph. GRPs have interesting practical applications, prevalent in waste collection, postal delivery and school bus routing. For instance, in an urban waste collection plan, the collection along a street may be modeled by means of links that must be traversed, whereas the collection occurring in specific points (e.g., hospitals or multi-storey apartment blocks) may be modeled by means of vertices that must be visited. Similarly, in the postal delivery services, depending on their demand and dispersion, customers may be modeled as individual vertices or groups of customers as street segments (edges or arcs). Finally, in school bus routing, several children living on the same street may be picked up either by stopping close to each ones home, implying a service on the respective street segments, or groups of them may walk from their home to a specific bus stop imposing just one stop.

The single-vehicle GRP was introduced by Orloff (1974) and shown to be $\mathcal{NP}$-hard by Lenstra and Rinnooy Kan (1976). Most works refer to the uncapacitated case. Specifically, Letchford (1996, 1999) and Corberán and Sanchis (1998) proposed valid inequalities for the GRP polyhedron. For the same problem, Corberán et al. (2001) described a cutting-plane algorithm based on several classes of facet-inducing inequalities. Reinelt and Theis (2008) studied the 0/1-polytope associated with the uncapacitated GRP defined

Email addresses: claudia.bode@uni-mainz.de (Claudia Bode), irnich@uni-mainz.de (Stefan Irnich), demetrio.lagana@unical.it (Demetrio Laganà), vocaturo@unical.it (Francesca Vocaturo)

over a connected and undirected graph. The contribution of Corberán et al. (2003) for the GRP defined on a mixed graph was a new integer programming formulation and a partial description of the related polyhedron. They reported remarkable computational results obtained by a cutting-plane algorithm. Corberán et al. (2005) considerably improved this algorithm by defining a new family of facet-defining inequalities and new separation procedures. Blais and Laporte (2003) proposed a transformation in order to solve the uncapacitated GRP defined over directed, undirected and mixed graphs. The GRP is transformed into an equivalent traveling salesman problem or rural postman problem and solved by means of available exact algorithms. The approach does not work equally well in all cases; it works best on directed problems and on mixed problems, in which the number of edges is relatively small. The uncapacitated GRP was also modeled by resorting to windy graphs. Corberán et al. (2007, 2008) presented a strong windy general routing polyhedron description and designed a powerful B&C algorithm able to solve a large number of benchmark instances.

The basic multiple-vehicle routing problem is the *Capacitated Vehicle Routing Problem* (CVRP, see Toth and Vigo, 2002, 2014), in which the demand occurs only at vertices. On the contrary, arc routing problems (ARPs, see Dror, 2000; Corberán and Laporte, 2014) are GRPs in which no vertices have to be serviced. While CVRP are defined on complete graphs, ARPs share with GRPs that they are defined on incomplete (often sparse) graphs, which are either undirected, directed, mixed, or windy.

Important contributions for the mixed CARP have been given by Belenguer et al. (2006). They presented a linear formulation, developed a lower bounding procedure based on valid inequalities, and described some upper bounds obtained through three constructive heuristics and a memetic algorithm. Gouveia et al. (2010) described a compact flow-based model for the mixed CARP and derived an aggregate lower bounding model. Moreover, they introduced a set of valid inequalities for the linear programming relaxation of the integer model and presented promising computational results.

Note that GRPs can be transformed into CARPs by adding loops, i.e., edges $\{i, i\}$ or arcs (i, i) to the underlying graph whenever in the GRP instance a vertex i has to be serviced. The edge or arc receives the same demand as the vertex that it substitutes. In this sense, then the mixed CARP and the MCGRP can be considered identical, at least if the mathematical formulation and solution approach is capable of handling loops. To the best of our knowledge, this equivalence has not yet been utilized.

The problem studied and solved in the paper at hand is the MCGRP. It may cause confusion that sometimes the MCGRP is referred to as *Capacitated General Routing Problem on mixed graphs*(CGRP or CGRP-m) and *Node, Edge and Arc Routing Problem* (NEARP). There exist lower bounding procedures and tailored exact algorithms for its solution (Bach et al., 2013; Bach, 2014; Bosco et al., 2013; Gaze, 2013; Gaze et al., 2013). Other studies present non-exact approaches tackling the problem. Particularly, Pandit and Muralidharan (1995) described a heuristic procedure which starts with a sub-graph obtained from the original one by considering only the links that must be traversed and the vertices that must be visited. Since the sub-graph is generally disconnected, the connection is reached by adding to it the shortest paths linking two vertices of disjoint connected components. The sub-graph is then converted into an Eulerian graph which admits a giant tour. A feasible solution is obtained by cutting the giant tour into smaller tours satisfying the capacity constraints. Gutiérrez et al. (2002) introduced an alternative procedure, based on the partition-first-route-next paradigm, improving previous results. Prins and Bouchenoua (2005) described a memetic algorithm for the MCGRP. Bosco et al. (2014) introduced a matheuristic algorithm for the MCGRP where the exact algorithm of Bosco et al. (2013) is incorporated in some steps of a neighborhood search. Hasle et al. (2012) carried out a computational study on three large scale MCGRP datasets. Finally, an extension of the MCGRP was tackled by Bräysy et al. (2011).

We propose an alternative exact approach to solve the MCGRP which combines beneficial ingredients from existing procedures in an effective way. The novelty of the approach substantially comprises two aspects. First, it is based on a new MCGRP formulation which uses two-index variables also to model the link flow. Second, it takes advantage from all results of a lower bounding procedure. This procedure produces, besides excellent lower bounds, valid inequalities that are used to initialize a B&C scheme.

The remainder of the paper is organized as follows. In Section 2, a formal definition and the new two-index formulation of the MCGRP are given. In Section 3.1, we present the lower bounding formulation used in the exact approach illustrated in Section 3 in order to determine lower bounds and general cuts. Section 4 presents computational results. Final conclusions are drawn in Section 5.

2. Problem Description and Formulation

A formal definition of the MCGRP relies on a mixed graph $G = (V, E, A)$ with vertices set V, edges set E and arcs set A. Vertex $1 \in V$ represents the depot, at which a set K of homogeneous vehicles with capacity Q is based. The remaining vertices form the set $C = V \backslash \{1\}$. Every element $b \in V \cup E \cup A$ has a demand $q_b \geq 0$, those elements with strictly positive demand are *required*, meaning that they must be serviced exactly once. Required vertices are in $V_R = \{v \in C : q_v > 0\}$, required edges are in $E_R = \{e \in E : q_e > 0\}$, and required arcs are in $A_R = \{a \in A : q_a > 0\}$. In order to ensure feasibility, we assume that the demand q_r of each required element r does not exceed Q.

For notational ease, we speak of links when we want to refer to both edges and arcs in $E \cup A$. Any link can be *deadheaded*, i.e., traversed without being serviced, any number of times. The traversal of a link $\ell \in E \cup A$ results in a non-negative traversal cost c_ℓ. In the following, required elements are referred to as $r \in V_R \cup E_R \cup A_R$ when distinction is not essential.

The MCGRP is the problem of finding minimum-cost vehicle tours, each starting and ending at the depot, that together serve all required elements exactly once, and respect the vehicle capacity.

In order to state the MCGRP models, we introduce further notation used throughout the paper: Let $S \subseteq V$ be a subset of vertices. We denote by $\delta^+(S)$ the set of arcs leaving S, by $\delta^-(S)$ the set of arcs entering S, by $\delta_R^+(S)$ the set of required arcs leaving S, by $\delta_R^-(S)$ the set of required arcs entering S, by $\delta(S)$ the set of edges with exactly one endpoint in S, and by $\delta_R(S)$ the set of required edges with exactly one endpoint in S. The associated link sets are $\delta^*(S) = \delta(S) \cup \delta^+(S) \cup \delta^-(S)$ and $\delta_R^*(S) = \delta_R(S) \cup \delta_R^+(S) \cup \delta_R^-(S)$. For the sake of brevity, singleton sets $S = \{i\}$ in the previous notation can be replaced by i so that, e.g., $\delta(i)$ stands for $\delta(\{i\})$. Finally, we denote by $V_R(S)$ the set of required vertices belonging to S, by $A_R(S)$ the set of required arcs with both endpoints in S, and by $E_R(S)$ the set of required edges with both endpoints in S.

We propose a new mathematical model based on variables with two indices, one for the respective vehicles $k \in K$ and the other for referring to an element of $V \cup E \cup A$. Let x_r^k be a binary variable equal to 1 if and only if the required element $r \in V_R \cup E_R \cup A_R$ is serviced by vehicle k. For a link $\ell \in E \cup A$ and a vehicle $k \in K$, let y_ℓ^k be a non-negative variable representing the number of deadheadings through ℓ by vehicle k. For a subset of required links $L \subseteq A_R \cup E_R$, we define $x^k(L) = \sum_{\ell \in L} x_\ell^k$, and for a subset of links $L \subseteq A \cup E$, we define $y^k(L) = \sum_{\ell \in L} y_\ell^k$.

The two-index formulation for the MCGRP reads as follows:

$$\lambda^* = \min \sum_{k \in K} \sum_{\ell \in E \cup A} c_\ell y_\ell^k \quad \left(+ \sum_{\ell \in E_R \cup A_R} c_\ell \right) \tag{1a}$$

$$\sum_{k \in K} x_r^k = 1, \qquad \forall\, r \in V_R \cup E_R \cup A_R \tag{1b}$$

$$\sum_{r \in V_R \cup E_R \cup A_R} q_r x_r^k \leq Q, \qquad \forall\, k \in K \tag{1c}$$

$$x^k(\delta_R^*(i)) + y^k(\delta^*(i)) \equiv even, \qquad \forall\, i \in V,\ k \in K \tag{1d}$$

$$x^k(\delta_R^-(S)) + y^k(\delta^-(S)) - x^k(\delta_R^+(S)) - y^k(\delta^+(S)) - x^k(\delta_R(S)) - y^k(\delta(S)) \leq 0, \qquad \forall\, S \subseteq V,\ k \in K \tag{1e}$$

$$x^k(\delta_R^*(S)) + y^k(\delta^*(S)) \geq 2x_r^k, \qquad \forall\, r \in V_R(S) \cup E_R(S) \cup A_R(S), S \subseteq C,\ k \in K \tag{1f}$$

$$x_r^k \in \{0, 1\}, \qquad \forall\, r \in V_R \cup E_R \cup A_R,\ k \in K \tag{1g}$$

$$y_\ell^k \in \mathbb{Z}_+, \qquad \forall\, \ell \in E \cup A,\ k \in K \tag{1h}$$

The objective (1a) minimizes the total traversal cost. Note that the service costs (in parenthesis) are constant and therefore not relevant for the routing decisions. Constraints (1b) state that each required element must be serviced. Constraints (1c) guarantee the vehicle capacity is never exceeded. Parity constraints (1d) stipulate that each route induces a Eulerian subgraph. This graph must also be balanced, which is formulated with the so-called *balanced set conditions* (1e). Constraints (1f) ensure that each route is connected. In particular, they impose that for each subset of vertices (excluding the depot) containing a link or vertex serviced by a vehicle, at least two links incident to the subset must be traversed; they also eliminate subtours that do not include the depot. Finally, constraints (1g) and (1h) define the domains of the service and deadheading variables.

2.1. Parity Constraints

The parity constraints (1d) are non-linear, and the only known way to completely replace them by linear constraints is the introduction of additional integer variables $d_i \in \mathbb{Z}_+$, one for each vertex $i \in V$. Setting the right-hand side in (1d) to $= 2d_i$ established the task.

However, there exist linear inequalities that partially cover this requirement:

$$x^k(\delta_R^*(S)\backslash H) + y^k(\delta^*(S)) \geq x^k(H) - |H| + 1, \qquad \forall\, k \in K,\ S \subseteq C,\ H \subseteq \delta_R^*(S),\ |H| \text{ odd}. \tag{2}$$

These so-called *blossom inequalities* are an extension of constraints proposed by Belenguer and Benavent (1998) for the CARP. Their validity can be shown as follows: If all required links in H are serviced by the k-th vehicle, i.e, $x^k(H) = |H|$, then, given that $|H|$ is odd, the k-th vehicle must cross $\delta^*(S)$ at least once more. Hence, $x^k(\delta_R^*(S)\backslash H) + y^k(\delta^*(S))$ must be at least 1. Otherwise, if $x^k(H) < |H|$, the inequality is trivial.

2.2. Breaking Symmetry

The formulation just described yields a large number of equivalent solutions. In fact, since all vehicles have the same capacity, for a given solution any permutation of the vehicle indices induces another equivalent solution. In order to avoid equivalent solutions, we introduce additional constraints. Let η be the number of required elements, i.e., $\eta = |V_R \cup E_R \cup A_R|$, and let $K = \{1, 2, \ldots, m\}$. Moreover, let r_t be the t-th required element ($t = 1, \ldots, \eta$). Any rule can be used to order the required elements. Let $s(k)$ be the smallest index of the required elements serviced by vehicle $k \in K$. In order to impose the condition $s(1) \leq s(2) \leq s(3) \leq \ldots \leq s(m)$, the following set of *symmetry breaking constraints* are valid:

$$x_{r_1}^1 = 1 \tag{3a}$$

$$x_{r_t}^k \leq \sum_{j=1,\ldots,t-1} x_{r_j}^{k-1}, \qquad \forall\, t = 2, \ldots, \eta,\ k = 2, \ldots, m \tag{3b}$$

$$x_{r_t}^k = 0, \qquad \forall\, t = 1, \ldots, m-1,\ k = t+1, \ldots, m. \tag{3c}$$

Constraint (3a) states that the first vehicle must serve the first required element. Constraints (3b) stipulate that if the t-th element r_t ($t \geq 2$) is serviced by the k-th vehicle ($k \geq 2$), then at least one required element associated with an index preceding t must be serviced by the vehicle $k-1$. Finally, constraints (3c) state that element r_t ($t \leq m-1$) cannot be serviced by any of the vehicles $k = t+1, \ldots, m$.

These symmetry breaking constraints seem effective for the MCGRP. Other inequalities can be adapted to the problem from the literature (e.g., see Adulyasak et al., 2014).

3. Solution Approach

We propose a B&C algorithm that works in two phases. In the first phase, an aggregate so-called *one-index formulation* that comprises a relaxation of formulation (1) is solved. The linear relaxation of the one-index formulation contains an exponential number of constraints, which have to be identified and added dynamically in a cutting-plane fashion. It typically provides an excellent lower bound that can be further strengthened by adding valid inequalities. Compared to formulation (1), solving the one-index formulation needs only a little fraction of computation time.

In the second phase, the two-index formulation (1) is initialized with the inequalities from the one-index formulation. Since also some of the constraints of the formulation (1) are exponential families of inequalities, the B&C algorithm (see Wolsey, 1998, for an introduction) then adds violated of these and other inequalities. If the solution of the linear problem is not integer or, alternatively, there exists at least one vertex with an odd degree, then the branching decision splits the problem into two complementary subproblems, and the same procedure is applied to each of them recursively. If a subproblem is infeasible or proven to be unprofitable for the search of optimal solutions, it is discarded. Our B&C algorithm also uses an external heuristic procedure to obtain an initial upper bound.

We will start with a more detailed description of the first phase for lower bounding, outline the heuristic that provides upper bounds and feasible solutions, and describe and summarize the major components of the B&C algorithm.

3.1. Lower Bounding and One-Index Formulation

Since the computational effort for solving the MCGRP with model (1) exactly is huge (sometimes prohibitive) it is fundamental to produce tight bounds very fast. In order to obtain good lower bounds, we solve a one-index formulation very similar to the model presented in Belenguer et al. (2006). It solely uses a vector y of aggregated deadheading variables

$$y_\ell = \sum_{k \in K} y_\ell^k \in \mathbb{Z}_+, \tag{4}$$

one for each link $\ell \in E \cup A$.

The major difference to the formulation of Belenguer et al. (2006) is that for the MCGRP the coefficients need to be defined differently: Recall that in the definition of $q(S)$, the demand on vertices is taken into account. Specifically, let $q(S)$ be the total demand of the required elements in $E_R(S) \cup A_R(S) \cup \delta_R^*(S) \cup V_R(S)$. Thus, for any subset S of vertices, let $K(S)$ be the minimum number of vehicles to serve $E_R(S) \cup A_R(S) \cup \delta_R^*(S) \cup V_R(S)$. This number can be approximated by $\lceil q(S)/Q \rceil$ and computed exactly solving a bin-packing problem. Moreover, let $b(S) = |\delta_R^-(S)| - |\delta_R^+(S)| - |\delta_R(S)|$ be the *unbalance* of S. For any link subset $L \subseteq E \cup A$, we define $y(L) = \sum_{\ell \in L} y_\ell$. The linear relaxation of *one-index formulation* reads as follows:

$$\underline{\lambda} = \min c^\top y \quad \left(+ \sum_{\ell \in E_R \cup A_R} c_\ell \right) \tag{5a}$$

$$y(\delta^*(S)) \geq 1, \qquad \forall\ S \subseteq C,\ |\delta_R^*(S)| \text{ odd} \tag{5b}$$

$$y(\delta^*(S)) \geq 2K(S) - |\delta_R^*(S)|, \qquad \forall\ S \subseteq C \tag{5c}$$

$$y(\delta(S)) + y(\delta^+(S)) - y(\delta^-(S)) \geq b(S), \qquad \forall\ S \subseteq V \tag{5d}$$

$$y_\ell \geq 0, \qquad \forall\ \ell \in E \cup A. \tag{5e}$$

The objective (5a) provides a lower bound $\underline{\lambda}$ for λ^*. The odd-cut inequalities (5b) require that at least one link is deadheaded whenever an odd number of required links in $\delta^*(S)$ occurs. The capacity inequalities (5c) require at least $2K(S)$ traversals (services and deadheadings) along some links of $\delta^*(S)$. Balance inequalities (5d) require at least $|\delta_R^-(S)| - |\delta_R^+(S)| - |\delta_R(S)|$ deadheadings if the difference between incoming and outgoing arcs cannot be compensated by edges in the cutset.

Note that we might solve (5a)-(5e) as an integer program by replacing $y_\ell \geq 0$ with $y_\ell \in \mathbb{Z}_+$. However, these integer "solutions" are solutions to a relaxation only. It can happen that there exists no feasible integer solution to the disaggregated model (1) compatible with (4).

At each iteration of a cutting-plane algorithm, we solve a linear program which contains the non-negativity constraints, a subset of (5b)-(5d) constraints, and disjoint-path inequalities (Belenguer et al., 2006, see). Separation routines are identical to those described in next Section 3.3 and seek for a set of valid inequalities violated by the current solution.

3.2. Upper Bounding and Location-based Heuristic

An initial feasible solution for the MCGRP is built on the basis of a partition-first-route-next heuristic. Herein, a feasible partition of the required elements is found by solving a *Capacitated Concentrator Location-based Problem* (CCLP), in which m required elements are selected as concentrator location, and the remaining required elements are grouped around the concentrators. Several location-based heuristics have been proposed in the literature (e.g., Bramel and Simchi-Levi, 1995). Recent discretized formulations for different versions of the CCLP have been provided by Gouveia and Saldanha-da Gama (2006) and Correia et al. (2010).

The following constraints must be fulfilled: (i) a required element is assigned to itself if it is a concentrator; (ii) each required element is assigned to only one concentrator; (iii) each required element may be assigned to another required element if and only if the latter is a concentrator; (iv) the overall demand of the required elements assigned to a concentrator cannot exceed its capacity, that is the vehicle capacity; (v) the number of required elements selected as concentrators is equal to the number of vehicles. The goal is the minimization of the total assignment cost that intended to approximate the routing costs. The cost for assigning a required element to a concentrator is equal to the shortest-path distance between the potential concentrator and the required element. Since the cost matrix associated with G is generally not symmetric, all the feasible shortest paths starting from a required element and ending to any potential concentrator, and vice versa, must be evaluated in order to select the one having the minimum distance. The solution of the CCLP gives a feasible partition of the required elements. An optimal routing associated with each partition is defined by solving an uncapacitated GRP on a mixed graph. We use the same B&C algorithm developed in this paper (second phase) to exactly solve the problem associated with each partition: We have to adjust the definition of the required elements and set $m = 1$.

The weakness of the partition-first-route-next approach lies in the fact that the objective function of the CCLP only approximates the routing costs. To mitigate this effect, an iterative scheme is designed: At each iteration, a set of diversification constraints is added dynamically to the CCLP stipulating the selection of a different concentrator set. More precisely, let $\mathcal{C}$ be the set of concentrators. Then, a feasible MCGRP solution, whose cost is denoted by $\lambda(\mathcal{C})$, remains associated with $\mathcal{C}$. The gap of $\lambda(\mathcal{C})$ with respect to $\underline{\lambda}$ is computed as $\frac{\lambda(\mathcal{C})-\underline{\lambda}}{\underline{\lambda}}$. Such a gap, named $GAP(\mathcal{C})$, is used to evaluate set $\mathcal{C}$. If $GAP(\mathcal{C})$ is more than a fixed $\overline{GAP}$, then a tabu constraint is added to the mathematical program used to solve the CCLP. The tabu constraint is implemented by imposing that all the binary variables, set to 1 only for the required elements belonging to $\mathcal{C}$, flip their value from 1 to 0. If $GAP(\mathcal{C})$ is less than or equal to $\overline{GAP}$, then a diversification constraint is added to the mathematical program. The diversification constraint ensures that at least one of the binary variables previously defined flips its value from 1 to 0 or from 0 to 1. The iterative diversification process ends whenever the model becomes infeasible due to the added tabu and diversification constraints, or a given number of iterations is reached. We set $\overline{GAP} = 0.05$ in our computational experiments. Finally, a last attempt to obtain a feasible MCGRP solution of minimum cost is made by solving a set partitioning model, in which the involved routes are all the different routes associated with the feasible MCGRP solutions found during the iterative algorithm.

3.3. Relaxed Constraints, Valid Inequalities and Separation Routines

Separations routines are used in both the B&C algorithm for solving the MCGRP and the one-index formulation (5).

Specifically, connectivity constraints (1f) can be separated by adapting the exact and heuristic procedures used in Bosco et al. (2013). The separation of odd-cut inequalities (5b) is straightforward and is already described in Padberg and Rao (1982). Arcs are handled as edges and the polynomial odd minimum cut algorithm is applied. To separate capacity inequalities (5c) two methods are known from the literature. A heuristic method was presented by Belenguer and Benavent (2003) and an exact method by Ahr (2004). Both methods are used for the one-index formulation (5), while in the second phase capacity inequalities (5c) are identified solely using the heuristic method. Belenguer and Benavent (2003) also presented disjoint-path inequalities as additional valid inequalities. Separation routines for these inequalities are adapted from their paper.

The separation of balance inequalities is also known from the literature (e.g. Benavent et al., 2000), but more intricate to implement for the MCGRP. We will give a short description of the procedure. The balanced set conditions (1e) can be rewritten as

$$y^k(\delta(S)) + x^k(\delta_R(S)) \quad + y^k(\delta^+(S)) + x^k(\delta_R^+(S)) \quad - y^k(\delta^-(S)) - x^k(\delta_R^-(S)) \geq 0, \qquad \forall S \subseteq V,\ k \in K$$

and in (5) as

$$y(\delta(S)) + |\delta_R(S)| \quad + y(\delta^+(S)) + |\delta_R^+(S)| \quad - y(\delta^-(S)) - |\delta_R^-(S)| \geq 0, \qquad \forall S \subseteq V.$$

In order to separate violated inequalities, the algorithm of Nobert and Picard (1996) can be adapted. Such a procedure is also described by Benavent et al. (2000) for an uncapacitated problem, in which all links are required. We define, for a solution $(\hat{x}_\ell^k, \hat{y}_\ell^k)$ to (1) concerning the k-th vehicle, or a solution $(\hat{y}_\ell)$ to (5)

$$w_\ell^k = \begin{cases} \hat{y}_\ell^k + \hat{x}_\ell^k, & \ell \in E_R \cup A_R \\ \hat{y}_\ell^k, & \ell \in (E \cup A) \setminus (E_R \cup A_R) \end{cases} \qquad \text{and} \qquad w_\ell = \begin{cases} \hat{y}_\ell + 1, & \ell \in E_R \cup A_R \\ \hat{y}_\ell, & \ell \in (E \cup A) \setminus (E_R \cup A_R), \end{cases}$$

respectively. For any link subset L, we define $w(L) = \sum_{\ell \in L} w_\ell^k$ (alternatively, $w(L) = \sum_{\ell \in L} w_\ell$). Then, for any vertex subset S, we can determine $f(S) = w(\delta(S)) + w(\delta^+(S)) - w(\delta^-(S))$. A set S for which $f(S)$ is minimum is the most unbalanced set, and $f(S) < 0$ identifies a violated balanced set condition. The point is that the algorithm of Nobert and Picard (1996) finds such a set S by solving a maximum-flow problem over another support graph with two extra vertices, even if some of the values w_ℓ^k and w_ℓ are negative.

3.4. Initial Relaxation and Cut Pool Management

The initial relaxation of the MCGRP, at the root node of the B&C tree, is a linear program which includes the following components: the objective function (1a), all constraints (1b)-(1c), the balanced set conditions (1e) associated with the unbalanced vertices i, for which $|\delta(i)| < |\delta^-(i)| - |\delta^+(i)|$ holds. Moreover, it includes the connectivity constraints (1f) associated with the R-sets, i.e., the connected components of the graph induced by all required elements.

The initial relaxation also contains inequalities (2) associated with each vertex i for which $|\delta_R^*(i)|$ is odd fixing $H = \delta_R^*(i)$, symmetry breaking constraints (3a)-(3c), and the valid inequalities generated by the lower bounding procedure.

An iteration of the B&C algorithm involves the selection of a subproblem from the list of active subproblems and the addition of violated constraints and valid inequalities to this subproblem. The set containing violated constraints and valid inequalities is called *cut pool*. In our implementation, the cut pool is cleaned every 50 iterations by eliminating non-binding inequalities, i.e., those with slack greater than ϵ or dual variables less than ϵ, where $\epsilon = 10^{-6}$ is the tolerance. Note that the cuts generated in the first phase are eliminated in the second phase whenever they become non-binding inequalities.

3.5. Branching on Vertex Degrees

Consider a solution which does not contain fractional variables, where $(\hat{x}_\ell^k, \hat{y}_\ell^k)$ refers to the k-th vehicle. In this case, the standard branching on fractional variables is not activated. If, however, the solution is not feasible for the MCGRP, then there exists at least a vertex with odd degree. Let $d_i^k = \hat{x}^k(\delta_R^*(i)) + \hat{y}^k(\delta^*(i))$ be the degree of the odd vertex i with respect to the k-th route. Two branches $x^k(\delta_R^*(i)) + y^k(\delta^*(i)) \leq 2p$ and $x^k(\delta_R^*(i)) + y^k(\delta^*(i)) \geq 2p + 2$ are created with $p \in \mathbb{Z}_+$ defined by $2p < d_i^k < 2p + 2$.

The specific variable to branch on is determined as follows. For branching on vertex degrees, we first compute for each vertex $i \in V$ and for each vehicle $k \in K$ the distance of d_i^k to the next even integer, i.e., $min\{2p + 2 - d_i^k, d_i^k - 2p\}$ for an integer p with $2p \leq d_i^k < 2p + 2$. We select the vertex i^* for which

$$\frac{\min\{2p + 2 - d_{i^*}^k, d_{i^*}^k - 2p\}}{\alpha + \beta 2p}$$

is maximal, where we use $\alpha = 6$ and $\beta = 1$ as suggested by Bode and Irnich (2012).

3.6. Outline of the Solution Algorithm

An outline of our solution approach is provided in the following. The management of the cut pool is omitted in order to simplify the scheme:

Step 1. Call the lower bounding procedure to compute $\underline{\lambda}$ and cuts used in (5).

Step 2. Call the CCLP heuristic to compute an upper bound $\bar{\lambda}$ and an initial solution.

Step 3. If $\bar{\lambda} = \underline{\lambda}$, STOP.

Step 4. Define a relaxed MCGRP formulation as described in Section 3.4 and insert the resulting subproblem in a list Θ.

Step 5. If Θ is empty or $\bar{\lambda} = \underline{\lambda}$, STOP. Otherwise extract a subproblem P from Θ.

Step 6. Solve the subproblem P. Let λ_P be the solution value. If $\lambda_P \geq \bar{\lambda}$, go back to Step 5.

Step 7. Separate violated constraints (1f). If the heuristic algorithm fails, apply the exact separation algorithm.

Step 8. Separate violated inequalities (1e).

Step 9. Separate violated inequalities (5b).

Step 10. Separate violated inequalities (5c).

Step 11. If some violated inequalities have been identified in Steps 7, 8, 9 and 10, add these inequalities to the cut pool and go back to Step 6.

Otherwise, if the current solution is feasible, set $\bar{\lambda} = \lambda_P$ and go back to Step 5.

Step 12. If the current solution is not integer, generate two subproblems by branching on a fractional variable. Otherwise, generate two subproblems by branching on vertex degrees as described in Section 3.5.

Insert the subproblems in Θ and go back to Step 5.

4. Computational Results

The instances used in the computational experiments stem from the paper by Bosco et al. (2013). These instances were derived from the `gdb` instances introduced by Golden et al. (1983) for the undirected CARP and from the `mval` instances provided by Belenguer et al. (2006) for the mixed CARP. For each original instance, six new instances were generated using an additional parameter β in the set $\{0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$, where β controls the number of required links whose demand is shifted to adjacent vertices. The new instances were named `mggdb` and `mgval` and additional details about their characteristics are discussed in Bosco et al. (2013). For the sake of brevity, we limited the investigation to instances with $\beta \in \{0.25, 0.30, 0.35\}$. For the other `mggdb` and `mgval` datasets the performance should remain rather similar (see Bosco et al., 2013). On the contrary, we consider all `mggdb` and `mgval` instances including those for which the number of vehicles m exceeds 7.

In Tables 1–6, the columns named "CG" report the results obtained by a column generation (CG) method (Gaze, 2013; Gaze et al., 2013); the columns named "B&C&P" report the results obtained by the branch-and-cut-and-price (B&C&P) algorithm of Bach (2014); the columns named "B&C" report the results obtained by the B&C algorithm of Bosco et al. (2013); the columns named "New.B&C" report the results obtained by our B&C algorithm. Other column headings are defined as follows:

FILE	instance name
m	number of vehicles
η	number $\|V_R \cup E_R \cup A_R\|$ of required elements
$\underline{\lambda}$	lower bound
$\bar{\lambda}^H$	initial upper bound provided by the CCLP heuristic
$\bar{\lambda}$	best solution value reached within a time limit
	(an optimal value certified by the algorithm is marked with an asterisk)
GAP	percentage gap
CON, CAP, ODD, BAL	number of connectivity, capacity, odd-cut, balanced set inequalities
	(inequalities generated in the second phase)
NOD	number of nodes from the search tree
SEC^1	computation time in seconds for the first phase
SEC^2	computation time in seconds for the CCLP heuristic
SEC	total computation time in seconds
	(the time limit corresponding to 6 hours is marked with TL)

All experiments to obtain New.B&C results were carried out on a PC equipped with 2 Intel Xeon Quad Core CPUs @3.0 GHz, with 6 GB RAM, i.e., the same PC used to obtain B&C results. In both cases, ILOG CPLEX library, release 12.2, was used and all standard CPLEX cuts were activated. Computational experiments to obtain CG results were carried out on a PC equipped with an Intel(R) Core(TM) i7 CPU @2.93 GHz, with 8 GB RAM. Finally, computational experiments to obtain B&C&P results were carried out on a HP EliteBook with an Intel Core 2 Duo CPU P8700 @2.53GHz and 4 GB RAM.

For B&C results, we just report times, objective function values and percentage gaps; the lower bounds are not reported because generally poor (outcomes at the root node of the search tree). For CG and B&C&P results, we just report lower bounds, objective function values and percentage gaps; the times are not explicitly reported but all the results were obtained within a time limit of 1 hour (2 hours for `mgval` instances) for CG results and 3 hours for B&C&P results. Note that, for the B&C&P algorithm, the percentage gaps were calculated as $100\frac{(\bar{\lambda}-\underline{\lambda})}{(\bar{\lambda}+\underline{\lambda})/2}$. With regard to the results of the B&C algorithm of Bosco et al. (2013), the CG method, and the B&C&P algorithm, we use "–" to indicate that no value is available in the literature. With regard to our algorithm, we use "–" in the column $\bar{\lambda}^H$ when the upper bounding procedure failed.

The tables show that the possibility of finding the optimal solution for our B&C algorithm generally decreases with the increase of the number of vehicles m. Frequently the value provided by our lower bounding procedure is excellent. 62 out of 69 instances of the `mggdb` benchmark set are solved to optimality by our algorithm (21 for $\beta = 0.25$, 21 for $\beta = 0.30$, and 20 for $\beta = 0.35$). In particular, all `mggdb` instances studied in Bosco et al. (2013), i.e., those with up to 7 vehicles, with β in $\{0.25, 0.30, 0.35\}$, are solved to optimality by our algorithm. Moreover, we solve to optimality all `mggdb` instances with 8 vehicles. For the group of instances with 10 vehicles, we can compare our results with the CG method and the B&C&P algorithm. Sometimes the lower bound value or the upper bound value provided by CG method are better than the values provided by our algorithm. Anyway, for `mggdb` instances with 10 vehicles, the maximum percentage gap of our algorithm is approx. 14% against the value of approx. 24% provided by the CG method. For the same instances, the best performance is obtained by the B&C&P algorithm. Nevertheless, our algorithm solves to optimality more `mggdb` instances than the B&C&P algorithm.

The number of `mgval` instances solved to optimality by our algorithm is equal to 62 out of 102 (24 with $\beta = 0.25$, 19 with $\beta = 0.30$, and 19 with $\beta = 0.35$). These results confirm that the `mgval` instances are harder to solve due to the structure of their graphs. In this case, some instances reported in Bosco et al. (2013), with β in $\{0.25, 0.30, 0.35\}$, are not solved to optimality by our algorithm. Anyway, upper bound $\bar{\lambda}$ reported in Bosco et al. (2013) is never better than upper bound provided by our algorithm. Moreover, for our algorithm the maximum percentage gap is equal to 2.61% against the value of 12.08 % provided by the algorithm of Bosco et al. (2013). The CG method ran just on six instances of `mgval` dataset considered in this paper. For $m \leq 7$, its maximum percentage gap is 25.24%. No comparison is possible for $m > 7$.

Table 1: Computational results for `mggdb` dataset with $\beta = 0.25$

			CG			B&C&P			B&C			New.B&C											
FILE	m	η	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\bar{\lambda}$	GAP	SEC	$\underline{\lambda}$	$\bar{\lambda}^H$	CON	CAP	ODD	BAL	NOD	$\bar{\lambda}$	GAP	SEC[1]	SEC[2]	SEC
`mggdb19`	3	10	53	53*	0.00	53	53*	0.00	53*	0.00	1.14	53	53	0	0	0	0	0	53*	0.00	0.61	0.74	1.35
`mggdb4`	4	18	273	292	6.51	289	289*	0.00	289*	0.00	22.68	286	332	807	96	29	363	48	289*	0.00	2.15	2.24	6.09
`mggdb10`	4	22	265	276	3.99	265	265*	0.00	265*	0.00	2.79	265	277	593	219	79	436	1	265*	0.00	0.39	1.08	2.68
`mggdb15`	4	20	54	56	3.57	55	55*	0.00	55*	0.00	0.47	55	57	10	19	1	195	2	55*	0.00	0.44	0.49	1.20
`mggdb20`	4	20	116	122	4.92	116	116*	0.00	116*	0.00	7.80	116	122	559	78	5	383	162	116*	0.00	0.37	1.74	11.71
`mggdb1`	5	21	274	285	3.86	280	280*	0.00	280*	0.00	2079.41	277	307	1386	332	62	608	709	280*	0.00	2.34	2.72	30.43
`mggdb3`	5	22	270	282	4.26	278	278*	0.00	278*	0.00	923.88	275	291	1589	187	70	575	215	278*	0.00	2.79	2.84	23.50
`mggdb6`	5	21	291	295	1.36	292	292*	0.00	292*	0.00	14.89	292	312	1083	129	28	379	1	292*	0.00	0.62	1.37	2.70
`mggdb7`	5	20	288	290	0.69	290	290*	0.00	290*	0.00	40.35	290	310	666	169	33	401	1	290*	0.00	1.85	1.48	3.92
`mggdb11`	5	41	347	366	5.19	349	356	1.99	356	3.09	TL	356	395	11737	666	324	2115	27	356*	0.00	1.31	10.87	230.57
`mggdb14`	5	20	107	107*	0.00	107	107*	0.00	107*	0.00	1.43	107	109	389	38	24	356	24	107*	0.00	0.40	1.19	3.06
`mggdb16`	5	25	97	169	42.60	98	98*	0.00	98*	0.00	4.99	98	103	709	143	28	545	28	98*	0.00	0.57	2.87	7.33
`mggdb17`	5	25	70	78	10.26	71	71*	0.00	71*	0.00	1.02	71	71	0	0	0	0	0	71*	0.00	0.48	1.35	1.83
`mggdb18`	5	32	139	157	11.46	140	144	2.82	144	3.47	TL	144	156	2871	217	109	951	1	144*	0.00	0.67	3.53	6.08
`mggdb2`	6	25	339	359	5.57	346	352	1.72	349	5.58	TL	349	380	2092	363	96	907	14	349*	0.00	1.68	2.55	15.90
`mggdb5`	6	24	384	404	4.95	394	394*	0.00	394	6.41	TL	393	439	2566	403	86	1167	245	394*	0.00	2.07	3.42	62.51
`mggdb13`	6	26	388	392	1.02	388	388*	0.00	388	3.38	TL	376	418	2075	254	61	996	64984	388*	0.00	0.51	6.27	5716.61
`mggdb21`	6	31	144	161	10.56	146	146*	0.00	146	0.68	TL	146	158	1213	177	72	915	57	146*	0.00	0.76	4.97	27.92
`mggdb12`	7	22	456	459	0.65	459	459*	0.00	459	3.74	TL	439	502	2355	313	114	1353	5982	459*	0.00	1.92	3.72	616.04
`mggdb22`	8	38	160	172	6.98	160	160*	0.00	–	–	–	160	168	3127	354	154	1764	461	160*	0.00	0.88	12.47	282.59
`mggdb8`	10	45	328	356	7.87	333	336	0.90	–	–	–	332	358	19792	7233	753	10120	4438	358	7.26	16.98	15.53	TL
`mggdb9`	10	47	305	331	7.85	308	309	0.32	–	–	–	300	349	35794	3756	950	9824	1244	349	14.04	14.63	20.92	TL
`mggdb23`	10	48	181	234	22.65	181	181*	0.00	–	–	–	181	196	4573	546	195	3026	7608	181*	0.00	1.79	28.84	19325.56
Number of optima			2			18			12			21											

No instance of `mgval` dataset with β in $\{0.25, 0.30, 0.35\}$ has been tackled by the B&C&P algorithm. This algorithm solely ran on the instances of `mgval` dataset with $\beta = 0.50$. Nevertheless, for most instances of this dataset, the B&C&P algorithm obtained no upper bound $\bar{\lambda}$ within a time limit of 6 hours.

5. Conclusions

In this paper, we proposed a new formulation for the MCGRP and a two-phase B&C algorithm to exactly solve it. The approach benefits from the typically very tight lower bounds computed fast in the first phase, in which an aggregated one-index formulation is solved. The performance of the overall B&C proposed in this paper has been evaluated by carrying out computational experiments on two benchmark sets: As a result, for all `mggdb` instances of Bosco et al. (2013), with β in $\{0.25, 0.30, 0.35\}$ and $m \leq 7$, we know optimal solutions now. Optimality was proved and lower bounds were improved for many other instances of this and the second `mgval` benchmark set. We also studied for the first time a group of larger instances: Although we never proved optimality in these cases, the remaining gaps provided by our algorithm remain below a threshold of 20%. We suspect that our bounds unevenly contribute to these gaps and that lower bounds are tighter than upper bounds.

References

Y. Adulyasak, J.-F. Cordeau, and Jans R. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26:103–120, 2014.

D. Ahr. *Contributions to Multiple Postmen Problems.* PhD thesis, Ruprecht-Karls-Universität Heidelberg, Germany, 2004.

L. Bach. *Routing and Scheduling Problems – Optimization using Exact and Heuristic Methods.* Ph.D. dissertation, School of Business and Social Sciences, Aarhus University, Denmark, 2014.

L. Bach, G. Hasle, and S. Wøhlk. A lower bound for the node, edge, and arc routing problem. *Computers and Operations Research*, 40:943–952, 2013.

J.M. Belenguer and E. Benavent. The capacitated arc routing problem: Valid inequalities and facets. *Computational Optimization and Applications*, 10:165–187, 1998.

J.M. Belenguer and E. Benavent. A cutting plane algorithm for the capacitated arc routing problem. *Computers and Operations Research*, 30:705–728, 2003.

Table 2: Computational results for `mgval` dataset with $\beta = 0.25$

			CG			B&C&P			B&C			New.B&C												
FILE	m	η	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\bar{\lambda}$	GAP	SEC	$\underline{\lambda}$	$\bar{\lambda}^H$	CON	CAP	ODD	BAL	NOD	$\bar{\lambda}$	GAP	SEC[1]	SEC[2]	SEC
mgval1A	2	54	157	210	25.24	–	–	–	177*	0.00	0.30	177	183	40585	335	255	873	3	177*	0.00	0.58	8.13	17.22
mgval2A	2	40	–	–	–	–	–	–	259*	0.00	6.10	259	273	34016	655	66	502	1	259*	0.00	1.29	9.75	18.15
mgval3A	2	44	–	–	–	–	–	–	89*	0.00	3.21	89	94	40428	323	98	539	1	89*	0.00	0.95	7.12	16.04
mgval1B	3	47	–	–	–	–	–	–	217*	0.00	0.13	217	227	21861	1158	161	1450	55	217*	0.00	0.55	9.05	41.93
mgval2B	3	48	–	–	–	–	–	–	336*	0.00	941.47	336	360	72095	1264	164	1277	1	336*	0.00	1.59	17.68	53.04
mgval3B	3	41	–	–	–	–	–	–	125*	0.00	7217.32	125	125	0	0	0	0	0	125*	0.00	1.85	15.81	17.66
mgval4A	3	89	–	–	–	–	–	–	514	1.55	TL	514	542	508752	4090	737	4930	2	514*	0.00	5.03	85.67	582.51
mgval5A	3	92	–	–	–	–	–	–	485*	0.00	2418.17	485	520	371080	2581	442	3093	4	485*	0.00	1.75	77.88	482.27
mgval6A	3	67	–	–	–	–	–	–	274*	0.00	11.81	274	278	66638	696	302	1875	2	274*	0.00	0.69	11.92	47.86
mgval7A	3	84	–	–	–	–	–	–	297*	0.00	121.29	297	314	30003	605	279	1926	17	297*	0.00	0.89	26.28	407.94
mgval8A	3	88	–	–	–	–	–	–	510	0.39	TL	510	538	627092	1608	425	3198	13	510*	0.00	0.81	18.30	384.06
mgval9A	3	122	–	–	–	–	–	–	371	1.08	TL	371	391	681073	2069	1005	5921	1	371*	0.00	1.20	109.32	669.04
mgval10A	3	129	–	–	–	–	–	–	492*	0.00	5.08	492	536	2132150	7443	1641	10441	6	492*	0.00	1.48	285.36	8200.45
mgval4B	4	96	–	–	–	–	–	–	537	4.74	TL	537	583	809353	6003	798	6204	2	537*	0.00	2.43	190.11	1328.10
mgval5B	4	86	–	–	–	–	–	–	493	4.26	TL	493	546	305773	2670	763	3507	2	493*	0.00	3.34	79.40	918.19
mgval6B	4	63	–	–	–	–	–	–	263	1.93	TL	257	305	26858	9334	205	2331	10748	263*	0.00	0.73	40.20	11664.95
mgval7B	4	85	–	–	–	–	–	–	355*	0.00	1859.86	355	367	159024	1384	474	3144	104	355*	0.00	1.90	28.13	2153.19
mgval8B	4	84	–	–	–	–	–	–	423	4.26	TL	423	481	282147	3708	766	4052	36	423*	0.00	2.27	34.11	1910.60
mgval9B	4	112	–	–	–	–	–	–	358	1.12	TL	358	390	344109	5369	826	8672	181	358*	0.00	2.36	121.53	20123.59
mgval10B	4	123	–	–	–	–	–	–	528*	0.00	1.64	528	570	722454	2354	427	4445	569	528*	0.00	1.81	410.34	19187.63
mgval4C	5	100	–	–	–	–	–	–	525	4.00	TL	525	620	706598	7107	1218	7485	4	525*	0.00	4.85	150.44	14925.20
mgval5C	5	93	–	–	–	–	–	–	584	3.94	TL	584	676	313010	3562	790	4068	2	584*	0.00	3.83	76.29	722.30
mgval9C	5	119	–	–	–	–	–	–	365	4.93	TL	361	365	733	0	0	180	1075	365	1.10	5.91	82.33	TL
mgval10C	5	125	–	–	–	–	–	–	483	0.62	TL	483	534	1104758	12277	2445	18675	427	483*	0.00	3.06	191.20	20129.18
mgval3C	7	41	146	154	5.19	–	–	–	153	11.01	TL	153	161	11741	2965	429	2855	2	153*	0.00	3.78	12.42	106.12
mgval1C	8	51	–	–	–	–	–	–	–	–	–	278	323	29116	2986	490	5418	4561	323	13.93	7.69	147.60	TL
mgval2C	8	48	–	–	–	–	–	–	–	–	–	479	512	45494	3636	541	5785	401	512	6.45	7.55	53.31	TL
mgval4D	9	96	–	–	–	–	–	–	–	–	–	675	778	102380	10161	1959	12048	1	778	13.24	18.54	116.88	TL
mgval5D	9	85	–	–	–	–	–	–	–	–	–	635	741	339945	12991	3736	18459	1	741	14.30	9.26	169.44	TL
mgval7C	9	85	–	–	–	–	–	–	–	–	–	374	437	154757	5129	1311	10841	1	437	14.42	16.26	240.32	TL
mgval8C	9	78	–	–	–	–	–	–	–	–	–	538	625	150870	9628	1676	13664	1	625	13.92	10.08	154.68	TL
mgval6C	10	66	–	–	–	–	–	–	–	–	–	316	378	40618	6154	1187	9566	2	378	16.40	13.19	65.49	TL
mgval9D	10	121	–	–	–	–	–	–	–	–	–	418	498	132184	5428	1288	7232	1	498	16.06	39.75	201.83	TL
mgval10D	10	119	–	–	–	–	–	–	–	–	–	565.5	655	381573	10886	2459	16002	1	655	13.59	34.52	108.40	TL
Number of optima				0			–		12										24				

Table 3: Computational results for mggdb dataset with $\beta = 0.30$

			CG			B&C&P			B&C			New.B&C										
FILE	m	η	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\bar{\lambda}$	GAP	SEC	$\underline{\lambda}$	$\bar{\lambda}^H$	CON	CAP	ODD	BAL	NOD	$\bar{\lambda}$	GAP	SEC[1]	SEC[2]	SEC
mggdb19	3	10	51	51^*	0.00	51	51^*	0.00	51^*	0.00	0.36	51	51	0	0	0	0	0	51^*	0.00	0.41	0.57	0.98
mggdb4	4	18	253	271	6.64	260	260^*	0.00	260^*	0.00	39.70	260	269	952	65	24	256	13	260^*	0.00	0.89	0.74	2.72
mggdb10	4	22	241	248	2.82	242	242^*	0.00	242^*	0.00	12.06	242	264	1293	263	41	519	60	242^*	0.00	1.37	1.29	8.94
mggdb15	4	19	43	44	2.27	44	44^*	0.00	44^*	0.00	0.48	44	44	0	0	0	0	0	44^*	0.00	0.32	0.50	0.82
mggdb20	4	18	94	100	6.00	94	94^*	0.00	94^*	0.00	27.27	93	100	631	151	21	345	398	94^*	0.00	0.66	1.51	24.34
mggdb1	5	21	262	276	5.07	273	273^*	0.00	273^*	0.00	4279.45	271	285	2099	850	67	689	2253	273^*	0.00	2.72	2.51	77.91
mggdb3	5	19	269	287	6.27	270	270^*	0.00	270^*	0.00	4447.90	267	298	979	172	46	622	363	270^*	0.00	1.67	1.05	17.17
mggdb6	5	22	265	283	6.36	276	276^*	0.00	276^*	0.00	443.39	276	286	2077	372	58	651	1101	276^*	0.00	1.35	1.49	56.81
mggdb7	5	20	258	281	8.19	273	273^*	0.00	273^*	0.00	2870.34	272	280	1670	194	37	617	14223	273^*	0.00	1.03	1.26	810.14
mggdb11	5	43	382	430	11.16	382	–	–	387	1.55	TL	387	–	13686	1154	481	2776	85	387^*	0.00	1.53	8.71	250.78
mggdb14	5	17	100	104	3.85	101	101^*	0.00	101^*	0.00	184.67	101	104	350	36	12	329	14	101^*	0.00	0.58	1.06	2.48
mggdb16	5	24	105	169	37.87	105	105^*	0.00	105^*	0.00	2.77	105	113	1532	253	43	601	315	105^*	0.00	0.64	2.05	30.55
mggdb17	5	22	65	75	13.33	65	65^*	0.00	65^*	0.00	0.64	65	67	73	39	1	286	2	65^*	0.00	0.39	0.74	1.42
mggdb18	5	30	142	153	7.19	144	144^*	0.00	144^*	0.00	1.59	144	157	1390	134	63	904	6	144^*	0.00	0.75	3.75	7.04
mggdb2	6	24	294	307	4.23	301	301^*	0.00	301	5.74	TL	300	324	5460	4264	61	1161	90859	301^*	0.00	1.28	2.65	11631.46
mggdb5	6	25	384	416	7.69	388	388^*	0.00	388	2.82	TL	388	412	1824	380	109	1103	256	388^*	0.00	0.85	3.42	37.69
mggdb13	6	24	479	529	9.45	483	483^*	0.00	486	8.02	TL	463	554	3663	11941	40	1233	347098	483^*	0.00	0.50	6.68	20676.03
mggdb21	6	28	121	121^*	0.00	121	121^*	0.00	121	0.83	TL	121	–	405	39	5	374	10	121^*	0.00	0.58	4.43	24.93
mggdb12	7	21	462	472	2.12	467	467^*	0.00	467	6.36	TL	452	490	3262	2928	127	1376	17080	467^*	0.00	1.31	1.83	1586.05
mggdb22	8	37	151	156	3.21	153	153^*	0.00	–	–	–	153	159	4771	325	171	2132	44281	153^*	0.00	1.42	6.12	13595.79
mggdb8	10	46	329	352	6.53	328	331	0.91	–	–	–	324	364	28144	2935	731	8445	6272	364	10.99	20.31	10.35	TL
mggdb9	10	46	278	290	4.14	281	281^*	0.00	–	–	–	273	317	22611	5213	814	9263	668	317	13.88	18.74	24.06	TL
mggdb23	10	47	167	188	11.17	167	167^*	0.00	–	–	–	167	189	6843	51390	316	4134	100200	167^*	0.00	1.77	10.74	20184.59
Number of optima			2			21			13			21											

J.M. Belenguer, E. Benavent, P. Lacomme, and C. Prins. Lower and upper bounds for the mixed capacitated arc routing problem. *Computers and Operations Research*, 33:3363–3383, 2006.

E. Benavent, A. Corberán, and J.M. Sanchis. Linear programming based methods for solving arc routing problems. In M. Dror, editor, *Arc routing: Theory, solutions and applications*, pages 231–275. Kluwer Academic Publishers, 2000.

M. Blais and G. Laporte. Exact solution of the generalized routing problem through graph transformations. *Journal of the Operational Research Society*, 54:906–910, 2003.

C. Bode and S. Irnich. Cut-first branch-and-price-second for the capacitated arc-routing problem. *Operations Research*, 60: 1167–1182, 2012.

A. Bosco, D. Laganà, R. Musmanno, and F. Vocaturo. Modeling and solving the mixed capacitated general routing problem. *Optimization Letters*, 7:1451–1469, 2013.

A. Bosco, D. Laganà, R. Musmanno, and F. Vocaturo. A matheuristic algorithm for the mixed capacitated general routing problem. *Submitted paper*, 2014.

J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. *Operations Research*, 43:649–660, 1995.

O. Bräysy, E. Martínez, Y. Nagata, and D. Soler. The mixed capacitated general routing problem with turn penalties. *Expert Systems with Applications*, 38:12954–12966, 2011.

A. Corberán and G. Laporte, editors. *Arc Routing: Problems, Methods and Applications.* MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2014.

A. Corberán and J.M. Sanchis. The general routing problem polyhedron: Facets from the rpp and gtsp polyhedra. *European Journal of Operational Research*, 108:538–550, 1998.

A. Corberán, A.N. Letchford, and J.M. Sanchis. A cutting plane algorithm for the general routing problem. *Mathematical Programming, Ser. A*, 90:291–316, 2001.

A. Corberán, A. Romero, and J.M. Sanchis. The mixed general routing polyhedron. *Mathematical Programming, Ser. A*, 96: 103–137, 2003.

A. Corberán, G. Mejía, and J.M. Sanchis. New results on the mixed general routing problem. *Operations Research*, 53:363–376, 2005.

A. Corberán, I. Plana, and J.M. Sanchis. A branch & cut algorithm for the windy general routing problem and special cases. *Networks*, 49:245–257, 2007.

A. Corberán, I. Plana, and J.M. Sanchis. The windy general routing polyhedron: A global view of many known arc routing polyhedra. *SIAM Journal on Discrete Mathematics*, 22:606–628, 2008.

I. Correia, L. Gouveia, and F. Saldanha-da Gama. Discretized formulations for capacitated location problems with modular distribution costs. *European Journal of Operational Research*, 204:237–244, 2010.

M. Dror, editor. *Arc Routing: Theory, Solutions and Applications.* Kluwer, Boston, 2000.

K.A. Gaze. Exact optimization methods for the mixed capacitated general routing problem. Master's thesis, Norwegian University of Science and Technology, 2013.

K.A. Gaze, G. Hasle, and C. Mannino. Column generation for the mixed capacitated general routing. 2013. Presented at

Table 4: Computational results for mgval dataset with $\beta = 0.30$

			CG			B&C&P			B&C			New.B&C											
FILE	m	η	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\bar{\lambda}$	GAP	SEC	$\underline{\lambda}$	$\bar{\lambda}^H$	CON	CAP	ODD	BAL	NOD	$\bar{\lambda}$	GAP	SEC^1	SEC^2	SEC
mgval1A	2	53	159	196	18.88	–	–	–	170*	0.00	4.38	170	170	0	0	0	0	0	170*	0.00	0.56	14.41	14.97
mgval2A	2	42	–	–	–	–	–	–	233*	0.00	1.58	233	243	54705	736	103	747	6	233*	0.00	0.47	7.64	33.46
mgval3A	2	46	–	–	–	–	–	–	105*	0.00	6.34	105	106	61430	858	184	1563	68	105*	0.00	0.92	8.00	73.88
mgval1B	3	47	–	–	–	–	–	–	194*	0.00	24.46	194	204	20288	1078	147	1174	135	194*	0.00	1.54	11.40	146.63
mgval2B	3	49	–	–	–	–	–	–	347*	0.00	3309.99	347	376	32314	706	71	756	150	347*	0.00	1.01	13.14	140.70
mgval3B	3	41	–	–	–	–	–	–	115*	0.00	31.27	115	131	41711	1386	389	1637	1	115*	0.00	1.76	8.15	51.05
mgval4A	3	87	–	–	–	–	–	–	477	1.75	TL	477	506	171935	3072	542	3294	347	477*	0.00	4.93	40.91	2098.60
mgval5A	3	86	–	–	–	–	–	–	445*	0.00	96.32	445	477	301480	2499	514	3201	1	445*	0.00	1.73	60.37	509.47
mgval6A	3	66	–	–	–	–	–	–	252	1.84	TL	252	258	79401	1212	458	2580	4	252*	0.00	2.42	9.88	139.83
mgval7A	3	77	–	–	–	–	–	–	324*	0.00	1.72	324	339	66313	1035	787	5080	13	324*	0.00	0.89	69.36	3197.46
mgval8A	3	88	–	–	–	–	–	–	431	0.81	TL	431	465	378638	1881	485	3054	26	431*	0.00	0.78	290.39	1108.07
mgval9A	3	118	–	–	–	–	–	–	357	0.28	TL	357	375	318780	1088	300	3607	3	357*	0.00	1.29	82.37	4549.30
mgval10A	3	127	–	–	–	–	–	–	484	0.62	TL	484	506	2430187	6405	1227	9470	2	484*	0.00	2.69	104.87	1986.67
mgval4B	4	98	–	–	–	–	–	–	533	5.19	TL	531	533	935	19	0	111	3976	533	0.38	4.06	100.37	TL
mgval5B	4	83	–	–	–	–	–	–	490	4.58	TL	484	490	1165	34	0	320	9764	490	1.22	4.21	80.31	TL
mgval6B	4	64	–	–	–	–	–	–	262*	0.00	6331.62	262	278	89991	2521	384	3758	1302	262*	0.00	1.30	24.84	1425.85
mgval7B	4	82	–	–	–	–	–	–	344	1.91	TL	344	354	125319	1456	464	3175	1	344*	0.00	2.53	65.55	632.25
mgval8B	4	83	–	–	–	–	–	–	400	2.35	TL	400	470	115184	4560	529	4563	778	400*	0.00	2.17	37.73	9578.23
mgval9B	4	110	–	–	–	–	–	–	348	1.15	TL	348	380	563806	12495	2372	13837	223	348*	0.00	2.65	102.65	12667.79
mgval10B	4	123	–	–	–	–	–	–	441	1.36	TL	441	491	1065874	4932	1121	6860	255	441*	0.00	3.32	85.53	14180.09
mgval4C	5	98	–	–	–	–	–	–	498	5.92	TL	492	498	398	9	0	106	2354	498	1.20	6.70	145.53	TL
mgval5C	5	87	–	–	–	–	–	–	551	6.80	TL	549	551	1532	1	0	187	2345	551	0.36	3.60	85.53	TL
mgval9C	5	112	–	–	–	–	–	–	335	1.49	TL	335	378	1080608	12121	2355	16082	50	335*	0.00	1.74	117.24	20185.91
mgval10C	5	125	–	–	–	–	–	–	478	2.93	TL	475	527	961174	14481	2408	19960	2301	476	0.21	4.55	156.85	TL
mgval3C	7	41	146	165	11.52	–	–	–	153	12.08	TL	149	153	223	12	0	239	5192	153	2.61	3.06	20.52	TL
mgval1C	8	48	–	–	–	–	–	–	–	–	–	255	310	19446	2451	575	5362	5045	310	17.74	4.06	125.91	TL
mgval2C	8	45	–	–	–	–	–	–	–	–	–	489	534	45832	3956	438	7535	5041	534	8.43	6.83	30.08	TL
mgval4D	9	94	–	–	–	–	–	–	–	–	–	652	765	107573	9627	1548	11953	1	765	14.77	21.27	149.42	TL
mgval5D	9	86	–	–	–	–	–	–	–	–	–	612	736	292451	11964	4020	17484	1	736	16.85	10.49	208.73	TL
mgval7C	9	85	–	–	–	–	–	–	–	–	–	347	388	74456	1609	397	6267	1	388	10.57	11.65	49.64	TL
mgval8C	9	75	–	–	–	–	–	–	–	–	–	510	590	131217	10126	1946	12947	2	590	13.56	10.51	84.55	TL
mgval6C	10	64	–	–	–	–	–	–	–	–	–	307	364	37530	6875	977	10187	1834	364	15.66	16.08	30.81	TL
mgval9D	10	122	–	–	–	–	–	–	–	–	–	421.13	501	248834	9511	1860	15279	1	501	15.77	49.43	100.60	TL
mgval10D	10	121	–	–	–	–	–	–	–	–	–	530.8	640	281506	9961	2079	14881	1	640	17.06	63.53	170.16	TL
Number of optima			0			–			9			19											

Table 5: Computational results for `mggdb` dataset with $\beta = 0.35$

			CG			**B&C&P**			**B&C**			**New.B&C**											
FILE	m	η	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\bar{\lambda}$	GAP	SEC	$\underline{\lambda}$	$\bar{\lambda}^H$	CON	CAP	ODD	BAL	NOD	$\bar{\lambda}$	GAP	SEC[1]	SEC[2]	SEC
`mggdb19`	3	9	51	51*	0.00	51	51*	0.00	51*	0.00	0.21	51	57	65	18	4	72	5	51*	0.00	0.26	0.33	0.79
`mggdb4`	4	17	240	256	6.25	242	242*	0.00	242*	0.00	27.59	237	250	1021	347	27	389	336	242*	0.00	1.93	0.87	14.07
`mggdb10`	4	24	267	306	12.75	267	268	0.37	268*	0.00	813.59	268	283	1902	311	95	662	10	268*	0.00	0.94	3.31	6.32
`mggdb15`	4	18	44	44*	0.00	44	44*	0.00	44*	0.00	0.72	44	44	0	0	0	0	0	44*	0.00	0.31	0.93	1.24
`mggdb20`	4	20	93	96	3.13	96	96*	0.00	96*	0.00	75.81	93	105	309	77	35	330	518	96*	0.00	0.35	2.49	23.65
`mggdb1`	5	21	236	252	6.35	251	252	0.40	252*	0.00	601.02	252	274	1517	264	44	681	875	252*	0.00	1.38	2.79	41.89
`mggdb3`	5	20	236	243	2.88	243	243*	0.00	243*	0.00	1700.95	243	251	2250	334	63	690	1669	243*	0.00	1.02	1.75	109.00
`mggdb6`	5	21	249	262	4.96	262	262*	0.00	262*	0.00	1552.37	250	262	1437	208	44	756	4525	262*	0.00	0.95	1.54	140.80
`mggdb7`	5	22	263	282	6.74	272	272*	0.00	272*	0.00	412.65	272	280	1271	174	61	553	103	272*	0.00	0.69	2.05	12.86
`mggdb11`	5	41	301	315	4.44	301	313	3.91	303*	0.00	3875.68	303	343	10929	818	385	2515	129	303*	0.00	1.90	7.69	255.60
`mggdb14`	5	18	83	84	1.19	84	84*	0.00	84*	0.00	369.34	83	89	491	66	20	354	1710	84*	0.00	0.48	2.22	45.15
`mggdb16`	5	22	73	75	2.67	75	75*	0.00	75*	0.00	2271.37	73	81	23884	6047	905	10478	4304	75*	0.00	0.43	16.57	9276.60
`mggdb17`	5	23	61	63	3.17	62	62*	0.00	62*	0.00	0.85	62	66	59	20	3	291	1	62*	0.00	0.49	0.89	2.60
`mggdb18`	5	30	132	157	15.92	135	135*	0.00	135*	0.00	0.41	135	138	1124	102	88	1023	47	135*	0.00	0.56	3.52	12.36
`mggdb2`	6	22	280	284	1.41	284	284*	0.00	284*	0.00	1435.16	284	348	1231	296	105	805	145	284*	0.00	1.55	2.74	19.05
`mggdb5`	6	23	299	323	7.43	309	309*	0.00	309	6.11	TL	308.5	323	2328	447	111	1206	7668	309*	0.00	2.95	2.97	1056.07
`mggdb13`	6	24	417	417*	0.00	417	417*	0.00	417	2.98	TL	410	–	1132	4722	4	433	7071	417*	0.00	1.08	1.56	674.09
`mggdb21`	6	28	116	125	7.20	120	120*	0.00	120	2.07	TL	120	136	1431	347	53	1013	55	120*	0.00	1.35	3.35	26.79
`mggdb12`	7	20	461	461*	0.00	461	461*	0.00	461*	0.00	15229.89	441	489	2807	778	141	1316	2342	461*	0.00	2.12	4.78	197.46
`mggdb22`	8	36	138	146	5.48	139	139*	0.00	–	–	–	139	145	2924	464	141	1790	459	139*	0.00	1.14	6.19	266.86
`mggdb8`	10	38	311	338	7.99	315	316	0.32	–	–	–	303	337	20701	1959	392	6803	7792	337	10.09	14.50	10.19	TL
`mggdb9`	10	45	260	275	5.45	265	266	0.38	–	–	–	260	292	24795	8738	780	9867	3985	292	10.96	18.43	20.17	TL
`mggdb23`	10	44	178	233	23.61	179	179*	0.00	–	–	–	176	194	5266	2089	224	3496	21541	194	9.28	1.77	13.89	TL
Number of optima				4			18		16										20				

WARP1 (1st Workshop on Arc Routing Problems), Copenhagen, 2013.

B.L. Golden, J.S. Dearmon, and E.K. Baker. Computational experiments with algorithms for a class of routing problems. *Computers and Operations Research*, 10:47–59, 1983.

L. Gouveia and F. Saldanha-da Gama. On the capacitated concentrator location problem: a reformulation by discretization. *Computers and Operations Research*, 33:1242–1258, 2006.

L. Gouveia, M.C. Mourão, and L.S. Pinto. Lower bounds for the mixed capacitated arc routing problem. *Computers and Operations Research*, 37:692–699, 2010.

J.C.A Gutiérrez, D. Soler, and A. Hervás. The capacitated general routing problem on mixed graphs. *Revista Investigacion Operacional*, 23:15–26, 2002.

G. Hasle, O. Kloster, M. Smedsrud, and K. Gaze. Experiments on the node, edge, and arc routing problem. *SINTEF ICT, Report No. A23265 (2012-05-21)*, 2012.

J.K. Lenstra and A.H.G. Rinnooy Kan. On general routing problems. *Networks*, 6:273–280, 1976.

A.N. Letchford. New inequalities for the general routing problem. *European Journal of Operational Research*, 96:317–322, 1996.

A.N. Letchford. The general routing polyhedron: A unifying framework. *European Journal of Operational Research*, 112: 122–133, 1999.

Y. Nobert and J.-C. Picard. An optimal algorithm for the mixed chinese postman problem. *Networks*, 27:95–108, 1996.

C.S. Orloff. A fundamental problem in vehicle routing. *Networks*, 4:35–64, 1974.

M.W. Padberg and M.R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7:67–80, 1982.

R. Pandit and B. Muralidharan. A capacitated general routing problem on mixed networks. *Computers and Operations Research*, 22:465–478, 1995.

C. Prins and S. Bouchenoua. A memetic algorithm solving the VRP, the CARP and general routing problems with nodes, edges and arcs. In W.E. Hart, N. Krasnogor, and J.E. Smith, editors, *Recent advances in memetic algorithms (Studies in fuzziness and soft computing, Vol. 166)*, pages 65–85. Springer, 2005.

G. Reinelt and D.O. Theis. On the general routing polytope. *Discrete Applied Mathematics*, 156:368–384, 2008.

P. Toth and D. Vigo, editors. *The Vehicle Routing Problem.* SIAM, Philadelphia, 2002.

P. Toth and D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications.* MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2014.

L.A. Wolsey. *Integer Programming.* Wiley, Chichester, New York, 1998.

Table 6: Computational results for `mgval` dataset with $\beta = 0.35$

			CG			B&C&P			B&C			New.B&C											
FILE	m	η	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\underline{\lambda}$	$\bar{\lambda}$	GAP	$\bar{\lambda}$	GAP	SEC	$\underline{\lambda}$	$\bar{\lambda}^H$	CON	CAP	ODD	BAL	NOD	$\bar{\lambda}$	GAP	SEC[1]	SEC[2]	SEC
mgval1A	2	47	152	185	17.84	–	–	–	158*	0.00	0.18	158	162	34126	270	84	477	1	158*	0.00	0.54	4.88	7.50
mgval2A	2	40	–	–	–	–	–	–	286*	0.00	1.99	286	299	32529	629	83	751	1	286*	0.00	0.50	3.31	9.48
mgval3A	2	43	–	–	–	–	–	–	84*	0.00	2.94	84	90	43329	445	102	655	4	84*	0.00	1.85	4.13	14.66
mgval1B	3	48	–	–	–	–	–	–	192*	0.00	39.57	190	208	26463	3302	251	1883	1276	192*	0.00	0.61	9.68	313.01
mgval2B	3	46	–	–	–	–	–	–	326*	0.00	135.13	326	344	65861	4763	217	1594	3704	326*	0.00	1.52	5.68	577.02
mgval3B	3	41	–	–	–	–	–	–	113*	0.00	128.98	113	115	37322	1311	118	1510	78	113*	0.00	1.82	4.60	32.98
mgval4A	3	84	–	–	–	–	–	–	430	1.94	TL	430	458	451528	4898	902	5302	2	430*	0.00	5.04	41.54	449.35
mgval5A	3	82	–	–	–	–	–	–	454	3.30	TL	454	477	491690	5732	801	4971	1672	454*	0.00	3.21	30.47	2396.47
mgval6A	3	64	–	–	–	–	–	–	248*	0.00	50.23	248	256	56801	1165	274	2288	8	248*	0.00	0.73	7.15	126.41
mgval7A	3	78	–	–	–	–	–	–	264*	0.00	1.19	264	278	13036	395	271	1539	6	264*	0.00	0.97	21.11	134.25
mgval8A	3	84	–	–	–	–	–	–	415	0.24	TL	415	446	306694	1680	427	3232	3	415*	0.00	0.82	22.25	296.02
mgval9A	3	116	–	–	–	–	–	–	324*	0.00	1071.31	324	341	454788	2815	760	5740	9	324*	0.00	1.30	95.38	2698.20
mgval10A	3	122	–	–	–	–	–	–	475*	0.00	2230.29	475	502	1468685	4928	1153	8683	16	475*	0.00	3.47	100.54	7015.32
mgval4B	4	90	–	–	–	–	–	–	531	5.85	TL	529	531	276	7	0	56	2128	531	0.38	4.52	50.54	TL
mgval5B	4	81	–	–	–	–	–	–	467	4.57	TL	466	502	249130	6257	915	6036	3254	467	0.21	5.51	36.38	TL
mgval6B	4	62	–	–	–	–	–	–	250*	0.00	1715.66	250	271	79128	3263	478	3909	571	250*	0.00	3.27	15.83	717.58
mgval7B	4	79	–	–	–	–	–	–	325*	0.00	208.79	325	338	105168	1536	288	3095	195	325*	0.00	1.75	30.89	1756.39
mgval8B	4	78	–	–	–	–	–	–	385	2.34	TL	385	454	204077	4259	846	4696	25	385*	0.00	0.90	29.02	999.01
mgval9B	4	106	–	–	–	–	–	–	332	3.46	TL	331	358	495547	9622	2060	12081	367	331*	0.00	1.53	147.02	12717.94
mgval10B	4	118	–	–	–	–	–	–	461	0.87	TL	461	492	918347	5674	1267	7526	24	461*	0.00	3.00	102.78	9490.73
mgval4C	5	93	–	–	–	–	–	–	516	7.13	TL	516	587	591685	9295	1005	7785	1282	516*	0.00	7.48	70.79	13985.61
mgval5C	5	82	–	–	–	–	–	–	586	7.77	TL	579	613	233054	4341	687	5933	2416	586	1.19	3.96	50.15	TL
mgval9C	5	115	–	–	–	–	–	–	329	3.95	TL	328	329	295	5	0	75	1178	329	0.30	5.39	145.38	TL
mgval10C	5	122	–	–	–	–	–	–	431	4.61	TL	428	431	694	8	0	165	1648	431	0.70	4.37	185.38	TL
mgval3C	7	40	143	149	4.03	–	–	–	150	6.84	TL	149	150	223	15	0	319	1814	150	0.67	3.35	18.45	TL
mgval1C	8	48	–	–	–	–	–	–	–	–	–	272	312	24820	2646	532	6369	10915	312	12.82	2.21	34.66	TL
mgval2C	8	45	–	–	–	–	–	–	–	–	–	482	520	46451	5405	474	7365	11806	520	7.31	7.14	23.39	TL
mgval4D	9	96	–	–	–	–	–	–	–	–	–	640.5	729	222925	25178	3875	22567	1	729	12.07	31.81	220.00	TL
mgval5D	9	80	–	–	–	–	–	–	–	–	–	568	658	223254	13130	3635	19493	1	658	13.68	9.28	61.38	TL
mgval7C	9	82	–	–	–	–	–	–	–	–	–	336	390	138293	4687	1061	9382	2	390	13.85	16.73	100.83	TL
mgval8C	9	75	–	–	–	–	–	–	–	–	–	487	602	123893	11772	2169	15512	939	602	19.10	13.95	53.41	TL
mgval6C	10	60	–	–	–	–	–	–	–	–	–	303	372	31749	8475	1119	9091	2237	372	18.55	17.01	31.32	TL
mgval9D	10	115	–	–	–	–	–	–	–	–	–	422	491	180649	8511	2230	15749	1	491	14.05	18.02	124.75	TL
mgval10D	10	114	–	–	–	–	–	–	–	–	–	519	594	455451	19154	3846	23545	1	594	12.63	43.53	230.53	TL
Number of optima				0			–		12										19				