Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7, D-14195 Berlin-Dahlem, Germany

Andreas Löbel

Experiments with a Dantzig-Wolfe Decomposition for Multiple-Depot Vehicle Scheduling Problems

Preprint SC 97–16 (May 1997)

Experiments with a Dantzig-Wolfe Decomposition for Multiple-Depot Vehicle Scheduling Problems*

Andreas Löbel**

Abstract

In this paper, we present a Dantzig-Wolfe decomposition for the $\mathcal{NP}$-hard multiple-depot vehicle scheduling problem in public mass transit. It turned out that such a decomposition approach is an unsuitable method to solve such kind of multicommodity flow problems. The major obstacle to solve such problems is that the continuous master problem relaxations become too hard to be solved efficiently. Especially for problems with more than one thousand timetabled trips, the LU factorization in solving a restricted master problem takes far too much time. We will describe our computational experiments in detail and discuss the reasons why the decomposition method fails in this case.

Our computational investigations are based on real-world problems from the city of Hamburg with up to 2,283 timetabled trips. Our decomposition implementation is compared with a delayed column generation to solve the linear programming (LP) relaxation directly. This LP method can solve the LP relaxations of the integer linear programming formulation exactly for truly large-scale real-world problems of the cities of Berlin and Hamburg.

Mathematics Subject Classification (1991): 49M27, 90B06, 90C05.

1 Introduction

In recent years, many research groups have reported on successful applications of a Dantzig-Wolfe column generation applied to multicommodity flow problems, especially in the fields of transportation and telecommunication. For instance, Ribeiro and Soumis [1994] report on a column generation approach for the Multiple-Depot Vehicle Scheduling Problem (MDVSP) and give computational results for randomly generated problems with up to 300 timetabled trips. Carraresi, Girardi, and Nonato [1995] and Desrosiers and Rousseau [1995] have worked on crew scheduling problems: The first report on real-world airline and bus crew scheduling problems having up to 2.2 billion columns; the latter do not report about the details of their test problems, but claim that their system is successfully applied to bus crew scheduling problems of several cities. Barnhart, Hane, and Vance

*This work has been supported by the German Federal Ministry of Education, Science, Research, and Technology grant no. 03-GR7ZIB -7. Responsibility for the contents lies with the author.

**Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Takustraße 7, D-14195 Berlin, Germany, E-mail loebel@zib.de, URL http://www.zib.de/.

[1996] report on experiments with randomly generated problems and commercial transportation problems. A further successful application of a column generation approach is given to on a project in telecommunication in Alevras, Grötschel, and Wessäly [1996].

On the other side, Ahuja, Magnanti, and Orlin [1993] claim that Dantzig-Wolfe decomposition is only an efficient method if we measure efficiency by the number of iterations of the decomposition algorithm. However, solving the Dantzig-Wolfe master problems can be a computationally expensive task, and the Dantzig-Wolfe decomposition method has generally not proven to be an efficient method for solving the multicommodity flow problem. Indeed, we will see that it can be too expensive.

This paper is addressed to a Dantzig-Wolfe decomposition for the $\mathcal{NP}$-hard Multiple-Depot Vehicle Scheduling Problem (MDVSP) in public mass transit. The MDVSP has the task to assign vehicles stationed at different depots to a given set of timetabled tripssuch that some operational restrictions are satisfied and a given objective function is minimized. A detailed problem description can be found, e. g., in Löbel [1997b], Löbel [1997c], or a German version in Grötschel, Löbel, and Völker [1997]; we give here only a summary.

We compare the decomposition approach with a direct LP method that is based on delayed column generation, see Löbel [1996]. This method employs new column generation techniques that are based on two different Lagrangean relaxation approaches. The implementation can solve the LP relaxation of all our real-world instances considered here to optimality within reasonable running times. Combining this LP method and an iterative LP rounding heuristic within a branch-and-cut method, we are able to compute an optimal integral solution for all of our test instances, see Löbel [1997b] or Löbel [1997c].

The results of this comparison is that the decomposition implementation is completely inferior compared to the direct LP method. We will report about our computational tests and discuss some difficulties we have encountered in solving our test problems using decomposition.

Our work is related to Ribeiro and Soumis [1994], who have reported on their experience with a decomposition for the capacitated MDVSP already in 1991 (preprint).

The decomposition method presented here for the MDVSP is due to Dantzig and Wolfe [1960]. The general decomposition principle and its economic interpretation is outlined in Chvátal [1980] and for network multicommodity flow problems in Ahuja, Magnanti, and Orlin [1993]. We will use most of the column generation concepts proposed for the MDVSP by Desrosiers, Dumas, Solomon, and Soumis [1995]. In the following, we assume the reader to be familiar with linear programming, decomposition and relaxation approaches, and network flows.

In order to prevent misunderstandings, we want to state explicitly that our paper does not make any judgements about the efficiency or the capability of the Dantzig-Wolfe decomposition in general. It just reports on an application of the Dantzig-Wolfe decomposition method for some concrete real-world vehicle scheduling problems, for which the decomposition method turned out to be completely useless, especially compared with a tailor-made LP method.

2 Mathematical Models

We first give a summary of the MDVSP: The set of vehicles of a transportation company is divided into so-called **depots**, denoted by $\mathcal{D}$. A given set of lines together with a timetable define a set of so-called **timetabled trips** (or **passenger trips**), denoted by $\mathcal{T}$. There are given for each trip $t \in \mathcal{T}$ its departure time s_t, its arrival time e_t, and a nonempty set of valid depots, called **depot-group** and denoted by $G(t) \subseteq \mathcal{D}$. Only the vehicles of the depots $G(t)$ are allowed to service the trip $t \in \mathcal{T}$; let $\mathcal{T}_d := \{t \in \mathcal{T} |\, d \in G(t)\}$ for all $d \in \mathcal{D}$. There are further types of trips, all running without passengers: A **pull-out trip** connecting a depot with some timetabled trip, a **pull-in trip** connecting some timetabled trip with a depot, and a **dead-head trip** connecting two successive timetabled trips. For simplification, we call them all **unloaded trips**.

For two trips t and $t' \in \mathcal{T}$, let $\Delta_{t,t'} \geqslant 0$ be given. The term $\Delta_{t,t'}$ can have different meanings: In the literature, it is mostly equivalent to the duration (travel plus layover time) from the location of the last stop of t to the location of the first stop of t', e. g., see Ribeiro and Soumis [1994]. However, our partners from practice use such a definition of $\Delta_{t,t'}$ only for those dead-head trips for which their idle times or the differences $s_{t'} - e_t$ do not exceed a predefined maximum duration ranging from 40 to 120 minutes, e. g., see Daduna and Mojsilovic [1988]. Otherwise, $\Delta_{t,t'}$ is set to infinity. Dead-head trips exceeding the maximum allowed duration are currently not considered in practice for various reasons, e. g., the driver's idle time (e. g., break) would become too long etc. To make it nevertheless possible to link two successive trips t and t' with some dead-head trip exceeding a maximum allowed duration, we set $\Delta_{t,t'} := s_{t'} - e_t$ whenever it is possible to park a vehicle between t and t' at the depot. We call these artificially generated trips also pull-in-pull-out trips.

If $e_t + \Delta_{t,t'} \leqslant s_{t'}$ is satisfied, the corresponding dead-head trip is called **compatible**. Pull-in and pull-out trips are always considered to be compatible. The unloaded trips are used to link the timetabled trips to **vehicle schedules**, each being a chain of trips such that the first trip is a pull-out trip, the last trip is a pull-in trip, and the timetabled and unloaded trips occur alternately. A vehicle schedule is feasible if there exists a depot that can service all its trips. If a vehicle schedule contains no pull-in-pull-out trip, it is also called a **block** (or **rotation**), and a block that contains just one single timetabled trip is also called a **tripper**.

The task of the MDVSP is to provide a set of feasible vehicle schedules covering each trip $t \in \mathcal{T}$ exactly once. The main objective is to use as few vehicles, i. e., vehicle schedules, as possible and, subordinate, with minimum operational costs for the used trips among all fleet minimal solutions.

We will now state an integer linear programming (ILP) model for the MDVSP. For each depot $d \in \mathcal{D}$, we introduce a digraph $D_d = (V_d, A_d)$ with node set $V_d := \mathcal{T}_d \cup \{d\}$ and arc set $A_d := \{a \in V_d \times V_d |\, a \text{ is a compatible unloaded trip}\}$. The tuple (d, d) is not considered to be compatible. For each $d \in \mathcal{D}$ and each $a \in A_d$, we introduce an integer variable x_a^d that denotes a decision variable indicating whether a vehicle of the depot d runs the trip a or not. The variables x_a^d are combined to vectors $x^d := (x_a^d)_{a \in A_d} \in \mathbb{R}^{A_d}$,

$d \in \mathcal{D}$, and $x := (x^d)_{d\in\mathcal{D}} \in \mathbb{R}^A$. Our two-stage cost function is realized as follows: With each unloaded trip $a \in A_d$, we associate a weight $c_a^d \in \mathbb{Q}$ representing its operational costs. In addition, we add to the weight of each pull-out trip a sufficiently large **big M** standing for the capital costs and being larger than the operational costs of any feasible solution.

Our ILP of the MDVSP reads

$$\min \sum_{d\in\mathcal{D}} \sum_{(i,j)\in A_d} c_{ij}^d \, x_{ij}^d \tag{1a}$$

subject to

$$\sum_{d\in G(t)} \sum_{(t^+,j)\in A_d} x_{t^+j}^d = 1, \qquad \forall\, t \in \mathcal{T}, \tag{1b}$$

$$\sum_{(t^+,j)\in A_d} x_{t^+j}^d - \sum_{(i,t^-)\in A_d} x_{it^-}^d = 0, \qquad \forall\, t \in \mathcal{T}_d \;\; \forall\, d \in \mathcal{D}, \tag{1c}$$

$$x \geqslant 0 \text{ and integral.} \tag{1d}$$

Constraints (1b), the so-called **flow conditions**, ensure that each timetabled trip is serviced exactly once, and constraints (1c) are the so-called **flow conservations**. For notational simplification, we deliberately do not consider depot capacities for our computational investigations. The LP relaxation of (1) is given by

$$\min_{\substack{x \geqslant 0 \text{ satisfying} \\ (1\text{b}) \text{ and } (1\text{c})}} \sum_{d\in\mathcal{D}} \sum_{a\in A_d} c_a^d \, x_a^d. \tag{2}$$

The LP relaxation (2) is formulated in terms of arc flows. It is also possible to reformulate the flow, which satisfies condition (1c), in terms of nonnegative cycle flows as follows (see the Flow Decomposition Theorem in Ahuja, Magnanti, and Orlin [1993] and its special application if x is a circulation as in our case):

For all $d \in \mathcal{D}$, let $X_d := \{x^d \in \mathbb{R}^{A_d} |\, x^d \geqslant 0 \text{ satisfying (1c)}\}$, $\mathcal{W}_d := \{S \subset A_d |\, S \text{ is a vehicle schedule for depot } d\}$, and $W_d := \{\chi^S |\, S \in \mathcal{W}_d\}$ (each $\mathcal{W}_d$ describes exactly the set of directed cycles in D_d and W_d describes the incidence vectors thereof). The set of all incidence vectors is denoted by $W := \dot{\bigcup}_{d\in\mathcal{D}} W_d$. We will consider W_d also as the matrix $(\chi^S)_{S\in\mathcal{W}}$ that is defined by arranging the elements of W_d column-wise. Now, for each $d \in \mathcal{D}$, it is easy to show that $x^d \in X_d$ if and only if there exists $0 \leqslant \mu^d \in \mathbb{R}^{W_d}$ such that $x^d = W_d \, \mu^d$. Inserting each $x^d = W_d \, \mu^d$ into (2) results in a so-called **master problem (MP)**

$$\min \sum_{d\in\mathcal{D}} \sum_{w\in W_d} c_w^d \, \mu_w^d \tag{3a}$$

subject to

$$\sum_{d\in\mathcal{D}} \sum_{w\in W_d} a_{tw}^d \, \mu_w^d = 1, \qquad \forall\, t \in \mathcal{T}, \tag{3b}$$

$$\mu^d \geqslant 0, \qquad \forall\, d \in \mathcal{D}, \tag{3c}$$

where each $c_w^d := \sum_{a \in A_d} c_a^d \, w_a$ denotes the costs of the vehicle schedule associated with the cycle $w \in W_d$, and each a_{tw}^d is set to 1 if $w \in W_d$ covers the trip $t \in \mathcal{T}$ and is set to 0 otherwise. Obviously, the master problem (3) is a set partitioning problem and yields the same optimal value as the LP relaxation (2). Moreover, each solution of (3) has a representation in (2) and vice versa.

Although the master problem (3) has significantly fewer equations than the LP relaxation, it contains exponentially many variables. Nevertheless, we can solve MP with a delayed column generation approach.

Let the **restricted master problem (RMP)** be the linear program that is defined by the columns of MP corresponding with some $\tilde{W} \subseteq W$. We assume that RMP is at least primal feasible. Let $\nu \in \mathbb{R}^{\mathcal{T}}$ denote the dual multipliers of (3b) for the optimal basis of a current RMP. For convenience of notation, we define for each $d \in \mathcal{D}$ an artificial variable $\nu_d \in \mathbb{R}$ and set it to zero. The reduced costs of the columns of the master problem are given by $\overline{c}_w^d := c_w^d - \sum_{t \in \mathcal{T}} \nu_t \, a_{tw}^d$, which can easily be rewritten as $\overline{c}_w^d = \sum_{ij \in A_d} (c_{ij}^d - \nu_i) \, w_{ij}$. A given basis is optimal if and only if for all $w \in W_d$ and for all $d \in \mathcal{D}$ the reduced costs $\overline{c}_w^d$ are nonnegative. Thus, a given basis of RMP is also optimal for MP if the so-called **pricing problem (PP)**

$$\min_{d \in \mathcal{D}} \min_{w \in W_d} \sum_{ij \in A_d} (c_{ij}^d - \nu_i) \, w_{ij} \tag{4}$$

yields a nonnegative value.

3 Implementation

We have implemented such a delayed column generation approach of the Dantzig-Wolfe decomposition for our MDVSP. Since we assume that the initial RMP is at least primal feasible, we initialize it with the columns associated with a feasible set of vehicle schedules, e. g., given by some primal heuristic as the nearest depot (ND) heuristic, see Löbel [1996]. ND assigns each timetabled trip to one depot of its depot-group (cluster first) and solves each resulting single-depot instance to optimality (schedule second).

We attack the pricing problem as follows: Let PP_d denote the pricing problem for a fixed $d \in \mathcal{D}$. We split the depot node d into two new nodes d^+ and d^- such that d^+ becomes the new tail node of all pull-out trips, and d^- becomes the new head node of all pull-in trips. Then, PP_d is the problem to find the shortest path from d^+ to d^- according to the arc weights $\tilde{c}^d := (c_{ij}^d - \nu_i)_{ij \in A_d}$. Fortunately, the underlying network for each PP_d with the two new depot nodes is acyclic. The shortest paths – even if negative arc weights occur – can therefore be computed in $\mathcal{O}(|A_d|)$ time using, e. g., the *reaching* algorithm as proposed in Ahuja, Magnanti, and Orlin [1993].

The algorithm proceeds in the following way: For $d \in \mathcal{D}$, let $\tilde{w}^d$ denote the shortest path for PP_d. If $\tilde{c}^d \, \tilde{w}^d \geqslant 0$ for all $d \in \mathcal{D}$, the optimal basis of the current RMP is also optimal for MP, and we are done. Otherwise, the column of at least one $\tilde{w}^d$ violating the optimality condition is generated and added to RMP, the enlarged RMP is reoptimized, and we iterate. Between two consecutive RMPs, we generate for each depot d the column corresponding with the shortest path $\tilde{w}^d$ if $\tilde{c}^d\,\tilde{w}^d < 0$.

Lagrangean relaxation can also be used to generate columns. This column generation technique, which we call **Lagrangean pricing**, can significantly accelerate the column generation. We explain Lagrangean pricing for our decomposition: Consider a slightly different formulation of the LP relaxation (2) where each node $t \in \mathcal{T}$ is replaced by two nodes t^- and t^+. t^- becomes the new head and t^+ becomes the new tail of all arcs that are incident to t. For each $d \in \mathcal{D}$, let $\tilde{A}_d$ denote the modified set A_d due to this node splitting. We introduce for each depot and each $t \in \mathcal{T}_d$ a new arc (t^-, t^+), a new variable $x^d_{t^-t^+}$, and a new, but redundant constraint $x^d_{t^-t^+} \leqslant 1$. The enlarged LP relaxation reads

$$\min \sum_{d\in\mathcal{D}} \sum_{(i,j)\in\tilde{A}_d} c^d_{ij}\, x^d_{ij} \tag{5a}$$

subject to

$$\sum_{d\in G(t)} \sum_{(t^+,j)\in\tilde{A}_d} x^d_{t^+j} = 1, \qquad \forall\, t \in \mathcal{T}, \tag{5b}$$

$$x^d_{t^-t^+} - \sum_{(i,t^-)\in\tilde{A}_d} x^d_{i\,t^-} = 0, \qquad \forall\, t \in \mathcal{T}_d \;\; \forall\, d \in \mathcal{D}, \tag{5c}$$

$$\sum_{(t^+,j)\in\tilde{A}_d} x^d_{t^+j} - x^d_{t^-t^+} = 0, \qquad \forall\, t \in \mathcal{T}_d \;\; \forall\, d \in \mathcal{D}, \tag{5c}$$

$$x^d_{t^-t^+} \leqslant 1, \qquad \forall\, t \in \mathcal{T}_d \;\; \forall\, d \in \mathcal{D}, \tag{5d}$$

$$x \geqslant 0 \tag{5d}$$

Consider the Lagrangean function of (5) where the flow conditions (5b) have been put into the objective function. It is easy to see that this Lagrangean function decomposes into $|\mathcal{D}|$ independently solvable minimum-cost flow circulation problems. The solution of each of these minimum-cost flow problems can be interpreted as a set of vehicle schedules that seem to be advantageous with respect to the given Lagrangean multipliers.

This simple observation can be used within a column generation as follows: We use the value of the optimal dual multipliers ν of the last RMP as estimators of the Lagrangean multipliers of (5b) and evaluate the Lagrangean function at ν. Then, we generate the columns of those vehicle schedules proposed by the Lagrangean relaxation for which the reduced cost criterion for MP is not satisfied and which are currently not active. In addition to the standard column generation scheme, this Lagrangean pricing is always used between two consecutive RMPs to generate further columns.

The concept of Lagrangean pricing is also used for the direct LP method that we compare to our decomposition implementation. Further details of the Lagrangean pricing for the LP method are given in Löbel [1996].

4 Computational Experiments

Our test data for the computational tests of this paper are based on some smaller and not so hard problems from the Hamburger Hochbahn AG (HHA). At the moment, HHA together with other transportation companies maintain 14 garages with 9 different vehicle types. The existing garage-vehicle combinations define 40 depots. In Hamburg, more than 16 thousand daily timetabled trips must be scheduled with about 15.1 million unloaded trips and about 800 buses. The problem in Hamburg decomposes into a 12-depot problem, a 9-depot problem, five smaller 2-depot problems, and nine small 1-depot problems. We consider for our tests the 19 smallest of the 40 depots as single-depot instances by assuming that all timetabled trips of $\mathcal{T}_d$ have to be serviced alone by the depot d. The other 21 depots are too large to be solved by decomposition. From the multiple-depot problems, we just consider the five 2-depot problems and the 9-depot problem. These considered problems have in common that the optimal LP and MP value is equal to the integer optimum.

Table 1 gives some statistics of our test set as the number of depots ($|\mathcal{D}|$), the number of timetabled trips ($|\mathcal{T}|$), the number of unloaded trips ($|A|$), the average number of depots in $G(t)$, the number of possible vehicle schedules, and the number of possible blocks. Besides the total number of arcs in $A^{\text{u-trip}}$, we also give the number of unloaded trips without pull-in-pull-out trips, which we denote as user-defined trips.

In the following, we first give some general observations about our computational tests. Second, we report on specific results for the single-depot problems and, third, on specific results for the multiple-depot problems. All tests have been performed on a SUN Model 170 UltraSPARC with 512 MByte main memory, all linear programs have been solved with CPLEX, version 4.0.7. For each test problem, we have used a time limit of three hours. This time limit is sufficiently large since the LP relaxation of the biggest test problem Hamburg 2 can be solved to optimality within less than 400 seconds using the direct LP method described in Löbel [1996].

General Observations.

- For our test set, the pricing problems PP_d are efficiently solvable within a small part of the total running time, which can be neglected.
- The right-hand side of MP is equal to $\mathbb{1}$ leading to degenerate LPs. From version 4.0 on, CPLEX provides a new, less aggressive perturbation method that can handle degenerate LPs efficiently. And indeed, we have never recognized any numerical problems stemming from degeneracy since we have used the first beta version of CPLEX 4.0.
- In one of the first versions of our implementation, each cost coefficient c_w^d was just the sum of big M and the operational costs of the corresponding vehicle schedule $w \in W_d$. The naive use of this two stage objective function with a big M equal to 10^8 has caused numerical difficulties in proving global optimality: The optimality tolerance of CPLEX can only be set to a value between 10^{-9} and 10^{-4}. This means

Test Set[a]	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert A\rvert$ User-def.	$\lvert A\rvert$ Total	$\varnothing\lvert G(t)\rvert$	$\lvert W\rvert$	#blocks
Hamburg 14	1	109	617	5,241	1.0	$2.0 \cdot 10^{13}$	$2.3 \cdot 10^{10}$
Hamburg 15	1	183	1,025	14,347	1.0	$2.0 \cdot 10^{16}$	$7.1 \cdot 10^{9}$
Hamburg 2 - Depot 4	1	211	2,651	16,854	1.0	$3.9 \cdot 10^{14}$	$7.5 \cdot 10^{10}$
Hamburg 7 - Depot 2	1	220	2,348	21,315	1.0	$5.5 \cdot 10^{16}$	$7.0 \cdot 10^{11}$
Hamburg 7 - Depot 1	1	232	3,922	12,510	1.0	$1.1 \cdot 10^{16}$	$4.7 \cdot 10^{11}$
Hamburg 2 - Depot 8	1	493	8,322	104,153	1.0	$1.6 \cdot 10^{18}$	$8.0 \cdot 10^{11}$
Hamburg 3 - Depot 1	1	521	19,003	65,650	1.0	$8.8 \cdot 10^{18}$	$2.4 \cdot 10^{13}$
Hamburg 2 - Depot 5	1	648	14,136	180,554	1.0	$3.9 \cdot 10^{18}$	$4.3 \cdot 10^{11}$
Hamburg 2 - Depot 7	1	695	16,526	186,863	1.0	$1.3 \cdot 10^{20}$	$6.3 \cdot 10^{13}$
Hamburg 1 - Depot 7	1	728	13,654	233,010	1.0	$2.3 \cdot 10^{23}$	$5.9 \cdot 10^{16}$
Hamburg 1 - Depot 12	1	892	25,100	348,230	1.0	$3.2 \cdot 10^{24}$	$1.8 \cdot 10^{17}$
Hamburg 5 - Depot 2	1	930	27,452	368,242	1.0	$5.3 \cdot 10^{25}$	$1.1 \cdot 10^{20}$
Hamburg 5 - Depot 1	1	986	56,760	211,153	1.0	$1.1 \cdot 10^{24}$	$1.6 \cdot 10^{18}$
Hamburg 1 - Depot 4	1	1,065	31,520	488,655	1.0	$1.4 \cdot 10^{25}$	$1.0 \cdot 10^{19}$
Hamburg 1 - Depot 11	1	1,296	77,201	459,468	1.0	$1.3 \cdot 10^{27}$	$2.3 \cdot 10^{20}$
Hamburg 6 - Depot 1	1	1,345	102,253	322,546	1.0	$2.4 \cdot 10^{22}$	$1.2 \cdot 10^{16}$
Hamburg 1 - Depot 10	1	1,588	79,955	404,016	1.0	$5.1 \cdot 10^{20}$	$1.4 \cdot 10^{14}$
Hamburg 6 - Depot 2	1	1,693	72,882	1,233,766	1.0	$5.0 \cdot 10^{28}$	$4.1 \cdot 10^{23}$
Hamburg 1 - Depot 3	1	1,716	116,142	850,828	1.0	$5.6 \cdot 10^{30}$	$1.3 \cdot 10^{25}$
Hamburg 4	2	238	2,000	23,000	1.04	$1.5 \cdot 10^{21}$	$2.4 \cdot 10^{15}$
Hamburg 7	2	341	6,000	34,000	1.32	$5.9 \cdot 10^{16}$	$1.2 \cdot 10^{12}$
Hamburg 3	2	791	30,000	200,000	1.32	$2.3 \cdot 10^{22}$	$3.9 \cdot 10^{17}$
Hamburg 5	2	1,461	85,000	580,000	1.31	$5.3 \cdot 10^{25}$	$1.1 \cdot 10^{20}$
Hamburg 6	2	2,283	176,000	1,600,000	1.33	$5.0 \cdot 10^{28}$	$4.2 \cdot 10^{23}$
Hamburg 2	9	1,834	99,000	1,000,000	2.02	$3.3 \cdot 10^{21}$	$5.3 \cdot 10^{15}$

[a] The problems are taken from Löbel [1996] and Löbel [1997b].

Table 1: Real-world test set taken from the city of Hamburg.

that the reduced costs $\tilde{c}^d\, \tilde{w}^d$ must be at least greater than or equal to -10^{-4} for each $w \in \tilde{W}$. If the total fleet size needs 3 digits or more and if the optimality tolerance is at least 4 decimal places, we have to compute accurately for at least 3+8+4 digits (the fleet size plus the operational costs plus the optimality tolerance), which is close to and can be smaller than the machine precision. For an "almost" optimal RMP (i. e., the value of PP (4) is smaller than, but close to -10^{-4}), cancellation of important digits can occur in computing the reduced costs. Sometimes the reduced costs of a generated column turned out to be feasible within the next RMP, but have been indicated to be infeasible within PP.

To keep out of such numerical troubles, we scale down the objective coefficients by 10^8 and set the optimality tolerance to 10^{-9}. Then, we have to compute accurately for 8 digits less than before. This simple trick makes the implementation more robust. Since then, we have never recognized any further cancellation of important digits in solving PP.

- Initializing the first RMP only with all columns corresponding with trippers leads to quite bad starting points: Within the time limit, none of the problems could be solved to optimality within, and only the single-depot problems with up to 493 timetabled trips and the multiple-depot problem Hamburg 7 could be solved fleet minimally with cost gaps ranging from 14 % up to 138 %. For all the other problems, we could not generate an (MP) solution yielding the minimum number of vehicles. Moreover, the basis solutions of the last RMPs are completely fractional and there are almost no variables with a value greater than or equal to 0.5, but most nonzero values are smaller than 0.2. This may have a negative influence on the performance of a branch-and-price algorithm.

  Instead of starting with all trippers, we use the columns corresponding with a heuristically generated nearest depot solution or, to see some computational effects, with an integer optimal solution provided by Löbel [1997b]. Note that ND already generates an optimal solution for single-depot instances.

- The average number of nonzero elements per column of the occurring RMPs is approximated by the average number ot timetabled trips of the generated vehicle schedules. For our test set, this number is 17.3 with a standard deviation of 4.8; the lowest average column length of our problems is 8.4 and the largest is 35.2. Our MPs are quite dense set partitioning problems. It is known that the LU factorization becomes the bottleneck in solving such problems with more than thousand timetabled trips (a similar observation has been made by Borndörfer [1997], he solves set partitioning problems arising from large-scale vehicle routing from a dial-a-ride system for handicapped people in Berlin).

- Fixing the number of timetabled trips, the investigations of Ribeiro and Soumis [1994] indicate that the running time is linear in the number of depots. We have too few test problems to make such a statement, but this result of Ribeiro and Soumis sounds reasonable.

The Single-Depot Problems.

To get a feeling about the size that a problem can have to be successfully solved with a decomposition approach, we have first investigated our single-depot instances. The results are displayed in Fig. 1 showing on the axis of ordinates the running times with respect to the number of timetabled trips, unloaded trips, and vehicle schedules. The left column shows the results that we have obtained without Lagrangean pricing: We started with an optimal integral solution and always generated only the column delivered by PP. Stopping whenever the time limit was reached, it turned out that this strategy solves only the master problem of tiny problems with up to 200 timetabled trips (which are about 20 thousand dead-head trips and 10^{16} vehicle schedules) to optimality. If we just want to find an RMP including a fleet minimal solution (proved via sufficiently small negative reduced costs), it is possible to solve problems with up to 700 timetabled trips (200 thousand deadhead trips and 10^{20} vehicle schedules). The right column shows the results that we have obtained using in addition Lagrangean pricing, which significantly accelerates the solution convergence. With this technique, we are able to solve the master problem of problems with up to 700 timetabled trips optimally and with up to 1700 timetabled trips fleet minimally. A negative side effect of Lagrangean pricing is that there are more columns generated between two consecutive RMPs making each single RMP much harder to solve. It is not clear to us whether there is some break even point where this acceleration of iteration convergence is canceled out by larger LP running times.

It is well known that minimum-cost flow algorithms are efficient tools to solve (single commodity) flow problems. This might be a lame comparison, but using our network simplex algorithm MCF together with a delayed column generation, all the single-depot instances can be solved to optimality in less than 20 seconds, see Löbel [1997b]. MCF is an implementation of the primal and the dual network simplex algorithm in C and is available for academic use free of charge via WWW at URL http://www.zib.de/Optimization, see Löbel [1997a].

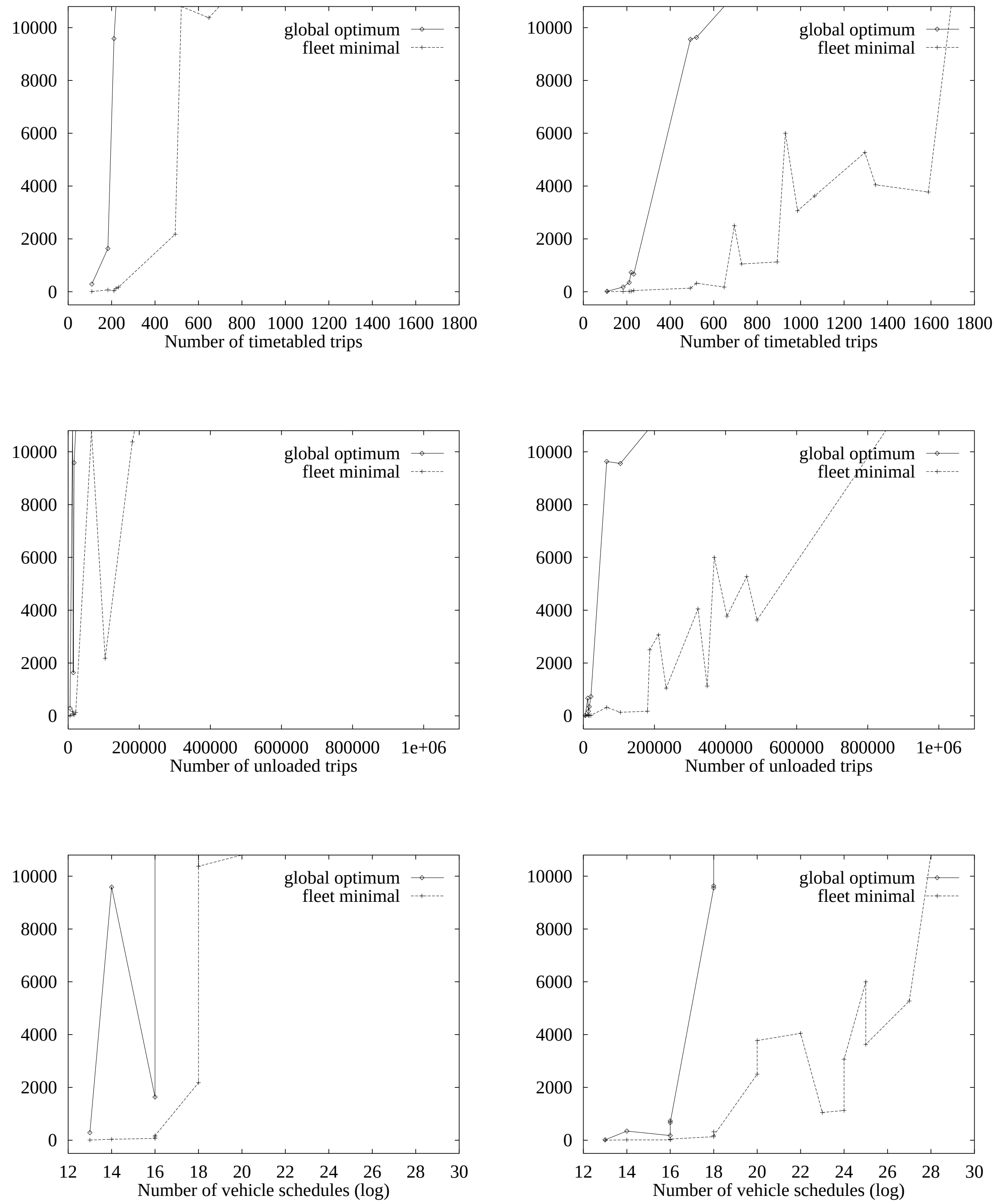
The Multiple-Depot Problems.

As we have already stated, starting from scratch (with all columns corresponding with trippers) does not work at all. So, we have used the solution computed with our ND heuristic as a starting point. The decomposition running times with and without Lagrangean pricing of these tests are compared with those provided in Löbel [1996]. All times are displayed in Tab. 2 showing that the decomposition without Lagrangean pricing solves only the two small problems Hamburg 4 and 7 fleet minimal, but not to global optimality. Using Lagrangean pricing accelerates this solution times, but does not lead to further significant improvements, e. g., solving more problems at least fleet minimal. Compared to a direct LP method, the decomposition implementation is completely inferior.

To see whether the decomposition method can at least prove the optimality of an optimal solution, we started the decomposition and the LP method with an integer optimal solution generated with the method presented in Löbel [1997b] and Löbel [1997c]. Table 3 gives the running times, the number of solved RMPs and restricted LPs, the number of generated columns, and the number of CPLEX iterations being performed.

First of all, the LP method is able to prove optimality for each of these problems within less than 400 seconds running time. Within a time limit of three hours, our decomposition implementation was unable to prove optimality of any of these problems, only the fleet minimality could be shown for the problems with up to 791 timetabled trips.

The larger the number of timetabled trips of an instance is, the more time is spent within the LU factorization. The worst case example is Hamburg 2: Applying in addition the Lagrangean pricing, only 34 RMPs could be solved within the time limit, which are by far too less to have a chance to solve such large problems within reasonable time.



WITHOUT Lagrangean pricing. WITH Lagrangean pricing.

Figure 1: Decomposition running times for the single-depot instances

Test Set[a]	Decomposition				LP[d]	
	Default[b]		Lagr. pricing[c]			
	fleet[e]	global[f]	fleet[e]	global[f]	fleet[e]	global[f]
Hamburg 4	84	TL	25	TL	2	3
Hamburg 7	4,210	TL	1,663	TL	8	9
Hamburg 3	TL[g]	—	TL[g]	—	18	31
Hamburg 5	TL[g]	—	TL	—	135	155
Hamburg 6	TL	—	TL	—	80	84
Hamburg 2	TL[g]	—	TL[g]	—	679	685

[a]ND generates an optimal solution for Hamburg 4 and 6.
[b]Without Lagrangean pricing.
[c]With Lagrangean pricing.
[d]Running times taken from Löbel [1996].
[e]Time to prove that the starting solution is fleet minimal.
[f]Time to prove global optimality of the last RMP and restr. LP.
[g]No objective improvement within the time limit.

Table 2: Running times starting with a solution generated with the ND heuristic.

5 Comparison to Ribeiro and Soumis

Already in 1991, Ribeiro and Soumis [1994] have reported about their computational results using a Dantzig-Wolfe decomposition for the MDVSP. They solved randomly generated capacitated MDVSPs with up to 300 timetabled trips and 6 depots to optimality using a branch-and-bound algorithm. Compared with our computational results, we wonder how they can solve even the integer formulation of their problems to optimality while we are unable to just solve the MP relaxations of our uncapacitated problems with a similar number of timetabled trips using a considerably faster workstation and a more sophisticated version of CPLEX:

- They do not mention the number of dead-head trips. If this number is small, the number of possible vehicle schedules is also relatively small. Each problem instance of Ribeiro and Soumis includes 60 % long timetabled trips with a duration uniformly distributed between three and five hours. The other 40 % are short timetabled trips with a duration uniformly distributed between 5 and 40 minutes. On the average, each timetabled trip has therefore a duration of 2 hours and 33 minutes. 70 % of the short timetabled trips are defined between 8.00 a. m. and 5.00 p. m. The duration of the timetabled trips given by our test set is, on the average, about 30 minutes, the morning peak begins earlier and the afternoon peak ends later than the peaks of their problems.
- The used depot capacities of Ribeiro and Soumis seem to be quite generous: For each test instance, they provide one vehicle for at most two up to three timetabled trips. In a city like Berlin, about 10 thousand buses would be necessary to run about

Test Set	Decomposition				LP[c]	
	Default[a]		Lagr. pricing[b]			
	A[d]	B[e]	A[d]	B[e]	A[d]	B[e]
Running times						
Hamburg 4	84	TL	27	TL	2	3
Hamburg 7	1,031	TL	204	TL	8	9
Hamburg 3	TL	—	656	TL	22	17
Hamburg 5	TL	—	TL	—	72	78
Hamburg 6	TL	—	TL	—	128	179
Hamburg 2	TL	—	TL	—	388	397
Number of RMPs and RLPs						
Hamburg 4	329	5,658	76	3,991	3	4
Hamburg 7	435	1,882	87	1,439	19	23
Hamburg 3	667	—	50	301	3	13
Hamburg 5	707	—	61	—	19	23
Hamburg 6	668	—	45	—	10	18
Hamburg 2	353	—	34	—	14	18
Generated columns / 1,000						
Hamburg 4	0	6	1	5	2	2
Hamburg 7	1	4	1	4	14	15
Hamburg 3	1	—	2	2	6	16
Hamburg 5	1	—	3	—	57	62
Hamburg 6	1	—	3	—	56	73
Hamburg 2	2	—	4	—	55	55
CPLEX iterations / 1,000						
Hamburg 4	5	487	3	332	0	0
Hamburg 7	41	293	11	210	1	1
Hamburg 3	57	—	7	54	0	1
Hamburg 5	3	—	26	—	2	3
Hamburg 6	1	—	12	—	4	7
Hamburg 2	38	—	19	—	19	19

[a]Without Lagrangean pricing.
[b]With Lagrangean pricing.
[c]Results obtained for the LP (2) with the delayed column generation code of Löbel [1996].
[d]Time, iterations, etc. to prove that the starting solution is fleet minimal.
[e]Time, iterations, etc. to prove global optimality of last RMP and restricted LP.

Table 3: Starting with an integer optimal solution.

25 thousand timetabled trips that have to be serviced daily in Berlin. The truth is that the Berliner Verkehrsbetriebe (BVG) maintain less than 2,000 buses. Even if we assume that for their problems only one half of the available fleet size is used – otherwise, a consideration of depot capacities becomes pointless – the resulting vehicle schedules would on the average contain five timetabled trips. In Berlin and Hamburg, this number is about 15 and 20.

- The only information provided by them is how many columns are generated, until optimality can be proved. The largest number of generated columns was 4,585 for an instance with 5 depots and 300 timetabled trips. They give neither information about the average length of the generated vehicle schedules nor about the vehicle demand of the optimal solutions. For our test set, we have generated, on the average, vehicle schedules with 17.3 timetabled trips (standard deviation was 4.8). The problem with the smallest value was 8.4 timetabled trips and the largest value was 35.2 timetabled trips.
- For the capacitated MDVSP, it is $\mathcal{NP}$-hard to find a feasible solution, see Löbel [1997b]. Ribeiro and Soumis, however, do not tell how they initialize the first RMP.

Based on the above considerations, we are unable to estimate the average length of the vehicle schedules of their problems, which may be an indicator for the "hardness" of their master problems. It seems to us that the combinatoric of our test set is harder than the combinatoric of their test set. All these open questions give us rise to the suspicion that the artificially generated problems of Ribeiro and Soumis are of a different quality and, from a computational point of view, are far easier than our real-world problems.

6 Conclusions

Using specialized direct LP methods (and minimum-cost flow methods as our network simplex code MCF, see Löbel [1997a]) combined with a delayed column generation, it is possible to solve each problem instance of the presented test set within less than 700 seconds to optimality. Compared to that method, the decomposition approach comes off badly: Within a generous time limit of 15 times the largest LP running time, only single-depot instances with up to 700 timetabled trips, but no multiple-depot instances could be solved to optimality. A proof that a current RMP solution is fleet minimally was only possible for single-depot instances with up to 1,700 timetabled trips and for multiple-depot instances with up to 791 timetabled trips, and the gaps in the operational cost ranges from 14 % to 138 %. From a practical point of view, such gaps are unacceptable.

The theoretical advantage of a Dantzig-Wolfe decomposition – handling significantly less LP rows – does not lead to running time improvements. Quite the reverse, the RMPs of problems with more than thousand timetabled trips become too hard to be solved efficiently. From a computational point of view, the disappointing results disqualify the decomposition method for our problem.

Our work has been related to Ribeiro and Soumis [1994] who have also reported on a Dantzig-Wolfe decomposition for the MDVSP. We have pointed out that their artificially generated test set seems to be far easier than our real-world problems, which explains why we are unable to reproduce or improve their computational results.

In our investigations, we have not considered path restrictions as maximal block lengths or durations. Such restrictions make even single-depot problems $\mathcal{NP}$-hard and cannot be considered easily within an LP formulation given in terms of arc flows. Not so for a decomposition approach which defines the problem in terms of path flows and can thus handle path restrictions easily. This may be the scope of further investigations.

As a résumé of our paper, we would like to cite Ahuja, Magnanti, and Orlin [1993]: "... the Dantzig-Wolfe decomposition method has generally not proven to be an efficient method for solving the multicommodity flow problem", and they are still right concerning the multiple-depot vehicle scheduling problem.

7 Acknowledgements

This work has been supported by the German Federal Ministry of Education, Science, Research, and Technology grant no. 03-GR7ZIB -7. We are grateful to the HanseCom GmbH for providing us with real-world problems from the city of Hamburg. We are grateful to the Hamburger Hochbahn AG for their kind permission to use and publish their data. We are grateful to Bob Bixby and CPLEX Optimization, Inc., for regularly providing us access to the newest versions of CPLEX and the CPLEX Callable Library.

References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Alevras, D., Grötschel, M., and Wessäly, R. (1996). A Network Dimensioning Tool. Preprint SC 96-49, Konrad-Zuse-Zentrum für Informationstechnik Berlin. Available via ftp or WWW*.

Barnhart, C., Hane, C. A., and Vance, P. H. (1996). Integer multicommodity flow problems. In Cunningham, W. H., McCormick, S. T., and Queyranne, M., editors, *Integer Programming and Combinatorial Optimiziation*, 5th International IPCO Conference, pages 58–71, Vancouver, British Columbia, Canada.

Borndörfer, R. (1997). *Packing, Partitioning, and Covering of Sets.* PhD thesis, Technical University of Berlin. To appear in 1997.

Carraresi, P., Girardi, L., and Nonato, M. (1995). Network models, lagrangean relaxation and subgradients bundle approach in crew scheduling problems. In Daduna, Branco, and Paixão [1995].

Chvátal, V. (1980). *Linear programming.* W. H. Freeman and Company, New York.

CPLEX (1995). *Using the CPLEX Callable Library.* CPLEX Optimization, Inc., Suite 279, 930 Tahoe Blvd., Bldg 802, Incline Village, NV 89451, USA. Information available at URL `http://www.cplex.com/`.

Daduna, J. R., Branco, I., and Paixão, J. M. P., editors (1995). *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems. Springer Verlag.

Daduna, J. R. and Mojsilovic, M. (1988). Computer-aided vehicle and duty scheduling using the HOT programme system. In Daduna, J. R. and Wren, A., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 133–146.

Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, 8:101–111.

Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. (1995). *Time Constrained Routing and Scheduling*. In Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*. Elsevier Science B.V., Amsterdam. Chapter 2, pages 35–139.

Desrosiers, J. and Rousseau, J.-M. (1995). Results obtained with Crew-Opt: A column generation method for transit crew scheduling. In Daduna, Branco, and Paixão [1995].

Grötschel, M., Löbel, A., and Völker, M. (1997). Optimierung des Fahrzeugumlaufs im öffentlichen Nahverkehr. In Hoffmann, K.-H., Jäger, W., Lohmann, T., and Schunck, H., editors, *Mathematik – Schlüsseltechnologie für die Zukunft*, pages 609–624. Springer Verlag. In German available as ZIB preprint SC 96-8 via ftp or WWW*.

Löbel, A. (1996). Solving the LP relaxations of large-scale multiple-depot vehicle scheduling problems exactly. Preprint SC 96-26, Konrad-Zuse-Zentrum für Informationstechnik Berlin. Available via ftp or WWW*.

Löbel, A. (1997a). *MCF Version 1.0 – A network simplex implementation*. Available for academic use free of charge via WWW at URL `http://www.zib.de/Optimization`.

Löbel, A. (1997b). *Optimal Vehicle Scheduling in Public Transit*. PhD thesis, Technische Universität Berlin. To appear.

Löbel, A. (1997c). Recent computational developments for large-scale multiple-depot vehicle scheduling problems. Preprint SC 97-17, Konrad-Zuse-Zentrum für Informationstechnik Berlin. Available via ftp or WWW*.

Ribeiro, C. C. and Soumis, F. (1994). A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42(2):41–52.

*Available via anonymous-ftp: `ftp ftp.zib.de` ... `cd /pub/zib-publications` or at URL `http://www.zib.de/ZIBbib/Publications/`.