The Design of a Letter-Mail Transportation Network by Intelligent Techniques

Tore Grünert
Department of Operations Research, RWTH Aachen, Germany
Tore@or.rwth-aachen.de

Hans-Jürgen Sebastian
Department of Operations Research, RWTH Aachen, Germany
Sebasti@or.rwth-aachen.de

Michael Thärigen
European Laboratory for Intelligent Techniques Engineering (ELITE) Foundation, Aachen, Germany,
Michael@elite-isbt.mitgmbh.de

Abstract

Many transportation providers such as package delivery companies and postal service organizations face the problem of designing a transportation network in order to service their customers. This network must balance the requirement of on-time delivery under tight time window constraints with the goal of low-cost operations of the fleet. Until recently this task was usually performed by planners without sufficient software aid. This paper describes a decision support system (DSS) which has been designed in order to assist planners of the German postal service, the Deutsche Post AG. It helps the planners in designing improved plans which can be generated either manually or with the help of sophisticated intelligent optimization techniques. Both optimization and system design factors which influenced the design of the DSS are discussed.

1. Introduction

The reorganization of the Deutsche Post AG imposed massive structural and organizational changes. These changes strongly influence the design and operations of the logistic network. Here we will focus on the letter-mail transportation network, although freight transportation is also planned by the methods we describe in the following. The main problems related to the letter-mail transportation network are

- the design of the network, using different locations (e.g. Letter Mail Centers (LMCs), airports, hubs) and modes of transportation (e.g. airplanes, trucks),
- vehicle routing and scheduling, and
- staff scheduling.

Due to the complexity of these tasks heuristic decomposition into subsystems is necessary. The Deutsche Post AG considers two main subnetworks:

- the Global Area Transportation Network (GATN): transportation between the LMCs
- the Local Area Transportation Network (LATN): transportation within the region of a LMC

Since 1996 the ELITE foundation in cooperation with the Department of Operations Research of the Aachen Institute of Technology has been developing distributed Decision Support Systems (DSSs) to support planning and scheduling within the GATN and the LATN. The DSS for the LATN, called ISLT-BP (Integrated System Letter Transportation), is in daily use by the 84 LMCs. A prototype of the DSS for the GATN, called ISLT-NP, is currently being used at the Deutsche Post´s headquarters in Bonn and is simultaneously being extended and improved. Both DSSs are linked to each other by common interfaces.

From the Operations Research and Information System perspective the GATN is extremely interesting. The simultaneous design of the network including transportation planning and scheduling on the basis of an aggregated optimization model seems to be impossible because of the complexity involved. Therefore, we defined a number of interrelated subnetworks to solve the problems related to the GATN. However, even the subnetworks are too complex in order to solve the respective optimization problems by classical exact methods. Instead, we use Intelligent Techniques (IT) such as the Tabu Search Metaheuristic and Evolutionary Strategies to solve the corresponding optimization problems. It could be verified that a combination of such Intelligent Techniques with exact approaches like Branch and Bound works well in practice. Technically, we embedded classical mixed-integer or linear programming solvers within the systems and obtained very promising results.

This paper is organized as follows: Section 2 introduces the planning problems encountered. We then focus on network design for transportation between LMCs and describe a decomposition into smaller problems in section 3. The solution to some of these problems is found by IT introduced in section 4. Section 5 elaborates two such algorithms more detailed. Section 6 focuses on the architecture and the use of the implemented decision sup-
2. Problem Description – the Letter Mail Transportation Network of the Deutsche Post AG

The main goal of the new letter mail transportation network of the Deutsche Post AG was to improve the performance and to reduce the costs of the letter mail transportation. Therefore, after introducing the new five-digit zip codes, the following design decisions were made before the development of the DSSs was initiated:

1. The number of the LMCs was reduced from more than 1,000 to 83 plus the International LMC at the Frankfurt Airport. Many of them have been constructed at new locations.
2. The performance requirement E+1, which means “next day delivery” of all letter mail within Germany, was postulated.

As a result of these decisions the locations of the LMCs are fixed. Also, the service quality measure E+1 dominates cost considerations. More precisely, the distribution and collection of letters within the region of each of the LMCs is strongly influenced by this requirement. In addition, if one assumes that the distribution and collection of mail has to be done during day time, a time window of about 7 hours length during night for the GATN results. Therefore, the GATN has to be designed for such an overnight transportation process. This is important for the availability of vehicles and for the available transportation time (Figure 1.).

Considering the geography of Germany, it becomes obvious that about 20% of the letter mail has to be transported by aircraft, because road transportation would take too much time. Therefore, in addition to the 84 LMCs currently 14 airports have to be added to the set of locations. In Germany the minimal number of origin-destination pairs (LMC, LMC), which has to be served by air transportation in order to stay within the time window, is about 1,500.

2.1. The GATN

Figure 2 shows a screen-shot taken from the ISLT-NP decision support system. It depicts a flight from Hamburg to Nürnberg. The shaded area shows the set of origin and destination LMCs of the shipments which may be assigned to the flight. The set of LMCs which can be served by such a flight clearly depends on the take-off time. A later take-off time increases the number of LMCs which can deliver mail to the flight and decreases the number of LMCs which can receive mail at the destination airport.

We now consider the road transportation in the GATN, i.e. between the 84 LMCs. All origin-destination pairs (all together there are 84·83 = 6,972 pairs) which are not covered by air transportation have to be realized by road transportation. There are about 40 different types of vehicles available, which differ in size and speed.

The average quantity of letter mail (in metric tons) which must be transported every night is 1,500. However, the distribution of letter quantities to origin-destination pairs differs significantly. There exist pairs with a very small quantity and – on the other hand – pairs which require one or several large truckloads. A well-known strategy which
allows transportation costs to be reduced significantly is to use road-hubs in order to consolidate freight. Consequently, one of the most important problems in road-network design is to select a number of LMCs to become hubs. There is only a small number of papers which deal with integrated road-network design and vehicle scheduling using hubs [2, 30]. In addition, at each LMC and at each airport there are processing capacity constraints. These constraints reflect the bounded capacity for sorting incoming mail. Figure 3. from the ISLT-DSS shows the model which is used for this type of capacity constraint. An input level is defined for each point in time. This is the aggregated quantity of mail, which has to be available for input sorting at the given time. We are not aware of any publications which handle such inventory-routing constraints. This shows that a road-network design problem with vehicle routing and scheduling, hubs, and processing capacity constraints is extremely difficult [29].

Figure 3. Input requirements on quantity

Summarizing our GATN design problem is as follows:

**Goal:** Design of a minimum cost letter mail transportation network between 84 LMCs with next-day delivery (E+1).

**Transportation Model:** Hybrid air- and road-transportation system with processing capacity constraints at each LMC and airport. There is a heterogeneous fleet of aircraft and vehicles. Shipment from LMC to LMC by direct links or via hubs.

**Costs:** Total costs are defined by a cost model taking into account costs for air and road transportation and hub handling costs.

2.2. The LATN

The Local Area Transportation Network consists of one of the Letter Mail Centers and its assigned region with customers, mailboxes, and intermediate depots (local delivery points). There is also a heterogeneous fleet of vehicles assigned to each of the LMCs. The LATN has to carry out two main tasks:

- Input mail collection: mail is collected directly from customers and mailboxes and transported to the LMC.
- Transportation to local delivery points: after sorting the mail is transported to the local delivery points where it is picked up and delivered to the customers.

Within the LATN there are the following planning tasks:

- **Travelling Salesman Problems (TSPs):** the TSP size differs from 10 to 150 clients, they are symmetric or asymmetric and there exist time windows.
- **Vehicle Routing Problems (VRPs):** there are One- or Multi-Depot-VRPs with up to 300 customers. The fleet is heterogeneous. Most of the problem instances are asymmetric, there are time windows and additional constraints. Figure 4. depicts a screenshot from ISLT-BP, where a small VRP is solved. The routes are shown both in a tabular and a geographical form. Costs as well as temporal information are also displayed.
- **Location Routing Problems (LRPs):** there is one main depot (the LMC) and several intermediate depot locations e.g. hubs. The intermediate depot locations might be selected from a finite set of predefined alternatives. Planning scenarios are:
  - choosing the intermediate depot locations,
  - creating or deleting such locations,
  - starting with a feasible layout and designing transportation chains.
- **Multiple use of vehicles and timetabling (scheduling) of staff.**

In Figure 5. the routing on multiple levels which occurs in the LATN design and planning is illustrated [26]. These problems are well known and have been studied in the literature [18, 21]. Nevertheless there are no exact methods and related software for the problem-sizes described above. Therefore, heuristics are needed and the state of the art is characterized by Metaheuristics e.g. Tabu Search, Evolutionary Strategies, Simulated Annealing [8, 19, 23]. However, for example, for the one depot VRP with time windows, heterogeneous fleet and 300 customers there is no known Metaheuristic which could be implemented, in order to find an approximate solution in a reasonable amount of time. The situation is even worse for the MDVRP and the LRP of realistic size. It would require another paper to describe how we deal with these problems within ISLT-BP. In the following we will focus on the GATN design.
3. Decomposition of the GATN-design problem

The first idea of a heuristic decomposition of the overall problem is to define the GATN and the LATNs separately for each LMC and to describe the dependencies of these networks. In section 1, we characterized the main tasks for each of the networks.

The second idea is to split the GATN, which is much too complex for an exact approach, into subtasks or subnetworks.

In the following we explain a heuristic decomposition of the GATN-design problem into subnetworks and/or subtasks.

Before, we introduce some basic definitions.

Definition 1: Locations and Routes

(1) A **location** \( l \) is either a Letter Mail Center LMC or an airport AP, \( l \in \{\text{LMC}_1, \ldots, \text{LMC}_n; \ AP_1, \ldots, \ AP_m\} \)

In our German letter mail application we have \( n=84 \) and \( m=14 \). Each of the LMC might be used as a road hub, and one airport might serve as an air-hub.

(2) A **route** is a sequence \( l = (l_1, l_2, \ldots, l_p) \) of pairwise different locations \( l_i \), where \( p \geq 2 \). A route always starts and terminates at a LMC.

(3) A **route element** is an ordered pair \( (l_i, l_{i+1}) \) of locations belonging to a route \( l \), where \( l_{i+1} \) is the successor of \( l_i \) within \( l \).
(4) A route \( l \) which consists only of LMC-locations is called a road route.

(5) A route which contains a route element \((l_i, l_{i+1})\) where both \(l_i\) and \(l_{i+1}\) are airports is called a flight route.

(6) The duration of a route measures the (average) time needed for the transportation from \(l_i\) to \(l_{i+1}\).

(7) A route is feasible if its duration allows input processing (sorting mail) at the destination LMCr.

**Definition 2: o-d pairs**

(1) A pair of locations \((l_k, l_l)\) is called an origin-destination pair (o-d pair), if \(l_k, l_l \in \{\text{LMC}_1, ..., \text{LMC}_n\}\) and \(l_k \neq l_l\).

(2) An o-d pair \((l_k, l_l)\) is a necessary flight pair, if there does not exist a feasible road route \(l\) which starts in \(l_k\) and terminates in \(l_l\) without violating the (7 hours) time window.

(3) All other o-d pairs are road pairs. A road pair is called an optional flight pair if it is selected to be covered by a flight-route \(l\).

(4) The quantity of mail, which has to be transported between \(l_k\) and \(l_l\) of an o-d pair \((l_k, l_l)\) is referred to as the o-d quantity. It is measured in units of letter mail.

Some comments are necessary in order to explain (2) and (3) more detailed. A road route \(l\) exists, if the duration (length) of the route stays within the time window and if there exists a shipment which must be shipped from \(l_k\) to \(l_l\). However, even if such a road route exists, it perhaps makes sense to ship the freight using air transportation instead. For example, sometimes the capacity of a flight is not well used by the allocated necessary flight pairs. In such a case freight which belongs to road pairs should be re-allocated to airplanes. Therefore, we introduced the term “optional flight pairs”.

It would be possible to continue these kind of definitions in order to introduce different types of flights and road routes as well. Because we must be brief we will only introduce two additional terms:

**Definition 3: Hubs and Allocated Routes**

(1) A hub is a location \(h_0\) where an exchange of freight between vehicles can occur. Such an exchange has a minimal duration and assigned costs. In Figure 6, there is a simple example, where a hub (a LMC) is used by two road routes and four o-d pairs \((\text{LMC}_k, \text{LMC}_l), (\text{LMC}_m, \text{LMC}_n), (\text{LMC}_o, \text{LMC}_p), (\text{LMC}_q, \text{LMC}_r)\). This means that freight from LMCk and LMCm arrives at the hub and, after unloading and reloading this freight is transported to LMCq and LMCp. A route is called hub-route if one of the LMCs incorporated acts as a hub.

(2) We have to establish a relationship between the quantities \(M_{kl}\) associated with an o-d pair \((\text{LMC}_k, \text{LMC}_l)\) and how they are covered by possibly different routes.

On the other hand, a route usually carries several o-d quantities or fractions of o-d quantities. One, therefore, associates a \(n \times n\) weight-matrix \(W(l_i, l_{i+1})\) with every route element \((l_i, l_{i+1})\). The Matrix consists of weights \(0 \leq W_{kl} \leq 1\), which measure the fraction of the o-d quantity \((\text{LMC}_k, \text{LMC}_l)\), which is allocated to the route element \((l_i, l_{i+1})\). We will refer to a route with such a matrix for every route element as an allocated route.

![Figure 6. Routing via a hub](image)

Using the terminology introduced by definitions 1, 2, and 3, we can describe the idea of decomposition of the design task of the GATN into subtasks and subnetworks (Figure 7.). The set of origin-destination pairs (level 1: o-d pairs) is divided into the set of necessary flight pairs and the set of road pairs by a preprocessing procedure using data from the geographical and the economical database. A subset of road pairs is selected to be covered by air transportation. This results in a decomposition into a set of flight routes and a set of road routes (level 2: routes). Within the flight route part a further decomposition is done by considering the “Flight Network Planning and Scheduling” problem and the “Ground Feeding Networks” for each of the airports \(\text{AP}_j\) (level 3: freight and vehicles). Within the road route subsystem (representation of road routes including hubs), a two step decomposition is performed. In a first step “Road Network Planning”, we have to determine all allocated hub routes and all allocated direct routes. This establishes a fixed schedule of freight flow in the system. In order to cover this flow, vehicle scheduling problems have to be solved. This is done separately (because of complexity reasons) for the allocated hub-routes and the allocated direct routes.
The heuristic decomposition procedure leads to five subnetworks:

1. Flight Network Planning and Scheduling
2. Ground Feeding Network Planning
3. Road Route Planning
4. Vehicle Scheduling for Hub-Routes
5. Vehicle Scheduling for Direct-Routes

Each of the design tasks is modeled as a combinatorial optimization problem [15]. However, although we did decompose and focus on “smaller” problems, these combinatorial problems are still too complex to be solved by exact methods in a reasonable time. This is the reason why heuristics are needed to construct “good” feasible solutions in acceptable time. Some of the most powerful Metaheuristics for the combinatorial type of optimization problems we are dealing with in (1) - (5) are Intelligent Techniques. We found that population-based methods like “Evolutionary Algorithms” and “Tabu Search” Metaheuristics are the most promising approaches. Furthermore, we found that a combination of the Intelligent Techniques with more traditional techniques from the Operations Research field like mixed-integer programming algorithms, and Branch and Bound seem to be the key to solve the subnetwork problems successfully. We will give two such examples in section 5.

4.1. Tabu Search and Evolutionary Algorithms

The Tabu Search Metaheuristic, which is based on ideas from network optimization and Artificial Intelligence, was first published by [10]. It uses flexible forms of memory in order to guide search processes dynamically. This enables local search algorithms to escape from local minima and to exploit information which has been gathered during the search.

Tabu Search in its basic version relies on a so-called tabu list. The tabu list stores attributes of a move, i.e. the transition mechanism which enables the search algorithm to move from a solution to its neighbor solution. Moves with these attributes are then forbidden during the next iterations. The number of such iterations is called the tabu tenure. Tabu Search also allows the tabu status to be revoked if an aspiration criterion is met. These principles define the basic, so-called simple or short-term memory, form of Tabu Search.

In addition to the basic algorithm, Tabu Search uses longer-term memory concepts. These include intensification and diversification. Intensification concepts are based on methods which identify promising regions of the...
search space and then search these region more thoroughly. Diversification strategies exploit information about the search history and guide the search towards yet unexplored regions. Both intensification and diversification are often based on counting principles.

The strength – and the weakness – of Tabu Search is its flexibility with respect to implementation. It contains a wide variety of ingredients and parameters, which have to be properly adjusted if the algorithm is to work well. We will come back to this issue when presenting computational results. A more detailed exploration of the Tabu Search Metaheuristic and an overview over applications can be found in [13].

Evolutionary Strategies and the related algorithms provide a flexible approach to generate good feasible solutions to highly constrained problems without easily exploitable mathematical structures [14, 16, 22, 25, 27, 28]. They produce a finite set of solutions, called a population of individuals, where in our case the individuals are e.g. route networks consisting of allocated hub routes and direct links. One of the most important decisions is the representation of the individuals. After evaluation of each individual of a population by a so-called fitness function and after applying a selection scheme to the current population a new population is generated by different genetic operators. The most important operators are recombination and mutation. If the representation of an individual is such that feasibility is guaranteed, then the operators should maintain feasibility. Otherwise, difficult-to-design repair algorithms become necessary. Evolutionary strategies aim at imitating the “survival of the fittest principle”, which is observed in nature. By simulating the evolution process on the computer, it is hoped that good individuals are generated after the computation of several hundred generations of populations within a short time.

### 4.2. Application of Intelligent Techniques (IT) to the Subnetwork Design Problems – An Overview

In Table 1, we give an overview on subnetwork design tasks and the related IT we developed.

The type of model we used for the subnetworks (2) and (4) is also population based because the basic idea of a set covering approach is to pre-compute a large finite set of partial solutions. The set covering model then selects a set of partial solutions in order to obtain a minimum cost feasible solution. More details about this approach are given in [17]. In general, it is somehow philosophical to distinguish between “simple Metaheuristics” and those which might also be considered to be IT. As a criterion for a Metaheuristic to be an IT we consider here the ability to imitate some aspects of intelligent behavior in nature.

We focus, therefore, on Tabu Search and Evolutionary Algorithms in the following. In the next section we briefly describe the main ideas of the ITs we developed for the Direct Flight Problem and for the Road Route Problem.

### 5. Two examples of IT for Subnetwork Design

In this section we describe a Tabu Search algorithm for the direct flight network design problem and an Evolutionary Algorithm for Road Route Planning Problem.

#### 5.1. A Tabu Search approach for the Direct Flight Network Design Problem

After briefly describing the modeling approach we have taken in the case of the Direct Flight Network Design Problem we outline the Tabu Search algorithm and present computational results.

<table>
<thead>
<tr>
<th>Subnetwork Design tasks</th>
<th>Intelligent Techniques / Metaheuristic / Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Flight Network Planning and Scheduling</td>
<td>Tabu Search with embedded branch and bound</td>
</tr>
<tr>
<td>(a) use of direct flights only</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>(b) use of one air hub: hub flights and direct flights</td>
<td></td>
</tr>
<tr>
<td>(2) Ground Feeding Network Planning for each Airport</td>
<td>Set Covering model solved by a heuristic</td>
</tr>
<tr>
<td>(3) Road Route Planning</td>
<td>Evolutionary algorithm with an embedded LP Solver</td>
</tr>
<tr>
<td>Allocated Hub and Direct Routes</td>
<td></td>
</tr>
<tr>
<td>(5) Vehicle Scheduling for Hub Routes</td>
<td>Set Covering model solved by a heuristic</td>
</tr>
<tr>
<td>(6) Vehicle Scheduling for Direct Routes</td>
<td>Saving-type of heuristic</td>
</tr>
</tbody>
</table>
5.1.1. A brief description of the model

We start with a brief description of the model for the Direct Flight Problem. The basis of our model is the observation that only a finite number of take-offs have to be considered at each airport. The take-off time can be derived from the time it takes to carry the letter mail from the LMC to the airport plus the time for loading. Figure 8. depicts the situation for a single airport AP. The arcs give the sum of transportation and loading time. For example, the letter mail from LMC 13 cannot take-off before 3.0 hours, assuming that all times begin at a given hour (for example 9 p.m.). It is clearly sufficient to consider the take-off times 1.5h, 2.0h, 2.5h and 3.0h in this example. As a consequence of this observation only a discrete and finite number of flights has to be considered.

Figure 8. Possible take-off times at an airport

The model we used can be considered as an integer fixed-charge network flow model with side constraints [12]. By integer fixed-charge we mean that a fixed-charge arc can be used an integer number of times, and that the fixed-costs are charged every time one complete unit of the arc is used. The model can easily be transformed to a standard (binary) fixed-charge model by using binary coding of the integer variables. This poses no problems in practice since the integer variables have small upper bounds. Figure 9. depicts the structure on the fixed-charge network for a small fraction of the model. The first flight arc models a flight with a B737 from Hamburg to Frankfurt am Main with take-off at 23:00. It can carry the o-d pair (LMC 23, LMC 61) and (LMC 30, LMC 64). The o-d quantity from LMC 23 to 61 is 20 in units of letter mail.

Furthermore, the direct flight model can be considered as a capacitated warehouse (facility, plant) location problem (CWLP) with side constraints. CWLPs have been extensively studied in literature [3, 6, 9, 20]. However, we must deal with problems having up to 20,000 potential warehouses. Problems of this size have not been considered so far.

5.1.2. The Algorithm

We use a two-phase approach when solving the direct flight problem. In the first phase, we construct an initial feasible solution by a greedy algorithm [5]. This solution is then improved by the hybrid Tabu Search and Branch and Bound algorithm. The core of the algorithm is the optimal solution to a restricted MIP with a number of free integer variables by branch and bound. The Tabu Search part guides the selection of the free variables by candidate lists and attributive memory.

Each solution to the direct flight problem contains a set of active flights, \( F_A \). All other flights are closed, denoted by \( F_U \). Thus, \( F_A \) is a small set whereas \( F_U \) is a (very) large set.

Suppose that a feasible solution to the direct flight problem with an optimal assignment of o-d quantities to the active flights is available. An improvement of the current solution requires at least one active \( f \in F_A \) flight to be closed, i.e. \( f \in F_U \). If the demand covered by the closed flight cannot be covered entirely by other active flights, then a closed flight has to become active. Such a replacement process can be generalized as follows. A replacement of \( k \) flights from the set of active flights by \( l \) inactive flights is called a (k,l)-replacement.

The neighborhood of the current solution consists of all (k,l)-replacements for a given value of \( k \) and \( l \). Our aim is to efficiently search the neighborhood for a restricted number of good and feasible replacements. The variables of the flights corresponding to these replacements are free in the Branch and Bound phase. For the purpose of computational efficiency it was decided to work only with replacements \( k \leq 2, l \leq 2 \). But even the number of possible (2,2)-replacements grows quadratically with the number of possible flights. This makes a direct evaluation of the entire neighborhood impractical for larger...
instances. It is, therefore, necessary to restrict the number of possible replacements by additional requirements. The respective techniques are discussed in detail in [4, 5].

The core of our approach is the construction of a decision tree for neighborhood evaluation. This neighborhood evaluation routine can be combined with a candidate list approach, e.g. [11, 13]. Given a threshold value on the potential saving of a replacement, a counter is activated the first time this threshold is satisfied. Then an additional number of (plus) replacements are considered and given to the Branch and Bound phase. This approach reduces the computational effort significantly.

In contrast to most Tabu Search implementations a selected replacement (or, more generally, move) is not necessarily performed due to the Branch and Bound phase. The Tabu status of a move is consequently based on the result of the Branch and Bound phase and not on the selection of a possible replacement. We thus designate all flights, which are closed during the Branch and Bound phase, tabu. The tabu status can be revoked if the saving associated with a replacement exceeds an aspiration threshold (aspiration by objective).

A fundamental difference between the approach employed here and implementations of simple Tabu Search, see [13], is that uphill moves are not possible due to the optimization phase. The tabu status is used to guide the neighborhood search instead. Consequently, for a restricted neighborhood size the procedure terminates in a local optimum. Instead of performing uphill moves we have chosen to directly diversify the solution based on search history information. We believe this approach to be more effective since the selection of uphill moves is quite arbitrary and a diversification routine is important for obtaining improved solutions.

Our diversification routine is based on frequency information: we simply count the number of iterations a flight has been active in the solution. All currently active flights are sorted according to this residence frequency and the m flights with the highest values are closed. The corresponding o-d pairs have to be covered by new flights. These new flights are determined by a greedy algorithm. The standard search routine is then re-started with the new solution and a tabu status on the newly activated flights. The aim of this tabu status is to avoid the immediate reversal of the diversification step by the standard search.

An important design issue which has not been considered so far is the management of the tabu lists. Moreover, the design of a stopping criterion has to be examined. We discuss different strategies in conjunction with the computational results in section 5.1.3.

The Tabu Search algorithm for the direct flight problem is summarized below. It requires a feasible solution as an input.

Algorithm 1: Tabu Search

1. (Local Search). WHILE the entire neighborhood has not been searched AND stopping criterion not satisfied DO
2.1 (Determine Frequency Information). Sort the active flights according to decreasing residence frequency count.
2.2 (Close some flights). Pick the m active flights with the highest count and close them.
2.3 (Call greedy algorithm). Compute the reduced problem arising from closing the flights. Solve the problem by the greedy heuristic.
2.4 (Tabu assignment). Designate all flights which were introduced by the greedy algorithm tabu.

5.1.3. Computational Results

The main objective of the computational tests was to evaluate how different parameter setting of the algorithm influence the quality of the solution. We are currently not able to judge the quality of the heuristic solutions with respect to a lower bound or an optimal solution. We can only rely on comparisons with existing solutions. Nineteen test problems were defined. The problems were of different size, resembling different load scenarios for the night airmail network within Germany. We did not generate any random problems since we believe that they would be too far from the planning situation encountered in practice.

The 19 problems have 485, 499, 572, 666, 999, 1,499 and 3,004 flight pairs respectively. Five different planning scenarios correspond to each of the problems with 485, 572 and 666 flight pairs. Table 2, gives the number of flights and the number of possible assignments of o-d quantities to flights in the reduced problems, i.e. after preprocessing phase for some problem sizes.

The Tabu Search algorithm has four parameters, which can be controlled by the user. These are: the value of plus in the Aspiration Plus strategy (the threshold is set to zero, i.e. no costs are saved) (plus), the number of free variables in the Branch and Bound phase (free), the tabu
tenure (tenure), and the number of closed flights during diversification (m). Table 4. shows the different parameter values which were used during the test.

The stopping criterion of the local search phase was based on the candidate list. Whenever no non-improving move could be found after scanning the entire neighborhood, diversification was invoked.

The algorithm was programmed in C++ using the Microsoft Visual C++ compiler, version 2.2, under Microsoft Windows NT, version 3.51. The tests were conducted on a Dual Pentium PC with 266 MHz and 256 MB RAM. The compiler option “target” was set to “release”.

### Computational Times

The computational times were measured for all possible parameter combinations. Table 3. gives the average time for 1,000 iterations of the algorithm and the average time it took until the best solution was found. All times are given in CPU-minutes.

The data suggest that the computation time grows about quadratically with an increasing size of the problem, although the sample size is too small to draw any statistically significant conclusions. The two most time-consuming tasks are the generation of possible replacements and the Branch and Bound phase. The quadratic growth of the time for the replacement search can be expected since the size of the (2,2) neighborhood grows quadratically with the size of the problem. However, the time required for the Branch and Bound phase was difficult to predict. Its portion falls from about 30% - depending on the parameter settings – for the small problems to less than 5% for the largest problems. This could be expected since the size of the Branch and Bound optimization problem does not grow significantly with an increasing size of the problem.

### Quality of the Solutions

This section focuses on the quality of the solutions as a function of the parameter settings. The same parameter settings as in the former section were used for the problems with less than 999 origin-destination pairs. Table 5. gives the average deviation from the best found solution.

Larger values of both plus and free have a positive influence on the quality of the solutions. This follows from the fact that the Branch and Bound phase is able to take advantage of combining the free variables so that larger gains result. The values of these parameters cannot, however, be increased much further due to the excessive computational effort this would require.

The number of flights which are closed in a diversification step does not seem to influence the quality of the solutions significantly. This suggests that closing one flight diversifies the solution sufficiently.

It was found that the tabu tenure has the most significant impact on the quality of the solutions. The best solutions were obtained for tabu tenures greater than five confirming the necessity of this active memory component. However, the performance remains relatively stable for tenures above this value.

### Table 2. Problem characteristic

<table>
<thead>
<tr>
<th>problems</th>
<th>499</th>
<th>999</th>
<th>1,499</th>
<th>3,004</th>
</tr>
</thead>
<tbody>
<tr>
<td># flights</td>
<td>733</td>
<td>2,533</td>
<td>4,282</td>
<td>11,839</td>
</tr>
<tr>
<td># assignments</td>
<td>24,078</td>
<td>126,738</td>
<td>288,211</td>
<td>1,188,987</td>
</tr>
</tbody>
</table>

### Table 3. Average computational times in CPU-minutes

<table>
<thead>
<tr>
<th>problems</th>
<th>485</th>
<th>499</th>
<th>572</th>
<th>666</th>
<th>999</th>
<th>1,499</th>
<th>3,004</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000 iterations</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>39</td>
<td>164</td>
<td>2.151</td>
</tr>
<tr>
<td>best solution</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>17</td>
<td>97</td>
<td>1.366</td>
</tr>
</tbody>
</table>

### Table 4. Parameter settings

<table>
<thead>
<tr>
<th>parameter</th>
<th>plus</th>
<th>free</th>
<th>tenure</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>values</td>
<td>1, 2, 5, 10</td>
<td>1, 2, 5, 10</td>
<td>0, 1, 2, 5, 10, 15</td>
<td>1, 2, 3</td>
</tr>
</tbody>
</table>
Table 5. Average deviation from the best found solution in percent for different parameters

<table>
<thead>
<tr>
<th>parameters</th>
<th>plus</th>
<th>free</th>
<th>tenure</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>values</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>485 - I</td>
<td>8.2</td>
<td>6.9</td>
<td>6.4</td>
<td>5.8</td>
</tr>
<tr>
<td>485 - II</td>
<td>10.7</td>
<td>7.2</td>
<td>5.8</td>
<td>6.3</td>
</tr>
<tr>
<td>485 - III</td>
<td>9.6</td>
<td>6.7</td>
<td>6.1</td>
<td>5.9</td>
</tr>
<tr>
<td>485 - IV</td>
<td>5.9</td>
<td>4.8</td>
<td>4.6</td>
<td>4.2</td>
</tr>
<tr>
<td>485 - V</td>
<td>1.3</td>
<td>1.1</td>
<td>0.9</td>
<td>1.1</td>
</tr>
<tr>
<td>499</td>
<td>5.2</td>
<td>4.5</td>
<td>4.1</td>
<td>4.4</td>
</tr>
<tr>
<td>572-I</td>
<td>6.5</td>
<td>5.2</td>
<td>5.6</td>
<td>5.1</td>
</tr>
<tr>
<td>572-II</td>
<td>6.7</td>
<td>6.9</td>
<td>6.4</td>
<td>6</td>
</tr>
<tr>
<td>572-III</td>
<td>7.1</td>
<td>6.4</td>
<td>5.4</td>
<td>4.8</td>
</tr>
<tr>
<td>572-IV</td>
<td>13.4</td>
<td>11.1</td>
<td>10.6</td>
<td>10.4</td>
</tr>
<tr>
<td>572-V</td>
<td>15.1</td>
<td>8.7</td>
<td>2.4</td>
<td>2.1</td>
</tr>
<tr>
<td>666-I</td>
<td>3.2</td>
<td>7.7</td>
<td>19</td>
<td>6.8</td>
</tr>
<tr>
<td>666-II</td>
<td>9.1</td>
<td>7.9</td>
<td>6.9</td>
<td>6.9</td>
</tr>
<tr>
<td>666-III</td>
<td>8.1</td>
<td>6.2</td>
<td>4.5</td>
<td>5</td>
</tr>
<tr>
<td>666-IV</td>
<td>4.1</td>
<td>3.8</td>
<td>3.2</td>
<td>3</td>
</tr>
<tr>
<td>666-I</td>
<td>7.2</td>
<td>6.1</td>
<td>5.2</td>
<td>4.2</td>
</tr>
<tr>
<td>999</td>
<td>7.8</td>
<td>4.4</td>
<td>3.2</td>
<td>2.4</td>
</tr>
</tbody>
</table>

5.2. An Evolutionary Algorithm for the Road Network Design Problem

We now consider the Road Network Design Problem. The goal is to determine a minimum cost set of allocated hub routes and allocated direct routes such that all road pairs which have to be served by road transportation are covered. The hubs are a pre-defined subset of LMCs. A hub is characterized by a capacity and the required time for exchanging the vehicles loads. Unfortunately, transportation costs are not uniquely determined by the allocated hub or direct routes. This is the case only if an assignment of a vehicle schedule to each allocated route has been made.

Therefore, in order to solve the Road Network Design Problem, we first introduce trips, where a trip means the assignment of a departure time and a vehicle type to a given route. Then, a set of such trips which meets particular requirements is generated and after the construction of such a set the allocated hub and direct routes are computed by assigning freight to the trips. Finally, only the allocated routes are kept. The vehicle scheduling is performed within the subsequent network, see Figure 7.

We chose this decomposition in order to first compute the flow of freight through the network. Then, in the second phase, vehicle schedules are constructed in order to cover the freight (letter mail) flow.

Coming back to the construction process of trips and allocated routes we use the following basic algorithm:

**Step 1:** Construction of a set of hub trips with

a) trips from a LMC to a hub:
   - departure at the earliest possible time
   - arrival time equals departure plus driving time

b) trips from a hub to a LMC:
   - arrival at the LMC such that the processing capacity constraints are guaranteed to be feasible
   - departure at the hub equals arrival time at the LMC minus driving time from the hub to the LMC

**Step 2:** allocation of the constructed hub trips by fractions of letter mail, which must be shipped on the respective o-d pair.

**Step 3:** all fractions of letter mail which cannot be allocated to hub trips are allocated to direct trips.

**Step 4:** construction of allocated hub and direct routes:

a) if a fraction of letter mail is allocated to a hub trip, then the corresponding allocated hub route is generated.

b) if a fraction of letter mail is not allocated to a hub trip, then a candidate allocated direct route is constructed.

This approach has the advantage that each allocated route construction in step 4 is evaluated by costs. However, due to the decomposition the vehicle schedules are unknown and the real costs are, therefore, not known at this stage. We, therefore, only use the costs of the route depending on the time and locations – and do not consider how the vehicle returns to its origin.
The Evolutionary Algorithm

The quality of the designed road network generated by the algorithm above depends, mainly on the quality of the set of generated hub trips within step 1. In order to generate such a set of “good hub trips” we use an Evolutionary Algorithm (EA) which follows the principles described in section 3.1. The interpretation of the general concepts of an EA is as follows:

1. **Individual**: set of hub trips (using the complete algorithm step 1 – step 4, the respective hub and direct routes, with other words a respective road route network, is computed from the set of hub trips) and, therefore, a complete allocated road route network.

2. **Fitness**: the fitness of an individual is measured by the costs

3. **Recombination**: construction of a set of hub trips from two other sets of hub trips (parents)

4. **Mutation**: random change of one or multiple trips of one individual (one complete road route network)

The evolution starts with a set of randomly generated individuals, where each individual is based on a set of randomly generated hub trips. Then, each individual is completed by applying steps 2-4 of the basic algorithm. The individuals of the next generation are generated by recombination of a subset of individuals of the parent generation. A small mutation-probability is given for each individual.

This shows that recombination and mutation are the specific methods of the algorithm. The solution process is standard based on the fitness-evaluation.

Recombination

The goal of recombination is to maintain good (cost-effective) trips between locations. Therefore we focus on a location when dealing with recombination (Figure 10. shows three locations (LMCs). One LMC serves as a hub and the two other LMCs as ordinary locations. There are two trips from LMC A to the hub and one trip from the hub to LMC B).

![Figure 10. Trips, routes, and hubs](image)

Within each of the parents we compute subsets of trips characterized by the same origin and the same destination. Figure 11. shows two such subsets between two LMCs and one hub (which is also a LMC).

![Figure 11. Subsets](image)

If there does not exist a trip between a location and a hub, an empty subset is created (Figure 12.).

![Figure 12. Empty subset](image)

The solution process is standard based on the fitness-evaluation. The interpretation of the general concepts of an EA is as follows:

1. **Individual**: set of hub trips (using the complete algorithm step 1 – step 4, the respective hub and direct routes, with other words a respective road route network, is computed from the set of hub trips) and, therefore, a complete allocated road route network.

2. **Fitness**: the fitness of an individual is measured by the costs

3. **Recombination**: construction of a set of hub trips from two other sets of hub trips (parents)

4. **Mutation**: random change of one or multiple trips of one individual (one complete road route network)

The evolution starts with a set of randomly generated individuals, where each individual is based on a set of randomly generated hub trips. Then, each individual is completed by applying steps 2-4 of the basic algorithm. The individuals of the next generation are generated by recombination of a subset of individuals of the parent generation. A small mutation-probability is given for each individual.

This shows that recombination and mutation are the specific methods of the algorithm. The solution process is standard based on the fitness-evaluation.

Recombination

The goal of recombination is to maintain good (cost-effective) trips between locations. Therefore we focus on a location when dealing with recombination (Figure 10. shows three locations (LMCs). One LMC serves as a hub and the two other LMCs as ordinary locations. There are two trips from LMC A to the hub and one trip from the hub to LMC B).

![Figure 10. Trips, routes, and hubs](image)

Within each of the parents we compute subsets of trips characterized by the same origin and the same destination. Figure 11. shows two such subsets between two LMCs and one hub (which is also a LMC).

![Figure 11. Subsets](image)

If there does not exist a trip between a location and a hub, an empty subset is created (Figure 12.).

![Figure 12. Empty subset](image)
Figure 13. shows an example of the recombination operator. For each origin and destination the respective subset of trips of a randomly chosen parent is copied to the successor.

The recombination process is completed by assigning fractions of letter mail to the hub trips. There exist other types of recombination operators, but we found the one described here to be most effective.

**Mutation**

The following mutation operators are used:

1. **Changing the vehicle type of a hub trip** This changes the capacity and influences the assignment of letter mail to the respective road route.

2. **Creation or deletion of a hub trip.** If a new trip is the first one to the hub, then this mutation operator starts the hub operation. If there is only one trip to the hub and the mutation deletes this trip, then the hub is automatically closed.

3. **Deletion of all hub trips from or to a hub** This option closes the hub.

The probability of mutation was set to approximately 0.5%.

In our computational experiment we used the following parameters:

- 100 individuals per generation.
- These 100 individuals produce 200 offspring (after selection of parents) by recombination and mutation.
- The 100 fittest individuals are copied and become the 100 individuals of the next generation. The other 100 individuals die.

The results we obtained from this approach are encouraging. Smaller fractions of letter mail are routed via hubs and bigger amounts of mail are shipped directly without consolidation. Not all potential hubs are used in the solutions which are generated by the algorithm. Instead, hubs which have geographical advantages are selected and their capacity is fully used. A drawback is the large computation times required in order to get very good results.

The Figures 16. and 17. depict a typical run of the Evolutionary Algorithm. It can be seen that the exchanged hub quantities increase strongly for some hubs during the first iterations. Moreover, the number of routes decreases due to the consolidation effect of the hubs. This behavior can often be observed. It is seen that large quantities are routed via a subset of the possible hubs in order to obtain a maximal consolidation effect.

![Figure 15. Deletion of all hub trips](image)

![Figure 16. Costs for hub and direct routes over generations](image)
6. Using the ISLT Decision Support System

The ISLT System is part of a larger information system in use or development at the Deutsche Post AG. Recall that the ISLT-NP system is a global planning system. It is, therefore, in use by the corresponding planning staff at the Deutsche Post AG’s headquarter. The local ISLT-BP system is installed at the LMCs. All systems are linked by an intranet. Communication is possible through pre-specified interfaces. It is, in principle, possible to share the data with other applications in order to obtain a more integrated system. This is indeed desirable and possible. However, due to performance considerations, all data exchange is done via the interfaces and the databases are controlled by the respective applications.

The hardware and software architecture is similar in both ISLT applications. A UNIX-based server is used for the database containing all relevant planning data. This includes locations, capacities, costs, vehicles, aircraft, routes etc. The server is also used for the geographical information system (GIS) data base. It contains detailed information about the geographical properties of the locations, possible road connections, driving times etc.

Users (clients) interact with the planning systems via a graphical user interface (GUI) as shown by Figure 18. All clients are Windows NT based systems running on PCs. The GUI provides dialogues as list boxes etc. as well as geographical input and output facilities which display planning information graphically.

For reasons of user-acceptance it is important that the system supports manual planning activities. The corresponding modules allow feasibility checks to be performed, capacity and slack capacity values to be calculated, temporal information to be displayed etc. Due to the large number of data, such calculations are extremely time consuming tasks if they are done manually – or even by (very large) spreadsheets. In order to further support the planner, analyses tools are provided. These tools analyze existing plans and suggest possible modifications,
e.g. alternative routes and vehicles. Eventually, optimization algorithms can be used in order to obtain alternative solutions for the networks described above. Again, the results can be displayed, analyzed, and modified with the methods already described.

7. Conclusions

In this paper we have introduced two decisions support systems, which are used for operational and strategic planning at the Deutsche Post AG. Both systems use modern graphical oriented user interfaces with geographical databases and output.

A key factor for successful implementation and user-acceptance of the system was the support of manual planning, the design of modules for computer-assisted analyses of the plans, and a decomposition of the planning task into smaller problems.

However, even those smaller problems were much too large to be handled by conventional optimization techniques. Here we briefly described two such algorithms. We, therefore, developed a number of heuristic algorithms based on Intelligent Techniques. The first algorithm combines Tabu Search, which uses ideas from Artificial Intelligence and network optimization, with the conventional Branch and Bound tree search method. This enables us to solve very large instances of the direct flight network design problem. The second algorithm is population-based and uses ideas from natural evolution. It can be considered an Evolutionary Strategy and represents entire routing plans as individuals. Recombination and mutation work on different route locations and modify existing routes in order to establish hub processes in the network.

We believe that the successful implementation of the optimization algorithms is due to the mixture of traditional OR techniques with newer Intelligent Techniques in a pure or hybrid manner. This approach has also been used by other researchers, for example, by combining local search and constraint programming algorithms for vehicle routing [1, 24].

Future research will focus on lower bound computations which allow us to judge the quality of the solutions and further improvements of the existing algorithms.

Acknowledgements

The authors are grateful to the Deutsche Post AG in particular to the director of the transportation department J. Weith for the possibility to engage in this interesting project. We acknowledge the constructive collaboration with the project-leaders Dr. R. Kuchem and M. Katz of the German Post AG.

Finally, we like to thank in particular K. Büdenbender, S. Irnich, C. Kranz and T. Kriese, who did an excellent job in inventing new algorithms and implementing the DSS.
8. References


