Florian Martin and Vera Hemmelmayr and Tina Wakolbinger

Integrated service selection, pricing and fullfillment planning for express parcel carriers - Enriching service network design with customer choice and endogenous delivery time restrictions

Paper

Integrated service selection, pricing and fulfillment planning for express parcel carriers — Enriching service network design with customer choice and endogenous delivery time restrictions

Florian Martin[a], Vera Hemmelmayr[b], and Tina Wakolbinger[b]

[a]Department of Socioeconomics, Vienna University of Economics and Business

[b]Department of Global Business and Trade, Vienna University of Economics and Business

Abstract

Express parcel carriers offer a wide range of guaranteed delivery times in order to separate customers who value quick delivery from those that are less time but more price sensitive. Such segmentation, however, adds a whole new layer of complexity to the task of optimizing the logistics operations. While many sophisticated models have been developed to assist network planners in minimizing costs, few approaches account for the interplay between service pricing, customer decisions and the associated restrictions in the distribution process. This paper attempts to fill this research gap by introducing a heuristic solution approach that simultaneously determines the ideal set of services, the associated pricing and the fulfillment plan in order to maximize profit. By integrating revenue management techniques into vehicle routing and fleet planning, we derive a new type of formulation called service selection, pricing and fulfillment problem (SSPFP). It combines a multi-product pricing problem with a cycle-based service network design formulation. In order derive good-quality solutions for realistically-sized instances we use an asynchronous parallel genetic algorithm and follow the intuition that small changes to prices and customer assignments cause minor changes in the distribution process. We thus base every new solution on the most similar already evaluated fulfillment plan. This adapted initial solution is then iteratively improved by a newly-developed route-pattern exchange heuristic. The performance of the developed algorithm is demonstrated on a number of randomly created test instances and is compared to the solutions of a commercial MIP-solver.

Keywords: asdlkfjasdlkfj

JEL Classifications: C2, C81, C82, C83, D3

1 Introduction

Driven by a booming e-commerce sector express parcel carriers have experienced rapid growth in transportation volumes. In 2016, the two dominant integrated express carriers UPS and FedEx had a combined U.S. domestic package operation of 26.5 million packages per day (FedEx, 2016; UPS, 2016). Given the dynamic development of online retailing with compound annual growth rates of 12.8% over the last decade (US Census Bureau, 2009, 2017) efforts have focused on increasing the efficiency of the underlying physical distribution networks. Service network design, i.e. tactical planning of operations based on optimized consolidation and routing of shipments and vehicles, has appeared particularly promising for improving capacity utilization of resource-restricted logistics networks.

Still, the ever growing number of shipments is not the only challenge faced by the package delivery industry. Tight competition together with customer's desire to receive their goods ever more quickly have encouraged express carriers to offer a wide range of transportation services. With carriers nowadays allowing their customers to choose from several guaranteed delivery times (e.g. overnight, two-day, etc.), revenue management is playing an increasingly prominent role in logistics operations. More granular service segmentation, however, adds a whole new layer of complexity to the task of optimizing the logistics operations. This is especially true once the optimization models account for the consequences of carrier's pricing decisions. Interestingly, while many sophisticated models have been developed to assist network planners in minimizing costs, few approaches account for the interplay between service pricing, customer decisions and the associated restrictions in the distribution process.

As such this paper attempts to add to this research gap by introducing a solution approach that simultaneously determines the ideal service offering as well as the cost-optimal fulfillment plan. We call this new approach the service selection, pricing and fulfillment problem (SSPFP). From a modeling perspective it combines a service network design problem (SNDP) with the optimization of a product line differentiated by guaranteed delivery time. Each of these models belongs to the class of NP-hard (check) optimization problems making a combination of them particularly challenging to solve. We tackle this problem by employing a heuristic procedure based on a genetic algorithm together with a newly-developed route pattern exchange heuristic. We show that the proposed solution framework delivers good-quality solutions within reasonable amounts of time.

To the best of our knowledge this is the first optimization approach that jointly determines the optimal service segmentation and the associated profit maximizing prices while accounting for time restrictions of shipments and the routing of vehicles. Consequently our first research contribution is the SSPFP model itself. Using this model we explore how the service segmentation and associated pricing affects the operational costs in a multi-modal package distribution network. The relevance of joint revenue and cost optimization within complex distribution networks is highlighted by a comparison between our integrated solution approach and a more conventional sequential optimization. Finally we demonstrate the flexibility of the model by solving it for various test instances and show how it can be tailored for further scientific and practical analysis.

As our topic is interweaving the concepts of vehicle routing and revenue management, the structure of the paper is aimed at making this topic accessible to both groups of readers. We begin with highlighting some previous work that involved integrating revenue and capacity management and discuss how these studies are related to our approach. We then provide some background on product segmentation and pricing and introduce our product line optimization model. Subsequently we turn to the literature on and formulation of service network design models. This then enables us to combine the two streams of research and derive our SSPFP model. Finally we conclude with a computational study and provide insight in the complex interrelation of customer decisions and fulfillment efficiency in the context of parcel logistics.

2 Background

Express package carriers have a long tradition of using operations research methods to improve their competitive position. Confronted with a rising number of shipments and an ever more demanding customer base, practitioners as well as researchers have put great effort into improving capacity utilization of resource-restricted logistics networks. The fulfillment side of consolidated freight services is often formulated as multi-commodity fixed-charge network design models, commonly referred to as service network design problems (SNDP) once they focus on the tactical planning of operations (Crainic, 2000; Crainic and Laporte, 1997). These models have been adapted to many applications in transport, including truck, train, container and airline services. As network design formulations can be extended rather easily, researchers have added many different features, like asset balancing contraints (Andersen et al., 2009), level of service requirements (Jarrah et al., 2009) or stochastic demand (Hoff et al., 2010).

With respect to the primary target industry of this study, i.e. express parcel delivery, there is also a small set of papers dealing explicitly with applications of service network design models for this transport sector[1]. Most notably in a series of papers Barnhart et al. (2002) (see also Barnhart and Schneur (1996) and Kim et al. (1999)) analyze the Express Shipment Delivery Problem of UPS. As the network size of an integrated express parcel carrier, like UPS, is typically huge, the main focus of these papers is on managing the complexity of the associated space-time graph in order to reduce the number of variables and constraints. These efforts eventually culminate in the works of Armacost et al. (2002, 2004) who introduce the so-called composite variable formulation of SNDPs. While this formulation allows for realistically-sized multi-modal express networks it prohibits some of the key features of the integrated, profit-maximizing models needed for adequate RM. This is due to

[1]For more general reviews on service network design the reader is referred to Wieberneit (2008) and **Martin2019b**.

the fact that flow and resource variables get reduced to just one set of composite variables. While this is computationally very efficient the lack of explicit flow variables greatly limits the possibilities to enforce any kind of delivery time requirements or similar path-based restrictions which are key to implementing RM-components.

A study allowing for two levels of service within integrated logistics networks is Smilowitz et al. (2003). They are able to distinguish between express and deferred delivery items and present a solution approach based on column generation for solving the LP-relaxation of the problem and then applying a number of rounding heuristics to derive good-quality integer solutions. In a subsequent paper Smilowitz and Daganzo (2007) circumvent the combinatorial difficulties of large-scale integer programs by using continuum approximation of a discrete network design problem instead. The authors argue that they can derive good quality approximations for fleet size and number of required terminals, provided that the network is sufficiently large. However, they also admit that evaluation of solution quality is problematic as the necessary scale of such a network limits comparisons with discrete approaches.

It is evident from the presented studies that researchers have made significant advances in making the distribution process within multi-modal LTL-networks more efficient. However, all of these approaches are focused on cost-side optimizations which cover only half of the cost-revenue relation. RM, defined by Talluri and Van Ryzin (2004) as the demand management decisions aimed at increasing a firm's revenue, can have an equally strong impact on profitability. According to these authors, conditions under which RM is particularly promising include customer heterogeneity, demand variability and limited resources. As parcel logistics usually involves markets characterized by all three of these conditions, the express package delivery industry should be primed to employ RM-techniques.

Interestingly studies that explicitly deal with demand management of consolidated freight services have remained scarce. One notable study is Lin et al. (2009) who developed an algorithmic framework based on implicit enumeration and Langrangian relaxation to derive origin-destination (OD) specific prices for LTL-carriers. The authors also accounted for delivery time restrictions in the selection of time-feasible delivery paths. Later Lin and Lee (2015) also developed an exact algorithm to determine profit maximizing zone prices for LTL-operations. They demonstrate that zone pricing is necessarily less profitable than OD-specific pricing. In both papers demand was aggregated using OD-specific inverse demand functions, yielding concave expressions for revenue.

In contrast, other fields of transportation, like air traffic and container shipping, have traditionally been on the forefront of RM-innovations (see McGill and Van Ryzin (1999) and Zurheide and Fischer (2014) for reviews). Research in these areas has resulted in a wide array of optimization approaches for tariff setting and product line design, including a number of linear programming based formulations. A well known example here is Bartodziej et al. (2007), who study OD-pricing for cargo airlines with a network flow representation. However, most papers dealing with joint design and pricing of transportation networks use bilevel formulations as these allow for a static representation of a non-cooperative Stackelberg game. As such the upper level of the program models the decision of a leader who attempts to maximize revenue while users of the network are treated jointly as cost-minimizing followers on the lower level. Such a setup was first introduced by Labbé et al. (1998) and subsequently extendet by Brotcorne et al. (2000, 2001) who designed a freight-tariff setting problem with a carrier controlling tolls on a set of arcs which are used by flows of goods. Brotcorne et al. (2008) later extend this approach by including investment fixed costs for operating arcs and solving the problem via a Langrangian relaxation based heuristic. Brotcorne et al. (2011) also developed an exact approach for solving these kinds of problems as well as a tabu search heuristic (Brotcorne et al., 2012). Conceptually similar is Crevier et al. (2012) who also use a bilevel model for an integrated pricing and operations planning problem with their use case being rail freight transportation.

What all of these bilevel programming based studies have in common is that they neglect congestion effects and thus assume an uncapacitated network. This also translates into a simplified treatment of demand which is assumed to be fixed and customers decide independently and fully rationally based on shortest paths, irrespective of the decisions of other individuals. The integration of more flexible demand structures into operations planning for intermodal transport was studied by Li and Tayur (2005). Instead of using some form of aggregate demand, they directly utilize customers' individual reservation prices, which can be obtained via standard marketing techniques (i.e. either conjoint analysis or discrete choice experiments). Based on these reservation prices they regress the parameters of a special kind of probability density function which yields a concave expression for revenue. A drawback of the approach, besides the evident limitations in suitable distributions, are the comparably small sets of routes and paths that can be handled. An alternative approach would be to regard the individual reservation prices as a representative sample and use them directly in a non-linear mixed integer problem (see Dobson and Kalish, 1988; Green and Krieger, 1985; McBride and Zufryden, 1988). A fulfillment model using this kind of demand representation appears very enticing as it would allow for a very flexible treatment of demand and underlying decision processes. Additionally, individual customer choices can be elegantly included in the mixed integer models used in network design formulations, enabling a reciprocal influence of consumer behavior and fulfillment costs. However, Dobson and Kalish (1993) have shown that such a deterministic product line model is NP-complete, which makes a combination with an equally NP-complete network design proplem particularly challenging to solve.

Our short literature review should indicate that a major shortcoming of existing approaches is an overly simplified treatment of demand. We believe that in order to enable more sophisticated revenue management in LTL-networks, adequate modelling of customer behavior will be necessary. As such the mathematical model and associated solution procedure presented in the following chapters shall demonstrate the feasibility of such an integrated approach. One could even think about bypassing the need for individual reservation prices entirely and directly express the market shares as a function of price. The issue here is that this typically yields multi-dimensional integrals which are difficult to handle in any optimization approach. Still, for some families of discrete choice models closed form expression for market shares exist. In this case our developed metaheuristic would allow for the direct implementation of a full discrete choice process. However, we decided to save this for a later contribution as it would prohibit comparison with an exact solution approach.

3 Problem statement

The proposed approach links service selection and pricing to the task of optimizing a logistics network. These two problems stem from separate research fields, exhibiting some major differences in mathematical formulations as well as in theoretical concepts employed. In order to streamline our presentation we first introduce the product line selection and pricing problem before turning to the fulfillment part of our model. At the end of this chapter these two parts are combined into an integrated optimization approach.

3.1 Service selection and pricing problem

In order to separate consumers who value quick delivery from those that are less time but more price sensitive, express carriers typically segment their services according to guaranteed delivery time (one-day, two-day, etc.). Contingent on the set of services offered, customers will pick a service that maximizes their personal net benefit (i.e. utility minus price) and the shipper will then be obligated to deliver the parcel within the specified guaranteed delivery time. We take the perspective of the shipper who tries to determine the optimal set of services and associated prices in order to maximize profit.

A basic segmentation that shippers frequently apply is according to service type into express and deferred delivery. This simple differentiation is already quite useful as it allows the carrier to dedicate capacity on fast but more costly transportation services to high revenue orders instead of wasting expensive resources on orders where the benefits of quick delivery are not rewarded. While the field of transport and logistics provides little formal guidance on how to design delivery services, product line optimization has been subject of extensive research in the fields of marketing and economics. A rather straightforward integer programming formulation that deals with price segmentation was introduced by Dobson and Kalish (1988) and heuristic approaches for dealing with this formulation were further discussed in Dobson and Kalish (1993). They assume the firm to be a local monopolist who seeks to determine the optimal combination of products and prices in a product line differentiated by quality.

For an express parcel carrier the product portfolio consists of only one homogeneous service, i.e., delivery of a parcel. However, when differentiating according to guaranteed delivery time, parcel delivery services can be viewed as a product line in which faster service is equivalent to higher quality. The shipper then needs to determine the optimal set of guaranteed delivery times to offer, given that customers will self-select themselves based on their individual sensitivities to delivery time and price.

For this we assume that we have data on preferences of a representative sample of customers (e.g. from conjoint analysis or discrete choice experiments) and that these customers behave rationally and choose the service that maximizes their personal welfare (i.e., utility from delivery minus cost of service). Each customers' utility function is dependent on random sensitivity parameters for delivery time and price and is determined outside of the model[2]. Using the model of Dobson and Kalish (1988) as a basis, the revenue management part of our approach, referred to as SSPP, can be formulated as follows:

$$
\text{(LP)} \quad
\begin{aligned}
\max \quad & \sum_{k \in K} \sum_{s \in S} \sum_{n \in N} \frac{d_k}{|N|} z_n^s p^s - \sum_{s \in S} f^s o^s && (1) \\
\text{s.t.} \quad & \sum_{\sigma \in S} (u_n^\sigma - p^\sigma) z_n^\sigma - (u_n^s o^s - p^s) \geqslant 0 \quad , \forall n \in N, s \in S && (2) \\
& \sum_{s \in S} z_n^s = 1 \quad , \forall n \in N && (3) \\
& \sum_{n \in N} z_n^s + p^s - M o^s \leqslant 0 \quad , \forall s \in S && (4) \\
& o^0 = 1 && (5) \\
& p^0 = 0 && (6)
\end{aligned}
$$

$p^s \in \mathbb{R}^+$	price variable of service s	S	set of services
$z_n^s \in \{0, 1\}$	is 1 if service s is selected by customer segment n	N	set of customer segments
$o^s \in \{0, 1\}$	is 1 if service s is offered	K	set of commodities
d_k	quantity of commodity k	f^s	fixed cost of offering service s

The objective (1) maximizes revenue minus fixed costs of offering the chosen services. These costs may represent overhead or any other costs incurred by offering additional services which are unrelated to the actual fulfillment process (delivery costs are included in the second part of the model). Using binary variables o^s the shipper can select any combination of services $s \in S$ and must determine the associated profit maximizing prices p^s. Note that the so-called no-purchase option, denoted s^0, is always offered, as it allows customers to exit the market in case none of the offerings is attractive to them. Starting with the fastest service s^1, all other services are indexed in order of increasing delivery due time.

Note that we do not model each shipment as a separate customer as this would require an enormous amount of variables. Instead we first aggregate shipments with identical availability time, origin and destination into a set of commodities K. Second, we consider our set of customers to be representative of a larger population. As prices are uniform to all and by assuming that customers form

[2] The appendix includes more detailed information on the type of utilities used. See also Akcay et al. (2010), Berry and Pakes (2007), and Song (2007).

a representative sample with preferences that do not differ among locations, we are able to translate the individual customers' decisions into arbitrarily large demand volumes as follows: Binary decision variables z_n^s are one if customer n decides to use service s. Since each representative customer chooses exactly one service, we can aggregate all decisions and interpret this as a market share. As demand is similarly structured across all locations, this share will apply to all commodities $k \in K$ and respective demands d_k between hubs. This way we model a potentially unlimited number of parcels by a modest sample of customer decisions. In our objective (1) market shares are obtained by summing over all decisions z_n^s of representative customers n on services s and then dividing by the total number of customers $|N|$. Multiplying by demand d_k of commodity k and price p^s we obtain revenue.

Constraint (2) enforces rational behavior of the representative customers. Each individual $n \in N$ decides for the service where the welfare $u_n^s - p^s$ gained is at least equal to what it would achieve with choosing any other service. Customers must choose exactly one service which is assured by constraint (3). If price exceeds utility for all offered services, the customer will choose the no-purchase option s^0, making the revenue contribution disappear. Constraint (4) states that customers can only decide for services that are offered and additionally forces prices of non-offered alternatives to zero[3]. Finally, constraints (5) and (6) specify that the no-purchase option is always offered and that its price is zero.

Owing to the products of binary variables o^s with real-valued price variables p^s, both the objective (1) and the rational behavior constraint (2) include bilinear terms. Dobson and Kalish (1993) developed heuristic procedures to solve this NP-complete problem. We experimented with modifications of their approach but concluded that it was inappropriate once you combine it with computationally intensive routing considerations. We therefore developed our own approach which will be described after we introduced the integrated formulation of the problem.

3.2 The express service fulfillment model

When determining the optimal set of guaranteed delivery times the basic intuition is that shorter guaranteed delivery times increase delivery costs as it reduces opportunities to realize economies of scale. As such the degree to which one can consolidate shipments at intermediate locations is vital for the efficiency of the fulfillment operations. One class of models primed for this purpose are SNDPs. These models have been successfully employed to solve a wide array of transportation problems.

SNDPs are typically dynamic, in the sense that they span multiple time-periods via an expanded space-time network. This is necessary to adequately model the routing of shipments and vehicles as they move through the system. The inherent size of such networks, however, limits the time scale that those formulations can cover. Since product selection and pricing usually involve longer planning periods than routing decisions, we should align the differing planning horizons by generating

[3]If non-offered services would have positive price we would have to multiply p^s in constraint (2) by o^s yielding another bilinear term. This is avoided by forcing those prices to zero.

distribution plans that can be executed repeatedly. To assure this we decided to use a cyclical SNDP formulation as the basis for our model (see Andersen et al., 2011; Andersen et al., 2009).

As last mile delivery is often operated on the same set of resources, regardless of differences in guaranteed delivery times, we refrained from a dedicated modeling of last-mile operations and use a uniform cost factor c_{LM} per unit of demand instead. As such we assume shipments to enter and leave our network at the hubs. The carrier's transport operations are modeled as a space-time network $G = (H, A)$, with the set of arcs A representing possible connections between the hubs H. Unless noted otherwise, arcs and hubs always refer to the time expanded version of the network, i.e. they denote a physical location at a specific point in time. The schedule length is divided into a set of periods $T = \{1, \ldots, T_{\text{MAX}}\}$. Note that our cyclical formulation implies that the first period is the successor of the last. An illustration of the type of network considered is depicted in Fig. 1.

We consider a multi-modal logistics network with two types of vehicles, cargo aircraft and trucks. Both operate between the same set of hubs, where packages can be consolidated and transferred from one vehicle to another. As aircraft can travel much larger distances during each time period, aircraft arcs form a very dense graph compared to trucks, whose connections are limited by the hubs they can reach within a reasonable amount of time. A set of holding arcs A_h allows shipments and trucks to be stored at the nodes while aircraft, due to their high fixed costs, need to be operated throughout the planning horizon. An illustration of such a network is depicted in Fig. 1.

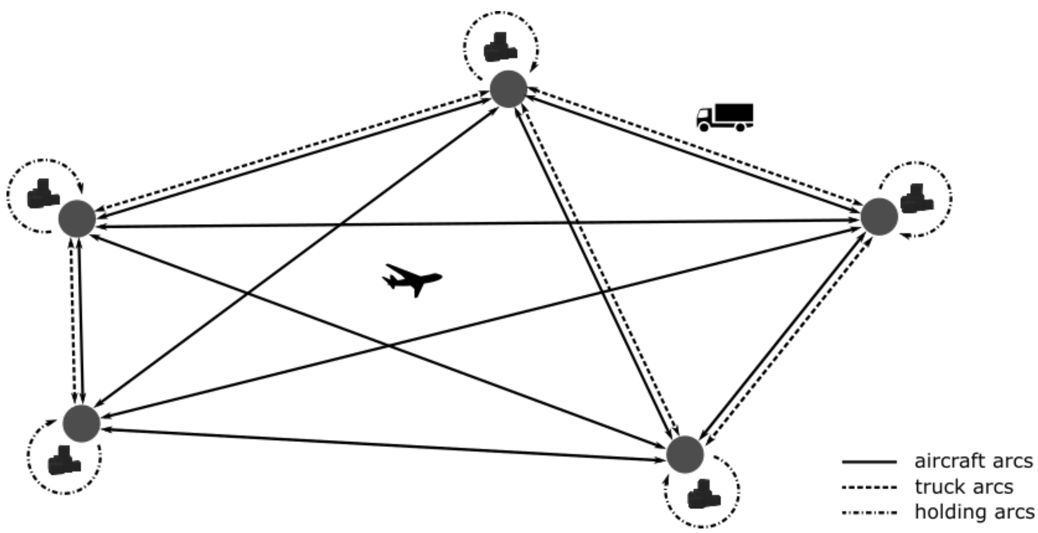


Figure 1: The physical structure of the express air hub network

Note that the shortest guaranteed delivery time the carrier considers, strongly influences the required density of the graph and thus the size of the overall model. For example, if the shortest service guarantees delivery within two periods, then it needs to be possible to reach every physical destination from each time-expanded node within these two periods. Otherwise customers might demand a delivery time which is not feasible in the underlying graph. Consequently, short delivery times require dense graphs. In combination with the fact that express carriers typically employ large fleets of vehicles this would become a major issue for any arc-based formulation as the number of constraints would grow too quickly. In line with results from Andersen et al. (2009) we thus chose to use a SNDP formulation based on routes and paths, as this has proven to computationally outperform arc-based formulations, especially on larger instances. Moreover, the fleet size does not directly influence the difficulty of our problem anymore, as the usage of a vehicle simply corresponds to a chosen route in the objective, while the number of constraints and variables remain unaffected.

The downside of this formulation is that one needs to consider an enormous set of possible routes and paths. Unless these are generated dynamically via column generation, the number of routes and paths that need to be generated a priori quickly becomes prohibitively large. However, a large portion of all possible routes and paths are typically not in line with the way express carriers operate (e.g. shipments making numerous changes between aircrafts and trucks or vehicles that do not regularly return to a given home base). Therefore, we generate only those routes and paths meeting a number of operational restrictions that typically arise in the context of express parcel delivery (see the appendix for a complete list of the restrictions).

We again aggregate shipments with identical availability time, origin and destination into a set of commodities K and respective delivery quantities q_k. Given that we know the market shares ϕ^s of each offered delivery service [4], we can specify our fulfillment model FP as follows:

$$\text{(FP)}\quad \begin{aligned} \min \quad & \sum_{v\in V}\sum_{r\in R_v} f_v^r y_v^r + \sum_{k\in K}\sum_{p\in P_k} c_k^p x_k^p + \sum_{k\in K} c_{LM}(1-\phi^0)d_k && && (7)\\ \text{s.t.} \quad & \sum_{k\in K}\sum_{p\in P_k} x_k^p \alpha_{ij}^p - \sum_{v\in V}\sum_{r\in R_v} \kappa_v \beta_{ij}^r y_v^r \leqslant 0 && ,\forall (i,j)\in A\backslash A_h && (8)\\ & \sum_{r\in R_v} y_v^r \leqslant e_v && ,\forall v\in V && (9)\\ & (1-\phi^0)d_k = \sum_{p\in P_k} x_k^p && ,\forall k\in K && (10)\\ & \sum_{\sigma\in S:\sigma\leqslant s} \phi^\sigma d_k - \sum_{p\in P_k: t^p\leqslant\tau^s} x_k^p \leqslant 0 && ,\forall k\in K, s\in S\backslash\{0\} && (11) \end{aligned}$$

$x_k^p \in \mathbb{R}^+$	flow of commodity k on path p	P_k	set of paths p_k for commodity k
$y_v^r \in \{0,1\}$	is 1 if vehicle v is used on route r	R_v	set of routes r_v for vehicles of type v
c_k^p	variable costs of commodity k on path p	A	arc set of the space-time network
c_k^p	variable costs of commodity k on path p	V	set of different vehicle types
α_{ij}^p	is 1 if arc (i,j) is part of path p	f_v^r	fixed costs of vehicle v on route r
β_{ij}^r	is 1 if arc (i,j) is part of route r	κ_v	capacity of vehicle of type v
d_k	total amount of commodity k	e_v	max. number of vehicles of type v
ϕ^s	market share of service s		

In the objective (7) we sum fixed costs f_v^r for using a vehicle of type v on route r and variable

[4] Market shares are denoted by ϕ^s for the moment but will be replaced in the integrated model by an expression representing aggregated customer decisions.

costs c_k^p, incurred by sending an amount of shipments x_k^p of commodity k on path p. Additionally, we may add the costs c_{LM} for last-mile delivery, which are constant in this case as market shares are fixed in this subproblem. Constraint (8) assures that enough vehicle capacity κ_v is provided to serve all selected paths, with α_{ij}^p and β_{ij}^r indicating whether arc (i, j) is part of path p and route r, respectively. Constraint (9) limits our fleet size to a type-specific maximum of e_v. The actual quantity to be transported is obtained by subtracting the market share of the no-purchase option ϕ^0 from demand d_k (10). This quantity needs to be equal to the sum of path flows. The delivery time requirements are enforced by (11), stating that a sufficient fraction of each commodity must be delivered on paths with a duration t^p less than or equal to the guaranteed delivery time τ^s of service s.

3.3 An integrated approach to service selection, pricing and fulfillment

Now that we have introduced the two basic components of our model we can combine them in order to derive an integrated approach of service selection, pricing and order fulfillment. The two models are merged in a joint profit maximizing objective function and are linked in such a way that the individual choices of the customers translate into the routing constraints and vice versa. More specifically we need to replace the constant market shares ϕ^s of constraints (10) and (11) by the ones aggregated from customer decisions, as already seen in the objective (1). We call the resulting integrated formulation service selection, pricing and fulfillment problem (SSPFP). It extends service network design by endogenous demand and endogenous delivery due times.

$$\max \sum_{k\in K}\sum_{s\in S}\sum_{n\in N}\frac{d_k}{|N|}z_n^s p^s - \sum_{v\in V}\sum_{r\in R_v} f_v^r y_v^r - \sum_{k\in K}\sum_{p\in P_k} c_k^p x_k^p - \sum_{s\in S} f^s o^s - \sum_{k\in K} c_{LM}(1-\phi^0)d_k \tag{12}$$

(SSPFP)

$$\text{s.t.} \quad \sum_{k\in K}\sum_{p\in P_k} x_k^p \alpha_{ij}^p - \sum_{v\in V}\sum_{r\in R_v} \kappa_v \beta_{ij}^r y_v^r \leqslant 0 \qquad ,\forall (i,j)\in A\backslash A_h \tag{13}$$

$$\sum_{r\in R_v} y_v^r \leqslant e_v \qquad ,\forall v\in V \tag{14}$$

$$\sum_{\sigma\in S}(u_n^\sigma z_n^\sigma - p^\sigma z_n^\sigma) - (u_n^s o^s - p^s) \geqslant 0 \qquad ,\forall n\in N, s\in S \tag{15}$$

$$\sum_{s\in S} z_n^s = 1 \qquad ,\forall n\in N \tag{16}$$

$$\sum_{n\in N} z_n^s + p^s - Mo^s \leqslant 0 \qquad ,\forall s\in S \tag{17}$$

$$\sum_{n\in N}\left(1 - z_n^0\right)\frac{d_k}{|N|} = \sum_{p\in P_k} x_k^p \qquad ,\forall k\in K \tag{18}$$

$$\sum_{\sigma\in S:\sigma\leqslant s}\sum_{n\in N}\frac{d_k}{|N|}z_n^\sigma - \sum_{p\in P_k:t^p\leqslant\tau^s} x_k^p \leqslant 0 \qquad ,\forall k\in K, s\in S\backslash\{0\} \tag{19}$$

$$o^0 = 1 \tag{20}$$

$$p^0 = 0 \tag{21}$$

$$z_n^s, o^s, y_v^r \in \{0,1\} \qquad x_k^p, p^s \in \mathbb{R}^+ \tag{22}$$

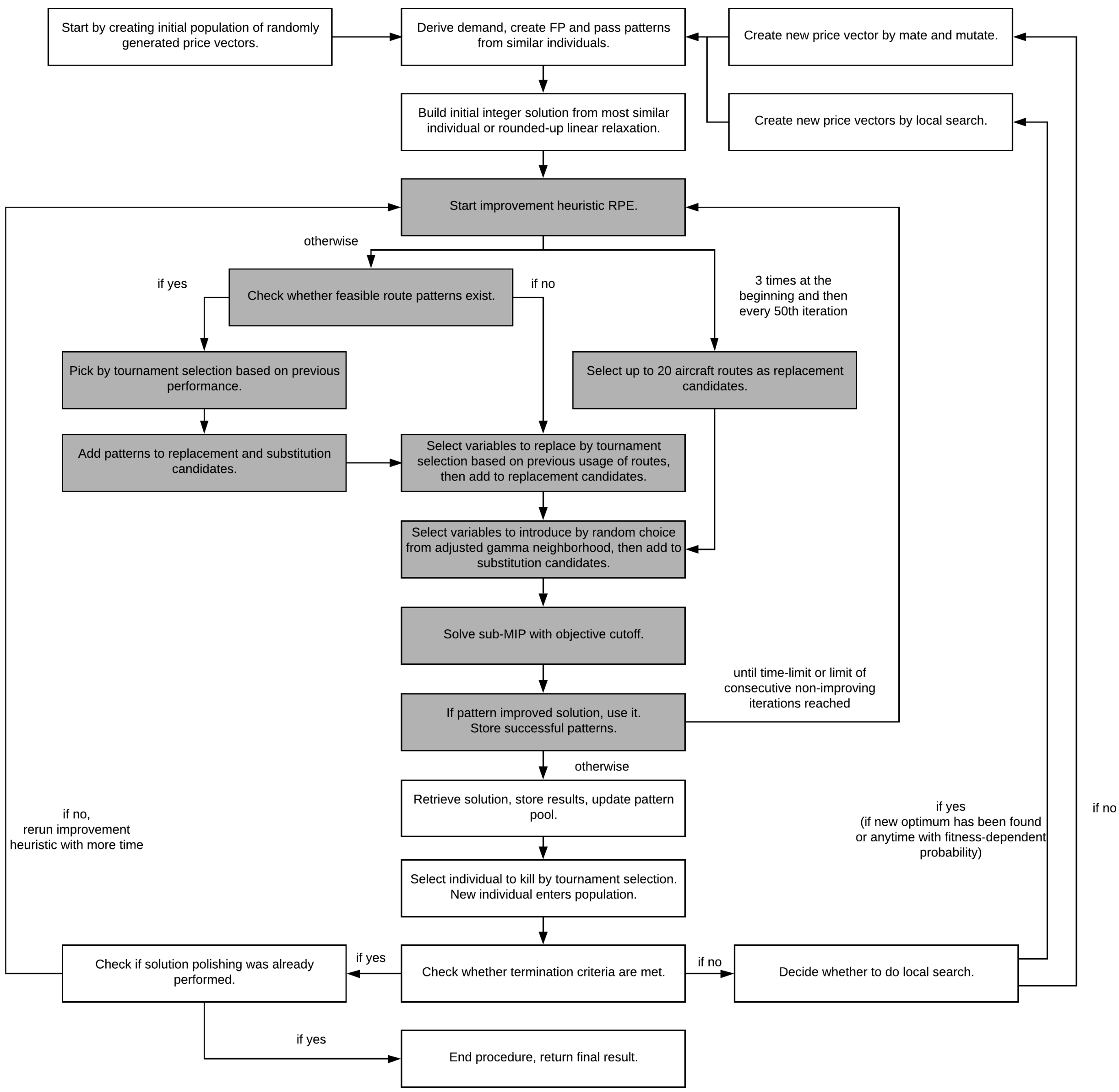
Note that due to the product of binary decision variables z_n^s and continuous price variables p^s, the integrated model features bilinear terms. It can be linearized by replacing this product with a new variable, whose behavior mimics the bilinear expression through a set of big-M constraints (see e.g., Wu, 1997). The resulting model has a very weak bound and any commercial MIP-solver will have issues solving instances of only moderate size. However, for small instances it can provide optimal values and thus serve as a benchmark for our heuristic solution approach which will be described subsequently.

4 Solution approach

Solving our integrated model is challenging as both components, the service selection problem (SSPP) and the fulfillment problem (FP), are known to be NP-complete problems with weak linear relaxations. However, for a given solution to SSPP, we would know the market shares of all services and thus the quantities that need to transported in the specified due times. We then would have everything needed to solve FP, which is a potentially large but linear service network design problem.

Following this logic we decompose the integrated model SSPFP into its two already known components, SSPP and FP, by using a asynchronous parallel genetic algorithm (GA) that operates on the price vectors and generates a new solution to SSPP in each iteration. This in turn means that we need to solve one instance of the remaining FP with each new individual created. As multi-commodity fixed-charge network design problems, like FP, are notoriously hard optimization problems, it appears fairly optimistic to iteratively solve a large number of them. Consequently our GA is aimed at keeping the number of required iterations as low as possible. Additionally we attempt to exploit historic knowledge in order to speed up the search for good quality solutions of the fulfillment problems. The different parts of the algorithm are explained in detail in the following sub-chapters. A schematic illustration of the complete solution procedure is depicted in figure 2. Associated pseudo code together with required parameter values can be found in the appendix.

Figure 2: Illustration of metaheuristic solution procedure AE-RPE. Note: Tasks in white boxes are performed by master process, tasks in grey boxes are being run in parallel on slaves.



4.1 Asynchronous evolution - AE

In order to perform a sufficient number of iterations within a reasonable amount of time we implemented a parallel GA, operating asynchronously on a shared population (commonly referred to as asynchronous evolution or AE). The master process manages the population and the pattern pool, generating new individuals and fulfillment problems whenever a worker is available. Observe that as the fulfillment problems exhibit very heterogeneous solution times asynchronous operation of the GA is a key feature of our algorithm as any generational model would have to wait for all workers to finish before proceeding. Particularly long evaluation times of high quality individuals, however, can still pose a problem as they cannot spread their properties in the population until their evaluation is finished, thus putting these individuals at a disadvantage. To circumvent this issue we exploit the fact that the differences in solution quality are quite high at early stages of the GA and become less pronounced later on. We thus stop the heuristic procedure after a given maximum amount of time even if it continues to improve the fulfillment problem. The high disparities among individuals partly offset the sub-optimal solution values at early stages of the population. Since the GA allows the same fulfillment problem to be evaluated multiple times the evaluation can be continued later on when small improvements are of higher importance.

4.2 Recombination and mutation

Starting from a random initial population the GA iteratively creates new price vectors. Parent individuals for mating are selected via tournament selection, with one parent being selected based on solution quality and the other one based on the distance with respect to the first parent. This way we strike a balance between intensification and diversification. Note that with an ordered set of delivery services it would be counterintuitive if a faster service has a lower price. As such, whenever we are mating or mutating we should enforce that prices must form a decreasing sequence. This greatly reduces the range of values each price can take and thus also the number of needed iterations. For the crossover operation we chose Extended Intermediate Recombination, a variant designed by Muhlenbein and Schlierkamp-Voosen (1993) which is particularly suitable for real-valued vectors. Their choice of a crossover parameter in the interval $[-0.25, 1.25]$ also performed best in our experiments. We randomly determine the elements that are crossed over and all remaining elements are directly inherited from one of the two parents. The frequency of mutations is determined by a fixed mutation rate, however, the variance of mutations is adaptive. As each price is an element of a decreasing price sequence it is bounded from above and below by its neighboring prices and the minimum and maximum prices possible, i.e. zero and maximum utility value. In order to make sensible mutations the standard deviation used should account for the size of this interval.

4.3 Solution of SSPP and derivation of FP

Observe that for a given price sequence $p^0, p^1, \ldots, p^s$ for $s = |S| - 1$ it is trivial to determine which alternative will maximize each customer's welfare in constraint (2) of SSPP. In this case we can easily obtain the optimal values of decision variables z_n^s and solve SSPP. Note that decision variables o^s can be neglected in this case as a service can always be forced to have zero market share by charging a price higher than the maximum utility obtained by any individual in the sample. Each new price vector can thus be mapped to a corresponding market share vector that uniquely defines the quantities and time restrictions of FP. What is more, Dobson and Kalish (1988) have shown that once an assignment of customers to products is given we can easily determine the revenue maximizing prices that would generate such market shares by solving a set of shortest path problems (as arcs can have negative values in these problems we use Bellman-Fords's algorithm for obtaining the revenue maximizing prices). This greatly reduces the number of iterations needed as we only have to generate each market share vector once in order to immediately know what the associated maximum revenue is. After logging this information we pass on the respective FP to one of the available workers.

4.4 Deriving good initial solutions from similar individuals

Genetic algorithms typically make a lot of minor tweaks to a real-valued solution vector, i.e., the price vector and thus also the customer decisions often remain largely the same. This in turn causes only minor changes in the quantities to be transported within the various delivery due times. As the quantities of the offered delivery services are the only aspect of our delivery problem that changes between iterations, we can expect that many instances of FP will resemble one another. Consequently we try to speed up solution times of FP by trying to learn as much as possible from similar, already evaluated instances. This is achieved by the following process.

Whenever we need to solve an instance of FP, we determine which already evaluated individual was most similar to the current one. We then fix all routes used in the similar individual's solution to one and solve the linear relaxation of the remaining sub-MIP. All routes having fractional values in the optimal LP solution are then rounded up, yielding a feasible initial solution which is easy to obtain and comparably tight (In our experiments this initial solution was typically within ten percent of optimality CHECK!!!).

One challenge that remains is how to determine similarity of individuals. The price vectors themselves would result in an inappropriate measure of similarity as different price sequences can be mapped to the same market shares. As such it is market shares, rather than prices that uniquely define a fulfillment problem. In the simple case where two individuals have equal market share vectors, it is trivial to observe that associated FPs are identical, as all quantities and time restrictions are then the same. In this case we can simply adopt the best solution of the identical individual and try to further improve it. In all other cases we need to find a measure of similarity that adequately reflects the implications that different quantities and time restrictions have on the selection of routes and paths. For this we tested three different measures of similarity based on the market share vector of individuals. We compared a Euclidean distance measure with one that we based on a version of Spearman's footrule which accounts for position weights and similarities among rank elements (Kumar and Vassilvitskii, 2010). As a benchmark we added a random similarity measure consisting of a number drawn from the uniform distribution between zero and one.

Table 1: Comparison of distance metrics: For a description of test instances see section 5. For each instance dataset A was used.

instance	average absolute value of initial solution			Spearman's footrule relative to	
	Spearman's footrule	Euclidian	random	Euclidian	random
i1	2,440,389.7	2,464,077.7	3,191,245.7	−1.0 %	−23.5 %
i2	3,353,373.6	3,380,773.7	4,493,127.5	−0.8 %	−25.4 %
i3	3,213,451.1	3,235,675.8	4,306,681.0	−0.7 %	−25.4 %
i4	2,946,210.9	2,993,484.6	4,342,959.3	−1.6 %	−32.2 %
i5	5,419,672.4	5,662,639.2	8,967,849.1	−4.3 %	−39.6 %
i6	6,895,625.5	7,492,870.5	8,906,659.1	−8.0 %	−22.6 %
i7	11,556,733.5	11,620,284.6	15,549,185.7	−0.5 %	−25.7 %
i8	10,953,117.8	12,328,449.6	17,709,100.6	−11.2 %	−38.1 %
i9	7,778,994.0	8,902,401.4	12,150,223.0	−12.6 %	−36.0 %
i10	15,302,044.0	16,951,162.5	23,828,417.7	−9.7 %	−35.8 %
mean				−5.0 %	−30.4 %

As can be seen in table 1 our similarity measure based on Spearman's footrule clearly performed best, with average improvements in initial solutions of 5% and 30% relative to a Euclidian and random distance measure. As our similarity-enabled version of Spearman's footrule is able to account for correlation among alternatives, it adequately mirrors that one- and two-day services are closer substitutes than one- and five-day services. As an example, an individual who has ten percent market share of service one shifted to service two is regarded as being more similar to the original than one with equal amount shifted from service one to service five. Euclidean distance metrics on the other hand would regard both cases as having identical similarity. Therefore, whenever we refer to a distance measure in the following chapters, we applied the one based on Spearman's footrule (also see algorithm 3 in chapter B of the appendix).

4.5 Route-pattern exchange heuristic - RPE

Each worker subsequently runs a newly developed improvement procedure, called route-pattern exchange (RPE) heuristic, which is also designed to exploit historic knowledge from previous evaluations. It involves a learning component that has been used in various forms by Rochat and Taillard (1995) and others. It stems from the idea that one can intensify and guide the search to promising regions of the solution space by identifying common variables in high-quality solutions and by exchanging components of those solutions among each other. This usage of so-called consistent variables, however, is usually employed in the context of local search heuristics and involves different solutions of one and the same routing problem. We, on the other hand, start from an initial solution and try to improve upon it by learning from other, similar problems. Therefore we adjusted this well-known strategy such that we identify small sets of currently chosen routes, referred to as replacement candidates, that are particularly inconsistent. For each of them we then search for promising substitution candidates and evaluate them together in a sub-MIP to see whether we can find a lower cost combination of routes and paths. All objective-improving sets of replacement and substitution candidates, called route-patterns, are shared via a pattern pool between workers.

Selection of replacement candidates is based on a distance-weighted usage statistic ω_v^r, providing information on how popular this route is in the best solutions of all other previously evaluated individuals. It is calculated according to

$$\omega_v^r = \sum_{i \in I} (y_v^r)_i \log \frac{1}{d_i} \qquad \forall v \in V, r \in R_v. \tag{23}$$

Each solution value of variable y_v^r across all individuals i is weighted by the log of the inverse distance measure d_i. As such, a low ω-score indicates that this route can be considered as weakly determined for the given individual. On the basis of this metric we use tournament selection to determine a small number of routes in the current solution which seem promising to reevaluate.

In order to select good substitution candidates for these variables we utilize the concept of a γ-neighborhood, found in the context of *k-opt* heuristics for MIP-solvers (see Achterberg, 2009; Hendel, 2011). This neighborhood consists of all variables sharing a minimum ratio γ of rows with the selected variable. As such, potential substitution candidates are those routes that use a certain fraction of the same arcs as the replacement candidate. Note, however, that a strict application of this concept is problematic, as different classes of vehicles do not share any arcs, thus making it impossible for a set of trucks to replace an aircraft. Likewise, routes connecting a similar set of physical nodes but at slightly different times should also be considered as substitutes. Before we can define a proper neighborhood, we thus need to determine which arcs have comparable properties. For this we introduce the concept of an adjusted coefficient matrix $\bar{A}$.

Definition 1. Let P be an instance of our fulfillment problem and let $a_{ij} \in A$ be the coefficients of route variables in constraints of type (8). Then each constraint $i \in I$ corresponds to an arc and each variable $j \in J$ represents a route. Associated with each constraint i are departure time t_i, physical origin o_i and physical destination d_i. If ϵ denotes the departure tolerance in periods, then for all $i \in I, j \in J$ we can define the coefficients $\bar{a}_{ij}$ of the adjusted matrix $\bar{A}$ as follows:

$$\bar{a}_{ij} = \begin{cases} 1 & \text{if } \exists\, k \in I \;:\; a_{kj} = 1 \;\wedge\; o_k = o_i \;\wedge\; d_k = d_i \;\wedge\; |t_i - t_k| \leqslant \epsilon \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Whenever coefficient matrix A has value 1, the adjusted coefficient matrix $\bar{A}$ receives value 1 in all rows that correspond to arcs with same origin and destination and within the departure tolerance ϵ, irrespective of the arc's vehicle type. Using this adjusted coefficient matrix we can now define the γ-neighborhood.

Definition 2. Let P be a mixed-integer program, $j \in J$ a given variable and $\bar{A}_j$ a column of its adjusted coefficient matrix. Then for a fixed matching rate $\gamma \in [0, 1]$ we can define the γ-neighborhood as

$$\Gamma_j^\gamma = \left\{ i \in J \setminus \{j\} \;\middle|\; \frac{|\bar{A}_j \cap \bar{A}_i|}{|\bar{A}_j|} \geqslant \gamma \right\} \tag{25}$$

to be all integer variables of P which share a certain ratio γ of rows in $\bar{A}$ with j.

For each replacement variable we then take a random sample of variables from its respective γ-neighborhood, together forming our set of promising substitution candidates. After fixing all route variables, except replacement and substitution candidates, we solve a sub-MIP with an objective cutoff. Any feasible solution to this sub-MIP improves the original FP. We adopt the best improvement made and log the differences with respect to the previous solution. The resulting set of m replacement candidates that left and n substitution candidates that entered the solution represents an *m:n*-improvement, referred to as a route-pattern exchange. Following the intuition that successful improvements provide valuable information for other FPs we add these patterns to a pattern pool that is managed by the master process. Whenever a new individual's FP is created we rank all patterns in increasing order according to the distance of the creating instance to the current one. The ten thousand least distant patterns are attached to the problem data and passed on to the worker.

After retrieving an initial solution the RPE heuristic checks whether any of the patterns match the current solution. If not we simply create new patterns as described above. In case there are matching patterns, a maximum number of five of them is selected via tournament selection, based on the absolute objective improvement these patterns generated. The winning patterns are appended to the set of previously determined replacement and substitution candidates and together with all real-valued path variables a sub-MIP is evaluated. Our experiments showed that it is beneficial to first concentrate on replacing variables representing aircraft routes before allowing all routes to be evaluated. This is intuitive as aircraft routes have much higher cost coefficients in the objective than routes representing trucks.

4.6 Local search

Besides random selection, new individuals can also be obtained through training of previous solutions via a local search procedure. For each service we identify the customer decision which prevents us from increasing or decreasing a given services' price and reassign this customer to the next best alternative. We then compute the profit maximizing prices for the new assignments and use these price sequences as our new individuals (see Dobson and Kalish (1988) for a more detailed description of this procedure). The best solution found in this local search is injected into the population. As this process is computationally expensive we employ local search only with a certain quality-scaled probability or every time a new best solution is found.

4.7 Solution polishing

As our heuristic is primarily designed to yield good solutions quickly it might happen that the best solution retained by the GA could still be somewhat improved if we would reoptimize its FP. Therefore the algorithm concludes with another ten minutes of RPE heuristic on the best individual's FP.

5 Data

As noted in chapter 3.2, the time-restrictions imposed by the delivery services imply that it needs to be possible to reach every physical node from each time-expanded node within the shortest offered delivery time. In case of dedicated networks (e.g. up to 3-day exclusively served by aircraft, beyond by trucks) a feasible model even requires that this condition is fulfilled for each mode separately. This imposes some unique requirements on the structure of networks which are typically not met by commonly used test instances. As such we decided to build our own networks by randomly allocating hubs on a plane. We then added all arcs that were possible given the vehicles' speed limitations. For aircraft this implies a complete graph in the physical network, repeated in each period. In order to reflect real-world limitations, as for example availability of airport slots, we randomly removed a few aircraft arcs. We subsequently checked whether the given network would be feasible with respect to the shortest delivery times. Speed and capacity of aircraft are modelled after a Boeing 757-200F, with operating costs of this model taken from section 4 of FAA (2016). Truck data is from table 11 of ATRI (2016). Customer utilities are assumed to be dependent on sensitivity parameters for price and delivery time and are randomly generated by a linear-in-parameters utility function consistent with random coefficients discrete choice models as in Akcay et al. (2010), Berry and Pakes (2007), and Song (2007). For the creation of routes and paths we used the algorithms of Andersen et al. (2009, p. 203) and adapted them in order to account for operational restrictions. For further information on customer utilities and restrictions on routes and paths we refer to section A in the appendix. With ten specifications, each with three datasets, we created thirty different test instances, twelve smaller ones and 18 with larger size. They are described in table 2 and are available upon request. The largest instances resemble the size of UPS' air hub network which features seven major air hubs in Nothern America.

Table 2: Description of test instances

ID	Hubs	Periods	Nodes	Services	Commo-dities	Cust-omers	Arcs min	Arcs max	Paths min	Paths max	Routes min	Routes max
1	4	6	24	3	72	200	125	140	2,084	2,772	129	168
2	4	8	32	4	96	200	178	197	2,471	4,012	211	580
3	5	6	30	3	120	200	178	203	3,493	5,942	184	262
4	5	8	40	4	160	200	249	271	4,766	7,925	297	827
5	5	10	50	5	200	2,000	277	358	6,870	15,909	2,082	2,869
6	5	14	70	5	280	2,000	412	439	8,577	12,188	19,563	115,577
7	6	10	60	5	300	2,000	420	460	21,463	29,737	4,822	22,338
8	6	14	84	5	420	2,000	588	644	22,309	33,342	96,937	172,361
9	7	10	70	5	420	2,000	682	682	56,896	77,562	18,222	34,534
10	7	14	98	5	588	2,000	787	899	32,288	78,319	421,209	464,690

6 Results

We decided to implement our metaheuristic AE-RPE using MATLAB R2017a. The associated Parallel Computing Toolbox lends itself well to the implementation of an asynchronous parallel algorithm and as our metaheuristic spends over 95 percent (CHECK!!!) of the overall computational time in solving sub-problems within CPLEX, a lower-level language would not lead to substantial speed-up. All calculations were perfomed on an IBM x3850 X5 Linux server with four Intel Xeon X7560 processors (32 cores in total) with 330GB of RAM. Table 3 exhibites a comparison on all thirty test datasets between CPLEX 12.8 and our metaheuristic AE-RPE. CPLEX was run for ten hours on the linearized version of model (SSPFP) using the fastest setting of 16 threads. The metaheuristic was able to use all 32 cores and was also stopped after a maximum time of ten hours. The upper part of the table contains results for the smaller test instances where a commercial MIP-solver can provide a meaningful benchmark. Despite the small size of these instances, CPLEX was only able to solve three of them to optimality. On most other small instances CPLEX came reasonable close to the LP bound, while it struggled on instances i4-A and i4-C. The issues CPLEX has in closing the MIP-gap on such small instances already indicate the overall difficulty of the integrated problem and also point to a very weak linear relaxation. As the bottom half of the table illustrates, any further increase in the size of the test instances makes the problem unsolvable for current versions of commerical MIP-solvers. In half of these cases CPLEX cannot even find a profitable integer solution, while the MIP-gaps are still enormous whenever it does. In comparison our metaheuristic AE-RPE proves competitive on the smaller test instances, coming reasonably close to proven high-quality solutions. On the larger test instances the metaheuristic clearly outperforms the MIP-solver, yielding integer solutions that are frequently several times better than the ones reported by CPLEX. The right columns of table 3 exhibit a performance comparison of solution values from AE-RPE after one, two and ten hours relative to what CPLEX reported after ten hours. We consider these results encouraging as they demonstrate that our metaheuristic is able to find high quality solutions in moderate time frames, despite the fact that it is generally designed to be fairly explorative.

Table 3: Performance of metaheuristic AE-RPE compared to CPLEX 12.8 (max. 10h)

instance	CPLEX			AE-RPE			
	objective (10h)	bound	time/MIP-gap	objective (10h)	relative to CPLEX (10h)		
					after 1h	after 2h	after 10h
i1-A	1,459,066.5	1,477,187.0	1.24 %	1,459,066.5	1.00	1.00	1.00
i1-B	1,191,321.7	1,209,595.9	1.53 %	1,189,506.9	1.00	1.00	1.00
i1-C	2,827,032.6	2,827,032.6	10412.5 sec.	2,820,357.4	0.99	0.99	1.00
i2-A	3,773,500.2	3,774,069.8	0.02 %	3,764,256.2	0.98	0.99	1.00
i2-B	4,347,932.7	4,349,655.1	0.04 %	4,335,405.4	0.98	0.99	1.00
i2-C	2,070,986.5	2,107,065.4	1.74 %	2,066,554.5	0.97	0.98	1.00
i3-A	2,208,034.2	2,208,034.2	3880.4 sec.	2,208,034.2	1.00	1.00	1.00
i3-B	1,872,402.3	1,894,799.9	1.20 %	1,870,115.2	0.98	1.00	1.00
i3-C	1,061,931.1	1,061,931.1	5303.8 sec.	1,058,737.4	0.98	0.99	1.00
i4-A	4,913,402.6	5,368,362.4	9.26 %	4,850,680.8	0.99	0.99	0.99
i4-B	2,541,269.9	2,560,361.6	0.75 %	2,513,229.6	0.93	0.95	0.99
i4-C	3,561,379.3	3,790,231.2	6.43 %	3,569,720.0	0.94	0.99	1.00
i5-A	4,666,587.4	9,289,007.5	99.05 %	5,015,584.1	0.97	1.01	1.07
i5-B	847,576.7	3,482,749.1	310.91 %	1,292,022.4	1.22	1.40	1.52
i5-C	2,577,689.7	5,368,963.4	108.29 %	2,885,247.7	1.00	1.07	1.12
i6-A	913,974.9	11,301,719.2	1136.55 %	5,668,927.5	6.20	6.20	6.20
i6-B	317,529.1	11,735,254.0	3595.80 %	5,635,295.2	16.57	17.00	17.74
i6-C	0.0	7,037,625.2	∞ %	2,864,373.6	∞	∞	∞
i7-A	0.0	12,346,340.4	∞ %	6,623,976.6	∞	∞	∞
i7-B	1,288,896.4	11,178,899.6	767.32 %	5,875,495.8	4.18	4.40	4.56
i7-C	4,849,986.4	11,114,527.8	129.17 %	5,454,812.5	0.88	1.01	1.12
i8-A	0.0	13,694,320.1	∞ %	6,517,507.2	∞	∞	∞
i8-B	1,273,962.9	18,168,402.9	1326.13 %	9,214,783.1	6.78	7.15	7.23
i8-C	0.0	17,545,428.6	∞ %	8,575,831.8	∞	∞	∞
i9-A	3,479,052.8	17,113,680.6	391.91 %	9,474,564.5	2.59	2.68	2.72
i9-B	0.0	16,454,841.2	∞ %	9,404,365.2	∞	∞	∞
i9-C	0.0	16,784,723.5	∞ %	8,477,000.3	∞	∞	∞
i10-A	0.0	26,183,204.9	∞ %	13,804,081.4	∞	∞	∞
i10-B	0.0	18,437,177.0	∞ %	9,902,871.0	∞	∞	∞
i10-C	0.0	28,425,385.8	∞ %	14,769,796.5	∞	∞	∞

As the poor performance of CPLEX limits the validity of these results, we also tested how well our route-pattern-based improvement heuristic is able to solve the fulfillment part of the problem. For this we selected the first instance from each of the ten different specifications and resolved it with the metaheuristic. However, this time each fulfillment problem was solved twice, once with CPLEX and once with the improvement heuristic. As is evident from table 4 the improvement heuristic outperformed CPLEX on the linear fulfillment problem in nine out of ten cases.

Table 4: Performance of improvement heuristic

	CPLEX			heuristic	
	best integer	bound	mipgap	best integer	relative to CPLEX
i1	2,258,131.45	2,215,508.02	1.89%	2,260,924.32	1.001
i4	1,520,159.36	1,492,654.27	1.81%	1,515,084.89	0.997
i7	2,438,845.01	2,399,091.83	1.63%	2,673,375.40	1.096
i10	2,363,550.10	2,285,194.16	3.32%	2,355,619.90	0.997
i13	5,219,954.63	4,862,827.04	6.84%	5,146,176.63	0.986
i16	4,684,262.43	3,127,281.99	33.24%	3,402,721.27	0.726
i19	10,913,680.04	10,287,134.07	5.74%	10,633,660.74	0.974
i22	89,345,612.68	3,102,741.30	96.53%	7,663,264.15	0.086
i25	8,708,560.15	4,579,062.33	47.42%	6,369,273.69	0.731
i28	96,365,088.68	431,538.52	99.55%	13,042,797.42	0.135

Given the consistently good performance and the lack of alternative approaches we think that our metaheuristic constitutes a proper solution approach to this challenging problem.

7 Conclusions

References

Achterberg, T. (2009). “SCIP : solving constraint integer programs”. In: *Mathematical Programming Computation* 1.1, pp. 1–41.

Akcay, Y. et al. (2010). “Joint Dynamic Pricing of Multiple Perishable Products Under Consumer Choice”. In: *Management Science* 56.8, pp. 1345–1361.

Andersen, J. et al. (2011). “Branch and Price for Service Network Design with Asset Management Constraints”. In: *Transportation Science* 45.1, pp. 33–49.

Andersen, J. et al. (2009). “Service network design with asset management: Formulations and comparative analyses”. In: *Transportation Research Part C: Emerging Technologies* 17.2, pp. 197–207.

Armacost, A. P. et al. (2002). “Composite Variable Formulations for Express Shipment Service Network Design”. In: *Transportation Science* 36.1, pp. 1–20.

Armacost, A. P. et al. (2004). “UPS optimizes its air network”. In: *Interfaces* 34.1 SPEC. ISS. Pp. 15–25.

ATRI (2016). *An Analysis of the Operational Costs of Trucking: 2016 Update.* Tech. rep. American Transportation Research Institute.

Barnhart, C. and r. r. Schneur (1996). “Air network design for express shipment service”. In: *Operations Research* 44.6, pp. 852–863.

Barnhart, C. et al. (2002). “Network design for express shipment delivery”. In: *Computational Optimization and Applications* 21.3, pp. 239–262.

Bartodziej, P. et al. (2007). "O&D revenue management in cargo airlines - A mathematical programming approach". In: *OR Spectrum* 29.1, pp. 105–121.

Berry, S. and A. Pakes (2007). "The Pure Characteristics Demand Model". In: *International Economic Review* 48.4, pp. 1193–1225.

Brotcorne, L et al. (2008). "Joint design and pricing on a network". In: *Operations Research* 56.5.

Brotcorne, L. et al. (2011). "An exact algorithm for the network pricing problem". In: *Discrete Optimization* 8.2, pp. 246–258.

— (2012). "A Tabu search algorithm for the network pricing problem". In: *Computers and Operations Research* 39.11, pp. 2603–2611.

Brotcorne, L. et al. (2000). "A Bilevel Model and Solution Algorithm for a Freight Tariff-Setting Problem". In: *Transportation Science* 34.3, pp. 289–302.

Brotcorne, L. et al. (2001). "A Bilevel Model for Toll Optimization on a Multicommodity Transportation Network". In: *Transportation Science* 35.4, pp. 345–358.

Crainic, T. G. (2000). "Service network design in freight transportation". In: *European Journal of Operational Research* 122.2, pp. 272–288.

Crainic, T. G. and G. Laporte (1997). "Planning models for freight transportation". In: *European Journal of Operational Research* 97, pp. 409–438.

Crevier, B. et al. (2012). "Integrated operations planning and revenue management for rail freight transportation". In: *Transportation Research Part B: Methodological* 46.1, pp. 100–119.

Dobson, G. and S. Kalish (1988). "Positioning and pricing a product line". In: *Marketing Science* 7.2, pp. 107–125.

— (1993). "Heuristics for Pricing and Positioning a Product-Line Using Conjoint and Cost Data". In: *Management Science* 39.2, pp. 160–175.

FAA (2016). *Economic Values for Evaluation of FAA Investment and Regulatory Decisions.* Tech. rep. Federal Aviation Administration.

FedEx (2016). *FedEx Annual Report 2016.* Tech. rep. Memphis.

Green, P. E. and A. M. Krieger (1985). "Models and Heuristics for Product Line Selection". In: *Marketing Science* 4.1, pp. 1–19.

Hendel, G. (2011). "New Rounding and Propagation Heuristics for Mixed Integer Programming". PhD thesis. Technische Universität Berlin.

Hoff, A. et al. (2010). "A metaheuristic for stochastic service network design". In: *Journal of Heuristics* 16.5, pp. 653–679.

Jarrah, A. I. et al. (2009). "Large-Scale, Less-than-Truckload Service Network Design". In: *Operations Research* 57.3, pp. 609–625.

Kim, D. et al. (1999). "Multimodal Express Package Delivery: A Service Network Design Application". In: *Transportation Science* 33, pp. 391–407.

Kumar, R. and S. Vassilvitskii (2010). “Generalized Distances between Rankings”. In: *Proceedings of the 19th international conference on World wide web*. 3, pp. 571–579.

Labbé, M. et al. (1998). “A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing”. In: *Management Science* 44.12-part-1, pp. 1608–1622.

Li, L. and S. Tayur (2005). “Medium-Term Pricing and Operations Planning in Intermodal Transportation”. In: *Transportation Science* 39.1, pp. 73–86.

Lin, C. C. and S. C. Lee (2015). “Zone pricing for time-definite LTL freight transportation with elastic demand”. In: *Computers and Operations Research* 62, pp. 51–60.

Lin, C. C. et al. (2009). “Price planning for time-definite less-than-truckload freight services”. In: *Transportation Research Part E: Logistics and Transportation Review* 45.4, pp. 525–537.

McBride, R. D. and F. S. Zufryden (1988). “An Integer Programming Approach To the Optimal Product Line Selection Problem”. In: *Marketing Science* 7.2, pp. 126–140.

McGill, J. I. and G. J. Van Ryzin (1999). “Revenue management: Research overview and prospects”. In: *Transportation Sciencecience* 33.2, pp. 233–256.

Muhlenbein, H. and D. Schlierkamp-Voosen (1993). “Predictive Models for the Breeder Genetic Algorithm”. In: *Evolutionary Computation* 1.1, pp. 25–49.

Rochat, Y. and É. D. Taillard (1995). “Probabilistic diversification and intensification in local search for vehicle routing”. In: *Journal of Heuristics* 1.1, pp. 147–167.

Smilowitz, K. R. and C. F. Daganzo (2007). “Continuum Approximation Techniques for the Design of Integrated Package Distribution Systems”. In: *Networks* 50.3, pp. 183–196.

Smilowitz, K. R. et al. (2003). “Deferred item and vehicle routing within integrated networks”. In: *Transportation Research Part E: Logistics and Transportation Review* 39.4, pp. 305–323.

Song, M. (2007). “Measuring consumer welfare in the CPU market : an application of the pure-characteristics demand model”. In: 38.2, pp. 429–446.

Talluri, K. and G. J. Van Ryzin (2004). *The theory and practice of revenue management.*

UPS (2016). *UPS Annual Report 2016.* Tech. rep. Atlanta.

US Census Bureau (2009). *Quaterly e-commerce retail sales 4th quarter 2007.* Tech. rep.

— (2017). *Quaterly e-commerce retail sales 4th quarter 2016.* Tech. rep. US Census Bureau.

Wieberneit, N. (2008). “Service network design for freight transportation: A review”. In: *OR Spectrum* 30.1, pp. 77–112.

Wu, T.-H. (1997). “A note on a global approach for general 01 fractional programming”. In: *European Journal of Operational Research* 101.1, pp. 220–223.

Zurheide, S. and K. Fischer (2014). “Revenue management methods for the liner shipping industry”. In: *Flexible Services and Manufacturing Journal* 27.2-3, pp. 200–223.

A Further information on test instances

Routes and paths have been created a-priori as in Andersen et al. (2009, p. 203). Additionally we enforced the following requirements which should reflect real-world operational restrictions. In an express air network parcels are either transported directly from origin to destination or via a central hub. As such the use of aircraft precludes frequent transshipments and overly long transit times. We therefore enforce that a feasible path featuring aircrafts has a maximum duration of six periods and uses at most two aircraft. Parcels may wait for further transport at any hub, the waiting time, however, has to be uninterrupted. With a planning horizon of up to a week vehicles might not return to their starting point for an extended amount of time. In reality, however, vehicles have fixed home bases to which they need to return on a regular basis (e.g. to exchange crews). We therefore define the first physical node of a route as its home base and enforce that aircraft need to return to this location once every four periods and trucks once every eight periods. Real-world express air networks typically feature a central hub with a special status (e.g. for aircraft maintanance). In case of UPS it is located in Louisville and is called a global hub. We randomly select one of the physical nodes as the global hub and enforce that aircraft need to visit it once every six periods.

B Pseudo code

This detailed pseudo code should be sufficient to recreate our algorithm. Values for necessary parameters are given in square brackets.

Algorithm 1 performGA (executed asynchronously in parallel)

$\mathbf{P} \leftarrow \{\}$ ▷ population of price vectors
$\mathbf{S} \leftarrow \{\}$ ▷ population of associated market share vectors
$\mathbf{Q} \leftarrow \{\}$ ▷ matrix of route patterns
$popSize \leftarrow$ desired population size [50]
$numPatterns \leftarrow$ number of route patterns that are passed to worker [10000]
$numRoutes \leftarrow$ number of routes in fulfillment problem
$maxShareLS \leftarrow$ maximum share of local search in total evaluations 0.5]
$i \leftarrow 0$

while $elapsedTime < maxTime - polishTime$ **and** $i < maxIteration - 1$ **do**
    $i \leftarrow i + 1;$
    $r \leftarrow$ random number drawn from uniform distribution between 0 and 1
    if $|\mathbf{N}| < popSize$ **then**
        $\mathbf{p} \leftarrow doRandomPrices()$ ▷ draw prices from uniform distribution and sort decreasingly
    else if $\mathbf{P}_{LS} \neq \{\}$ **and** $r \leqslant maxShareLS$ **then**
        $\mathbf{p} \leftarrow$ price vector from local search pool $\mathbf{P}_{LS}$ ▷ remove this price vector from local search pool after selection
    else
        $\mathbf{p} \leftarrow doMateAndMutate(\mathbf{P}, \mathbf{S}, \mathbf{\Pi})$ ▷ see algorithm 2
    end if
    $(\mathbf{s}, \mathbf{a}) \leftarrow getMarketShares(\mathbf{p})$ ▷ determine customer decisions $\mathbf{a}$ and get market shares $\mathbf{s}$
    $\mathbf{p}_{max} \leftarrow getMaxPrices(\mathbf{a})$ ▷ determine maximal prices yielding this set of customer decisions

$R \leftarrow getRevenue(\mathbf{p})$ ▷ market shares, quantities and prices yield revenue
$R_{max} \leftarrow getRevenue(\mathbf{p}_{max})$ ▷ revenue achievable with maximal possible prices
$\mathbf{d} \leftarrow getDistances(\mathbf{s}, \mathbf{S})$ ▷ see algorithm 3
for $j \leftarrow 1$ to $numRoutes$ **do**
  $\omega_j \leftarrow \sum_{k=1}^{i} (y_j)_k d_k$ ▷ sum weighted solution values y_j of all evaluated individuals $k = 1, \dots, i$
end for
sort all patterns in $\mathbf{Q}$ according to distance d_k of the generating individual k relative to the current individual i
$\mathbf{Q}_i \leftarrow \mathbf{Q}(1 : numPatterns)$
$(C, \mathbf{Q}_i^{new}) \leftarrow solveFP(\mathbf{s}, \omega, \mathbf{Q}_i)$ ▷ cost C is derived by algorithm 4
$\mathbf{Q} \leftarrow [\mathbf{Q}, \mathbf{Q}_i^{new}]$
$\Pi \leftarrow R - C$ ▷ calculate profit Π and maximal possible profit Π_{max}
$\Pi_{max} \leftarrow R_{max} - C$
if $|\mathbf{P}| = popSize$ **then**
  kill one individual by tournament selection based on Π
end if
$\mathbf{P} \leftarrow [\mathbf{P}, \mathbf{p}]$
$\mathbf{S} \leftarrow [\mathbf{S}, \mathbf{s}]$
if $\Pi_{max} > \Pi^*_{max}$ **then**
  $\Pi^*_{max} \leftarrow \Pi_{max}$
  $\mathbf{p}_{LS} \leftarrow performLocalSearch(\mathbf{p}, \mathbf{s})$ ▷ see algorithm 6
  $\mathbf{P}_{LS} \leftarrow [\mathbf{P}_{LS}, \mathbf{p}_{LS}]$
end if
end while
$doSolutionPolishing()$ ▷ try to further improve best solution by using 4 for some more time
return Π^*_{max}

Algorithm 2 doMateAndMutate

$\mathbf{P} \leftarrow$ population of price vectors $\langle p_1, p_2, \dots, p_l \rangle$
$\mathbf{S} \leftarrow$ population of market share vectors $\langle s_1, s_2, \dots, s_l \rangle$
$\mathbf{\Pi} \leftarrow$ profits
$t \leftarrow$ tournament size [2]
$c \leftarrow$ positive value which determines how far long the line a child can be located [0.25]
$iterationCount \leftarrow$ GA iteration counter
$minPrice \leftarrow 0$
$maxPrice \leftarrow$ maximum value among all customers' utilities
$\sigma^0 \leftarrow$ initial standard deviation for mutation [maxPrice / 8]
$bestImproved \leftarrow$ binary vector indicating which iteration improved best objective
$windowSize \leftarrow$ number of iterations for exponential moving average [500]
$minEMA \leftarrow$ lower limit for EMA, controls when standard deviation should be increased [0.001]

$parent(1) \leftarrow$ individual picked at random from $\mathbf{P}$ with replacement
for $i \leftarrow 2$ to t **do** ▷ pick mother according to fitness
  $next \leftarrow$ individual picked at random from $\mathbf{P}$ with replacement
  if $\Pi_{next} > \Pi_{parent(1)}$ **then**
    $parent(1) \leftarrow next$
  end if
end for
$\mathbf{s}_{parent(1)} \leftarrow$ market share vector associated to this price vector
$\mathbf{d} \leftarrow getDistances(\mathbf{s}_{parent(1)}, \mathbf{S})$ ▷ see algorithm 3

$parent(2) \leftarrow$ individual picked at random from $\mathbf{P}$ with replacement
for $j \leftarrow 2$ to t **do** ▷ pick $parent(2)$ according to distance from $parent(1)$
  $next \leftarrow$ individual picked at random from $\mathbf{P}$ with replacement
  if $d_{next} > d_{parent(2)}$ **then**
    $parent(2) \leftarrow next$
  end if
end for
$m \leftarrow$ random integer from 1 to 2 ▷ perform crossover by Extended Intermediate Recombination
$\mathbf{p} \leftarrow parent(m)$ ▷ offspring is based on randomly selected parent
$n \leftarrow$ random integer from 1 to l
$shuffle \leftarrow$ random permutation of integers 1 through l
$recombine \leftarrow shuffle(1:n)$ ▷ these are the prices to cross over
for each service s of market share vector $\mathbf{s}$ **do**
  if $s \in recombine$ **then**
    if $s = 1$ **then** ▷ determine bounds (new prices need to form a decreasing sequence)
      $lb \leftarrow \max\{p^{s+1}, minPrice\}$
      $ub \leftarrow maxPrice$
    else if $s = l$ **then**
      $lb \leftarrow minPrice$
      $ub \leftarrow \min\{p^{s+1}, minPrice\}$
    else
      $lb \leftarrow \max\{p^{s+1}, minPrice\}$
      $ub \leftarrow \min\{p^{s-1}, maxPrice\}$
    end if
    $\alpha \leftarrow uniformRandom(-c, 1+c)$
    $p^s \leftarrow mother^s\alpha + father^s(1-\alpha)$
  end if
  $EMA \leftarrow$ exponential moving average of $bestImproved$ with window size $windowSize$
  if $iterationCount \geqslant windowSize$ **then**
    $\sigma \leftarrow \sigma^0$
  else if $EMA < minEMA$ **then**
    $\sigma^s \leftarrow$ standard deviation of prices in population for service s
    $\sigma^s \leftarrow (1 - \frac{EMA}{minEMA})\sigma^0 + \sigma^s$ ▷ gradually increase standard deviation if GA is unsuccessful
  else
    $\sigma^s \leftarrow$ standard deviation of prices in population for service s
  end if
  $u \leftarrow$ random number drawn from uniform distribution between 0 and 1
  if $u \geqslant rateMutation$ **then**
    $v \leftarrow$ random number drawn from standard normal distribution
    $p^s \leftarrow p^s + v\sigma^s$
  end if
  if $p^s < lb$ **then** ▷ if new price is invalid then clip it to violated bound
    $p^s \leftarrow lb$
  end if
  if $p^s > ub$ **then**
    $p^s \leftarrow ub$
  end if
end for
return p

Algorithm 3 getDistances

$\mathbf{s} \leftarrow$ vector of new market shares $\langle s_1, s_2, \ldots, s_m \rangle$
$\mathbf{S} \leftarrow$ array of previous (unique) market shares $\langle S_1, S_2, \ldots, S_n \rangle$

for $i \leftarrow 1$ to n **do** ▷ outer loop can be replaced by vector operations in inner loops
  for $j \leftarrow 1$ to m **do**
    $d_{ij} \leftarrow 0$
    $\delta_{ij} \leftarrow \mathbf{S}_{ij} - \mathbf{s}_j$
    for $k \leftarrow 1$ to m **do**
      if $j < k$ **then**
        if δ_{ij} and δ_{ik} have opposite signs **then**
          $r_{ij} \leftarrow \min\{|\delta_{ij}|, |\delta_{ik}|\}$
          if $\delta_{ij} > 0$ **then**
            $\delta_{ij} \leftarrow \delta ij - r_{ij}$
          else
            $\delta_{ij} \leftarrow \delta ij + r_{ij}$
          end if
          $d_{ij} \leftarrow d_{ij} + \log(1 - j + k)\, r_{ij}$
        end if
      end if
    end for
  end for
  $d_i \leftarrow \sum_{j=1}^{m} d_{ij}$
end for
return d

Algorithm 4 solveFP

$\mathbf{s} \leftarrow$ current market shares $\langle s_1, s_2, \ldots, s_m \rangle$
$\mathbf{s}_{sim} \leftarrow$ market share vector of most similar individual
$\mathbf{v}_{sim} \leftarrow$ solution of most similar individual
$\mathbf{a} \leftarrow$ vector of objective coefficients
$\omega \leftarrow$ vector of usage statistics for each route variable
$\mathbf{Q} \leftarrow$ array of route patterns
$c \leftarrow 0$

if $\mathbf{v}_{sim} \neq \{\}$ **then** ▷ build initial solution from most similar individual
  if $\mathbf{s} = \mathbf{s}_{sim}$ **then**
    $\mathbf{v}^* \leftarrow \mathbf{v}_{sim}$
  else
    $\mathbf{v}^* \leftarrow$ fix chosen routes in $\mathbf{v}_{sim}$, solve LP, round up and fix route variables, resolve LP
  end if
else ▷ build initial solution from rounded up LP if no similar solution exists
  $\mathbf{v}^* \leftarrow$ solve LP, round up and fix route variables, resolve LP
end if
repeat
  $c \leftarrow c + 1$
  if $c \leqslant 3$ **or** $\mod(c, 50) = 0$ **then**
    $\mathbf{r}_{rep} \leftarrow$ all currently used aircraft routes are replacement candidates
  else
    $\mathbf{r}_{rep} \leftarrow getReplacementCandidates(\omega)$ ▷ tournament selection based on usage statistic ω
  end if
  $\mathbf{r}_{sub} \leftarrow getSubstitutionCandidates(\mathbf{r}_{rep})$ ▷ see algorithm 5

$\hat{\mathbf{Q}} \leftarrow$ route patterns that match current solution
if $\hat{\mathbf{Q}} \neq \{\}$ **then**
$\left(\hat{\mathbf{Q}}_{rep}, \hat{\mathbf{Q}}_{sub}\right) \leftarrow$ select a number of patterns by tournament selection based on previous cumulative objective improvement
$\mathbf{r}_{rep} \leftarrow \left[\mathbf{r}_{rep}, \hat{\mathbf{Q}}_{rep}\right]$
$\mathbf{r}_{sub} \leftarrow \left[\mathbf{r}_{sub}, \hat{\mathbf{Q}}_{sub}\right]$
end if
$\tilde{\mathbf{v}} \leftarrow$ solve sub-MIP with all routes fixed except replacement and substitution candidates
if $\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{a} < \mathbf{v}^{*\mathrm{T}}\mathbf{a}$ **then** ▷ if cost is lower for new solution adopt it
$\mathbf{v}^* \leftarrow \tilde{\mathbf{v}}$
$\tilde{\mathbf{Q}} \leftarrow$ changes from previous to current best solution define a new route pattern
$\mathbf{Q}^* \leftarrow \left[\mathbf{Q}^*, \tilde{\mathbf{Q}}\right]$ ▷ append new route pattern to output
end if
until either number of consecutive non-improving iterations or time reached limit
return $\mathbf{v}^*, \mathbf{Q}^*$

Algorithm 5 getSubstitutionCandidates

$\mathbf{r}_{rep} \leftarrow$ vector of route replacement candidates
$\mathbf{r}_{sub} \leftarrow \{\}$
$\bar{\mathbf{A}} \leftarrow$ adjusted coefficient matrix
$n \leftarrow$ number of substitution candidates per replacement candidate [10]
$t \leftarrow$ consecutive number of times this instance of FP was not improved by heuristic
$t_{max} \leftarrow$ maximum allowed number of non-improving consecutive heuristic runs [5]
$\phi^A, \phi^T \leftarrow$ required mode shares for selection of substitution candidates [0.5,0.8]
$AA, AT, TT, TA \leftarrow$ initial values for minimium ratio of shared rows, specific for air-air, air-truck, truck-truck, truck-air (necessary due to different number of arcs in air and truck routes) [0.66,0.33,0.66,0.99]

for each route $j \in \mathbf{r}_{rep}$ **do**
$\mathbf{E}_j \leftarrow$ binary vector indicating if an arc is in route j
$\gamma_j \leftarrow \frac{\bar{\mathbf{A}}^{\mathrm{T}}\mathbf{E}_j}{\bar{\mathbf{A}}_j^{\mathrm{T}}\mathbf{E}_j}$ ▷ gamma is the ratio of shared rows in adjusted coefficient matrix, it gets
$\alpha \leftarrow \left(1 - \frac{t}{t_{max}}\right)$ ▷ minimum ratio gets lowered with each non-succesfull heuristic run until selection of substitution candidates if fully random
if route j is an aircraft route **then**
$\mathbf{\Gamma}_j^{AA} \leftarrow$ all aircraft routes whos γ_j is bigger than $\alpha(AA)$
$\mathbf{\Gamma}_j^{AT} \leftarrow$ all truck routes whos γ_j is bigger than $\alpha(AT)$
$\mathbf{r}_{sub}^j \leftarrow$ random sample of $n\phi^A$ air routes and $n\left(1 - \phi^A\right)$ truck routes
else
$\mathbf{\Gamma}_j^{TT} \leftarrow$ all truck routes whos γ_j is bigger than $\alpha(TT)$
$\mathbf{\Gamma}_j^{TA} \leftarrow$ all aircraft routes whos γ_j is bigger than $\alpha(TA)$
$\mathbf{r}_{sub}^j \leftarrow$ random sample of $n\phi^T$ truck routes and $n\left(1 - \phi^T\right)$ air routes
end if
$\mathbf{r}_{sub} \leftarrow \left[\mathbf{r}_{sub}, \mathbf{r}_{sub}^j\right]$
end for
return $\mathbf{r}_{sub}$

Algorithm 6 performLocalSearch

$\mathbf{p} \leftarrow$ input price vector
$\mathbf{s} \leftarrow$ input market share vector
$\mathbf{a} \leftarrow$ binary array indicating service that maximizes net utility for each customer
$\mathbf{U} \leftarrow$ utilities of customers (scaled by customers' price-sensitivity)
$N \leftarrow$ maximum number of customers to be switched to other service [5]

$n \leftarrow$ draw random integer from 1 to N
$\mathbf{U}_{net} \leftarrow \mathbf{U} - \mathbf{p}$
$slack \leftarrow \mathbf{U}_{net} - \mathbf{U}_{net}(\mathbf{a})$
for $i \leftarrow 1$ to n **do**
    for each service s **do**
        $\mathbf{I}_i^s \leftarrow$ first i customers whos $slack^s$ turns negative if price p^s is increased
        $\mathbf{D}_i^s \leftarrow$ first i customers whos $slack^s$ turns positive if price p^s is decreased
        $\mathbf{A}_i^s \leftarrow [\mathbf{I}_i^s, \mathbf{D}_i^s]$ ▷ new assignment if price p^s is changed so that i customers switch
        $\mathbf{A}_i \leftarrow [\mathbf{A}_i, \mathbf{A}_i^s]$
    end for
    $\mathbf{A} \leftarrow [\mathbf{A}, \mathbf{A}_i]$ ▷ array of new assignments
end for
for each new customer assignment $\mathbf{a} \in \mathbf{A}$ **do**
    $\mathbf{p}_{new} \leftarrow getMaxPrices(\mathbf{a})$ ▷ see Dobson and Kalish (1988) for a detailed explanation
    $\mathbf{p}_{LS} \leftarrow [\mathbf{p}_{LS}, \mathbf{p}_{new}]$
end for
return $\mathbf{p}_{LS}$