European Journal of Operational Research 215 (2011) 325–336

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

# A lookahead partitioning heuristic for a new assignment and scheduling problem in a distribution system

Michal Tzur*, Ehud Drezner

*Industrial Engineering Department, Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel*

ARTICLE INFO

*Article history:*
Received 8 July 2010
Accepted 12 June 2011
Available online 16 June 2011

*Keywords:*
Heuristics
Distribution systems
Assignment
Routing
Scheduling

ABSTRACT

We introduce a new assignment and scheduling problem in a distribution system, which we refer to as the ASTV problem: Assigning and Scheduling transportation Tasks to Vehicles. In this problem, commodities need to be delivered directly from their origins to their destinations within specified time windows, using a fleet of homogenous capacitated vehicles. A set of routes, each of which performs one or several direct deliveries, need to be constructed such that the operational costs, including vehicle fixed cost, variable traveling and variable waiting costs, are minimized. The problem arises, for example, when delivering food products from several factories, where they are manufactured, to several distribution centers, from which they are delivered to the final customers. We define the problem and describe its relationship to existing problems studied in the literature, in particular pickup and delivery, assignment and scheduling problems. Subsequently we develop a solution method which is based on decomposing (partitioning) the ASTV problem into two interdependent sub-problems. The first consists of Assignment of Tasks to origin–destination full-load Trips (ATT), while the second determines assignment and Scheduling of these Trips to Vehicle routes (STV). We use a bi-criterion objective function in the first problem, whose purpose is to connect the two problems by looking ahead to the rest of the decisions, determined in the second problem. Thus, the solution method is referred to as *lookahead partitioning*. In this way, decisions of the first problem determine a favorable input for the second problem, which is solved last. An extensive numerical study was conducted to evaluate the performance of the overall heuristic method. The results indicate that our heuristic method is quite efficient.



## 1. Introduction

About 40–50% of the logistic costs in a distribution system can be attributed to transportation costs. Therefore, the financial potential of improving transportation-related decisions in a distribution system can be significant. This paper addresses these decisions in a distribution system with a set of characteristics that has not been studied together in previous research.

The problem is described by the following sets and parameters: a set of factories and a set of distribution centers; a fleet of homogenous capacitated vehicles, each located at one of the distribution centers; a set of tasks, each characterized by an origin (a specific factory), a destination (a specific distribution center), weight, volume, and a time window for its execution; parameters for the operational costs: vehicles fixed cost and vehicles variable traveling and waiting costs. The objective is: construct a set of vehicle routes such that each vehicle delivers immediately and directly one or several tasks from their origin to their destination within their ascribed time window, such that the vehicles weight and volume capacities are not exceeded and the total operational costs are minimized.

We denote this new problem as the Assignment and Scheduling of transportation Tasks to Vehicles (ASTV) problem. It has two important characteristics whose combination has not been studied before: first, the requirement for a direct delivery of each task from its origin to its destination, i.e., no transshipment is allowed through any other facility, and second, delivering less-than-truck-load (LTL) tasks, i.e., goods that are small relative to the capacity of the vehicle. The requirement for a direct delivery is aligned with the goal of delivering the tasks quickly, as required for certain food products. As opposed to the situation in classical pickup and delivery problems, it means that once a vehicle visited a certain pickup location, it must continue directly to the delivery location of the tasks picked up, see below a detailed comparison between the problems. Direct deliveries also exclude the common practice of LTL carriers which use sorting facilities (breakbulks) and smaller terminals (end-of-lines), see Jarrah et al. (2009). Thus, the ASTV problem involves decisions concerning assignment (of goods to vehicles), routing (sequence of points visited by each vehicle) and scheduling (time in which vehicles need to leave and arrive to various points). The problem is NP-hard in the strong

* Corresponding author. Tel.: +972 3 6407420; fax: +972 3 6407669.
*E-mail addresses:* tzur@eng.tau.ac.il (M. Tzur), udrezner@gmail.com (E. Drezner).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.06.013

sense as several special cases of it are NP-hard in the strong sense. Although it is similar in some dimensions to routing and/or transportation problems discussed in the literature, existing solution methods cannot be used to solve it due to its unique characteristics and since it involves all three problem dimensions mentioned above. Moreover, while the objective in most existing problems is to minimize traveling costs only, our objective is to minimize total costs which includes additional components related to fixed costs and waiting times. Hence, new formulation and solution method need to be developed.

The ASTV problem is mostly related to two other problems addressed in the literature: the *pickup and delivery problem with time windows* (PDPTW) and the *multi depot vehicle scheduling problem* (MDVSPTW). These and other related problems are discussed in the Literature Review section. However, to complete the description of the ASTV problem, we demonstrate here the connection and differences between ASTV and each of these problems, see Figs. 1 and 2. In Fig. 1a, a typical vehicle route in LTL pickup and delivery applications, such as the PDPTW (or the more general GPDPTW, see Savelsbergh and Sol, 1995), is illustrated. In this route the vehicle leaves the depot, continues in the pickup of several LTL tasks, delivers the tasks in a certain order to their delivery locations, and returns to the depot. However, in applications where a task must be delivered directly from its origin to its destination, as in the case of certain food products deliveries, such routes are forbidden. Direct deliveries are also obligated in vehicle scheduling and in FTL pickup and delivery applications. In those cases, the vehicle starts from the depot, travels to pickup a full load task from its origin and delivers it to its destination. After performing one or several such trips, the vehicle returns to its depot, as illustrated in Fig. 1b.

In the ASTV problem, the vehicles are housed at the destinations (DCs) of the tasks and a large number of LTL tasks need to be delivered between each origin–destination pair. Then, most vehicle trips are round-trips, i.e., they start at the DC, continue in commodity pickup at the factory, and finish in commodity delivery to the DC. Several such trips are combined into a vehicle route, see Fig. 1c. Exceptions may be made when trips destined to different DCs are combined together in the same vehicle route in order to reduce the fixed cost component, in which case the routes involve more than just one DC. This is described in the second phase of the algorithm that solves the second sub-problem (Section 4.3). Note that in this case too, the deliveries are still direct. Since it is a LTL setting, the assignment of tasks to vehicle trips has to be determined on top of the routing and scheduling decisions, thus enlarging the solution space of the problem even further. The connection between the three problems discussed above, in terms of the solution space, is illustrated in Fig. 2. As mentioned earlier, the objective function of the ASTV problem is also generalized, since it includes fixed and waiting costs in addition to the typical traveling distance.

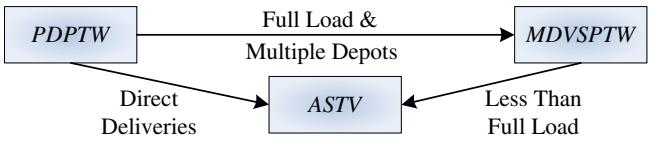


**Fig. 2.** The connection between existing research streams and the ASTV problem.

This paper makes the following contributions. First, it introduces the above described new problem and associated application, and demonstrates its relationship to two research streams in the literature. The second contribution is the design of a heuristic solution method to solve the problem, including detailed procedures for two sub-problems of it. Our method is based on a new decomposition idea which we refer to as *lookahead partitioning*. The partitioning is *along decision types*, such that when solving for one type, subsequent decisions and their associated costs are also accounted for. This helps reducing the sub-optimality caused by the decomposition. It may be applicable to other problems in which decisions naturally decompose into two (or more) types, and where the associated problems may be solved sequentially (in a heuristic manner). Our third contribution is related to the results obtained by our solution method, based on a comprehensive numerical study. The study demonstrates the efficiency of our solution method by comparing its cost to a lower bound on the total cost of the problem, and reveals insights on properties of the solutions obtained. Finally, our problem definition is inspired by a real application. Unfortunately, our results could not be applied to that application due to administrative difficulties. Nevertheless it has a great potential to be applicable to problems with similar settings.

The rest of the paper is organized as follows. In Section 2 we review related literature. Notation and description of the general solution approach are given in Section 3. In Section 4 the detailed solution method for the ASTV problem is presented. In Section 5 we derive a lower bound on the total operational cost of an ASTV feasible solution. Numerical results for evaluating our proposed solution method and for gaining insights are presented in Section 6. Conclusions are provided in Section 7.

## 2. Literature review

In this section we review related literature and its evolution from basic problems to the one discussed in this paper. The ASTV problem extends and connects previous work from two research streams: *vehicle routing* and *vehicle scheduling*. A vehicle *route* refers to a sequence of pickup and/or delivery points which the vehicle traverses, while a vehicle *schedule* is a sequence of pickup and/or delivery points together with associated arrival and departure times, see Bodin and Golden (1981). Combined vehicle routing and scheduling problems include both routing and scheduling

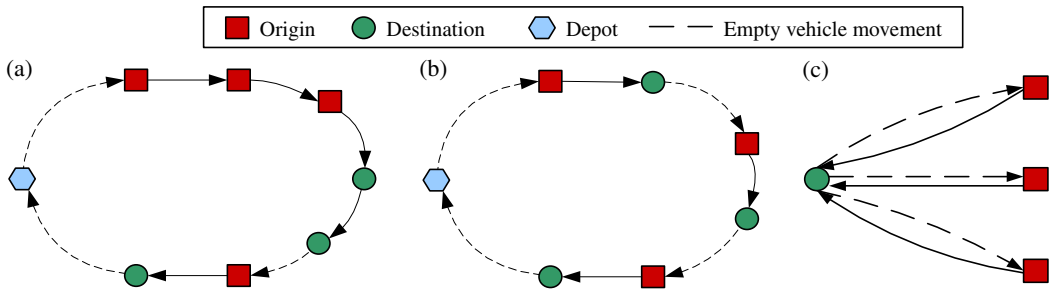


**Fig. 1.** (a) A typical vehicle route in less-than-full-truck-load pickup and delivery applications. (b) A typical vehicle route in vehicle scheduling and in full load pickup and delivery applications. (c) A typical vehicle route in a less-than-full-truck-load and direct delivery applications.

decisions, and arise in many real-world applications which include, for example, time windows and/or precedence relationships. In these problems routes are designed to minimize total transportation costs, while assuring feasibility of their schedule. We show that the ASTV problem extends general forms of two well known and important problems: the pickup and delivery problem, and the vehicle scheduling problem.

The *pickup and delivery problem* (PDP) can be described as follows (Savelsbergh and Sol, 1995): a set of known transportation requests has to be satisfied by a given fleet of vehicles. Each request is characterized by its pickup location (origin), its delivery location (destination) and the size of the load that has to be transported between them. The load capacity, the maximum route length, a start location and an end location are given for each vehicle. To fulfill all requests, a set of routes has to be planned such that each request is transported from its origin to its destination by exactly one vehicle, although not necessarily directly. Typical objective functions are minimum number of vehicles employed, total distance traveled, total schedule duration or combination thereof. The *PDP with time windows* (PDPTW) is a PDP in which a time window is specified for each pickup and delivery location. Integer programming (IP) formulations of the PDPTW have been presented by Dumas et al. (1991) and by Savelsbergh and Sol (1995). Two different branch and cut methods for the problem were recently developed by Lu and Dessouky (2004) and by Ropke et al. (2007). Recent heuristic methods include, for example, the work of Pankratz (2005) and Bent and Van Hentenryck (2006). Recent surveys on the PDP and its applications include, for example, Toth and Vigo (2002), Berbeglia et al. (2007) and Cordeau et al. (2008). The full-truck-load (FTL) PDP arises when the load of the vehicle is given and it has to be transported directly from its origin to its destination. Similarly, FTL PDPTW (when time windows exist) and multi depot FTL PDPTW may be defined. A detailed description of various versions of the problem can be found in Desrosiers et al. (1988).

The other related well known problem is the *vehicle scheduling problem* (VSP), which can be described as follows: given a depot housing a set of vehicles, a set of trips characterized by an origin, a destination and fixed starting and ending times, and given the traveling times between all pairs of locations, find a feasible minimum-cost schedule such that each trip is assigned to exactly one vehicle, and each vehicle performs a feasible sequence of trips, starting from and ending at the depot. The duration of a trip may include not only the travel time but also loading, unloading and any idle time that may occur between them. When there are several depots, each housing a set of vehicles, the problem is referred to as the *multi depot VSP* (MDVSP), which is known to be NP-hard (Bertossi et al., 1987). In the presence of time windows the problem is referred to as the *MDVSP with time windows* (MDVSPTW), see Desaulniers et al. (1998). In the MDVSPTW the starting time of service is defined by a time window rather than a single point in time, which introduces the element of routing into the pure scheduling problem. No fast and efficient exact method exists in the literature for this problem. Heuristic algorithms include, for example, Potvin and Rousseau (1993), Brãndao and Mercer (1997), Tung and Pinnoi (2000) and Dondo and Cerdá (2007). A survey of vehicle scheduling problems including numerous additional references can be found in Desrosiers et al. (1995).

We observe here, that the MDVSPTW and the multi depot FTL PDPTW are identical problems, an observation which has not been made before. This may be due to the distinction that exists between the problems when there are no time windows, namely, that associated starting and ending times of each trip are given in the MDVSP but not in the FTL PDP. As shown in Fig. 2, the ASTV problem is related to these problems by incorporating both LTL and direct delivery requirements, a framework that has not been discussed in the literature.

Finally, we briefly mention decomposition methods that are often used for solving NP-hard problems heuristically. General decomposition methods for large-scale mathematical programming problems include, e.g., Lagrangean relaxation or Danzig–Wolf and Benders decompositions. Despite their generality, a considerable effort still has to be invested in applying them to a specific application, see Sandhu and Klabjan (2007). Other decomposition methods may be ad hoc for a specific application, for example, the well known partitioning (of the plane) algorithm for the TSP by Karp (1977) or the partitioning (of time) algorithm of Federgruen and Tzur (1999) for multi-echelon dynamic lot-sizing problems. Our method is new and is based on a lookahead principle, while decomposing the problem along decision types.

## 3. Problem description

An ASTV problem instance is characterized by the following sets and parameters:

Sets and general parameters:

- $D$ set of destinations of tasks (DCs)
- $R$ set of origin–destination (of tasks) pairs
- $S^r$ set of transportation tasks to be delivered between origin–destination pair $r \in R$
- $S$ Set of transportation tasks to be delivered between all origin–destinations pairs in $R$ i.e., $S = \cup_{r \in R} S^r$

Note that we refer to origins and destinations (factories and distribution centers (DCs), respectively) as they relate to tasks. Vehicles, on the other hand, start their trip from a DC and return to it. The latter determination is not crucial for the solution method and was chosen according to the practice in the application that motivated this research.

- $T$ length of the planning period, in time slots

Task parameters:

- $w_s$ weight of the commodity delivered in task $s \in S$
- $v_s$ volume of the commodity delivered in task $s \in S$
- $l_s$ earliest pickup time of task $s \in S$ at the origin
- $u_s$ latest delivery time of task $s \in S$ at the destination
- $[l_s, u_s]$ is referred to as the time window for executing task $s \in S$

Vehicle parameters: (all vehicles are identical).

- $Q^W$ vehicle weight capacity
- $Q^V$ vehicle volume capacity
- $V_i$ number of vehicles available in DC $i \in D$
- $t_r$ vehicle traveling time between the origin–destination pair $r \in R$
- $t_{ij}$ vehicle traveling time between DCs $i$ and $j$ ($i,j \in D$)
- $\tau$ Commodity loading plus unloading time (identical for each vehicle trip)

Cost parameters:

- $f$ fixed vehicle cost incurred for performing a route
- $\lambda^a$ traveling cost incurred per time slot in which the vehicle is in movement
- $\lambda^w$ waiting cost incurred per time slot in which the vehicle is waiting for and during loading and unloading commodities

We note that both weight and volume constraints are important. For example, in the food industry, the weight constraint is likely to be effective when dairy products are concerned, while the volume constraint is likely to be effective when dry food products are concerned.

The following assumptions are used:

1. The triangle inequality holds for all traveling times.
2. All traveling times are symmetric.
3. Time parameters are specified in units of a basic time slot and therefore are integers.
4. A task has to be delivered directly from its origin to its destination, that is, commodity transfers between factories or between DCs (transshipments) are forbidden.
5. A task may not be split, that is, it should be loaded on one vehicle only.

We define a vehicle *trip* as a delivery of one or several tasks from their origin to their destination in the same vehicle. All tasks in a trip are required to have identical origins and destinations, due to the direct delivery requirement. We further define a vehicle *route* as a sequence of trips, along with the segments between the trips, which the vehicle performs, starting and ending at the same DC. The ASTV problem consists of finding a set of vehicle routes, each of which starts from and ends at the vehicle's DC and includes one or more trips. Commodities (one or more tasks) need to be assigned to each trip such that they do not exceed the vehicle's weight and volume capacities, all tasks are delivered from their origin to their destination within their ascribed time windows, and the sum of fixed, traveling and waiting costs of all routes is minimized.

While it is possible to formulate the ASTV problem as a mixed integer linear program, the resulting formulation is extremely large and cumbersome and we did not find it useful to solving the problem. We use, however, formulations of sub-problems of the problem in various parts of the solution method or as part of the lower bound development.

The ASTV problem is NP-hard in the strong sense, as special cases of it are NP-hard in the strong sense. Consider, for example, the special case with a single DC, only fixed costs and where it is already determined which commodities (tasks) are loaded on each vehicle. In this case it only remains to determine the vehicles' routes and schedule so that the number of vehicles used is minimized. This problem can be shown to be equivalent to the known NP-hard scheduling problem referred to as *Scheduling with Release times and Deadlines on Minimum number of Machines* (SRDM), which was studied, for example, by Chuzhoy and Naor (2004). The objective in the SRDM problem is to schedule a given set of jobs which have release and deadline times on a minimum number of parallel machines. Another way to establish the complexity of the ASTV problem is by considering the special case with a single DC, a single factory and only traveling costs. The problem reduces to deciding on the set of tasks to be loaded on each vehicle since this determines the trips of the vehicles, which determines the total traveling costs. It can be shown that minimizing the number of trips is equivalent to the well known NP-hard *bin packing* problem (Garey and Johnson, 1979).

Indeed, while the problem consists of three types of decisions: assignment, routing and scheduling, the last two are closely related since the routing decisions have to obey the scheduling constraints. Hence, our solution approach initially separates the two types of decisions, namely: assigning tasks to vehicles and creating vehicle routes with associated schedules. Even though each of the resulting problems is still NP-hard, the complexity of the decomposed problem is reduced, which motivates this approach. It still remains a major challenge to solve the separate problems in synergy and overcome the sub-optimality caused by the decomposition action. Our solution method is designed to do exactly that. The two resulting problems and the interaction between them are described in the next section.

## 4. Solution approach

In Section 4.1 we outline our overall solution approach. In Sections 4.2 and 4.3 we specify our detailed solution method to each of the resulting problems, and discuss their integration.

### 4.1. Overall solution approach

Recall that a *trip* is a delivery of one or several tasks from their origin to their destination in the same vehicle, and a *route* is a sequence of trips, along with the segments between the trips which the vehicle performs, starting and ending at the same DC. Thus, the ASTV problem is to arrange all tasks into a set of trips and to integrate them into a set of routes. These problems are referred to as:

1. Assignment of Tasks to vehicle Trips (ATT).
2. Scheduling of Trips to Vehicle routes (STV).

The ATT problem is stated as follows: assuming an unlimited number of vehicles at each DC (this assumption is removed when solving the STV problem), and given a set of tasks, each characterized by an origin, a destination, weight, volume, and a time window for its execution, assign tasks to vehicle trips such that the vehicles' weight and volume capacities are not exceeded, each task is delivered directly from its origin to its destination within its ascribed time window, and the total assignment cost is minimized.

However, it is unclear how to account for the assignment cost term. In the ATT problem, the assignment of tasks to trips determines directly only the traveling cost component. Minimizing the traveling costs is achieved by minimizing the number of trips, which is equivalent to maximizing the vehicles' capacity utilization (see Lemma 1 in Section 4.2). However, the assignment of tasks to trips also determines the time window of each trip (derived from the time windows of all tasks assigned to it), and consequently the ability to combine the trips into routes (in the STV problem). This, in turn, determines the fixed and waiting costs of the problem. Hence it is required to account for the indirect effect of the assignment decisions on the latter cost components, but we are not aware of a way to compute it precisely without actually solving the STV problem. Moreover, it is expected that some assignment solutions that are attractive in terms of minimizing the number of trips, (i.e., highly utilized trips), are quite unattractive in terms of minimizing the fixed and waiting costs determined subsequently, because those trips may have tight time window constraints. Therefore we take the approach of including in the ATT objective function two expressions which attempt to balance between all cost components, namely, those that are incurred immediately, and those that are likely to be incurred subsequently. In this way, the lookahead partitioning approach is implemented.

Specifically, in the ATT problem we define a multi-objective function, where the two objective terms which we aim to minimize are: non-utilized capacities of trips and reduction in time window flexibility (as a result of task assignments) of trips. High trips utilizations lead to fewer number of trips, and therefore to low traveling costs; high time window flexibilities enable a more efficient construction of routes in the STV problem, and therefore to lower fixed and waiting costs. The utilization and flexibility terms are presented and discussed in Section 4.2 where the solution method for the ATT problem is described.

Based on the set of vehicle trips determined by the ATT problem, the STV problem is stated as follows: given a set of trips, each characterized by an origin, a destination and a time window for its execution, arrange the trips in vehicle routes and determine their schedule such that each vehicle starts and ends in its housing DC and the sum of fixed and waiting costs is minimized. Solving the STV problem is discussed in Section 4.3.

### 4.2. Solving the ATT problem

In this section we describe our solution method to the ATT problem. We first claim that it may be further decomposed by origin–destination pairs, and develop in detail the bi-criterion objective function for a given pair. Subsequently we describe a Tabu Search (TS) procedure to solve each of the resulting problems.

As mentioned in the problem description, tasks can only be combined in the same vehicle trip if they share the same origin and destination (the direct delivery requirement). In addition, given the triangle inequality assumption with respect to the traveling times between all locations involved, and the requirement that a vehicle returns to its housing DC, the traveling cost associated with a set of tasks delivered between a given origin–destination pair is minimized when it is delivered by a vehicle located at the destination of those tasks. Thus, given that enough vehicles are available at each DC, solutions of the ATT problem are separable by origin–destination pairs, without loss of optimality. In cases where the vehicle availability assumption is violated in the ASTV problem, this constraint is treated in the STV problem by adjusting the vehicle routes accordingly.

Therefore we denote the problem for a given origin–destination pair $r \in R$ by $ATT_r$ and develop a solution method to solve the problem for a given $r$. We start with a few definitions which describe the parameters of each trip based on the parameters of the tasks assigned to it.

Recall that $S^r$ denotes the set of tasks to be delivered between origin–destination pair $r \in R$. Let $S_j^r \subseteq S^r$ denote the set of tasks assigned (in the solution) to trip $j \in J^r$ where $J^r$ is the set of trips generated in the solution to $ATT_r$, $r \in R$, and $j \in J^r$ is a trip in this set. Similar to the definitions of the earliest pickup and latest delivery times of tasks, we define those terms for trips. For a given $r \in R$, let (we omit the dependency on $r$ when no ambiguity occurs):

$L_j$ = Earliest pickup time at the origin of tasks assigned to trip $j \in J^r$;
$U_j$ = Latest delivery time at the destination of tasks assigned to trip $j \in J^r$, where:

$$L_j = \max\{l_s : s \in S_j^r\}, \tag{1}$$

$$U_j = \min\{u_s : s \in S_j^r\}. \tag{2}$$

Let also,
$T_j = t_r + \tau$ for $j \in J^r$ denote the execution time of trip $j$, which consists of the traveling time between the origin and the destination ($t_r$) plus the loading and unloading time of the vehicle ($\tau$). With these definitions, a trip $j$ is time-window feasible iff:

$$L_j + T_j \leqslant U_j \tag{3}$$

that is, if after pickup from the origin of the tasks, there is enough time for the vehicle to travel to the destination, load and unload the commodities and meet the latest delivery time.

Similarly, we define the weight and volume associated with trip $j \in J^r$ as follows:

$$W_j = \sum_{s \in S_j^r} w_s = \text{the weight of goods delivered in trip } j,$$

$$V_j = \sum_{s \in S_j^r} v_s = \text{the volume of goods delivered in trip } j.$$

The first objective function term of $ATT_r$, associated with feasible weight and volume capacity utilization of the vehicles, is denoted by $F_r^{util}$ and defined by:

$$F_r^{util} = \sum_{j \in J^r | S_j^r \neq \Phi} \left( \frac{Q^W - W_j}{Q^W} + \frac{Q^V - V_j}{Q^V} \right). \tag{4}$$

The above term expresses the sum of percentage unutilized weight and volume capacities of all trips between origin–destination $r$ (all $j \in J^r$) that are not empty ($S_j^r \neq \Phi$). Lemma 1 shows that minimizing $F_r^{util}$ also minimizes the number of trips and hence the traveling costs. Note that when trips satisfy the capacity restrictions, each term in $F_r^{util}$ is non-negative.

**Lemma 1.** *Minimizing $F_r^{util}$ is equivalent to minimizing the number of vehicle trips.*

**Proof.**

$$F_r^{util} = \sum_{j \in J^r | S_j^r \neq \Phi} \left( \frac{Q^W - W_j}{Q^W} + \frac{Q^V - V_j}{Q^V} \right)$$
$$= \sum_{j \in J^r | S_j^r \neq \Phi} 2 - \frac{1}{Q^W} \sum_{j \in J^r | S_j^r \neq \Phi} \sum_{s \in S_j^r} w_s - \frac{1}{Q^V} \sum_{j \in J^r | S_j^r \neq \Phi} \sum_{s \in S_j^r} v_s.$$

In the expression above the last two terms are constants since the value of the summation terms are equal to the sum of all tasks' weights and volumes, respectively, between origin–destination $r$. Therefore, minimizing $F_r^{util}$ is equivalent to minimizing the first term which is twice the number of trips. □

Towards the derivation of our second objective function term, we define for $s \in S^r$:

$\delta_s$ = the slack of task $s$, where $\delta_s = u_s - l_s - t_r - \tau$

and similarly for $j \in J^r$:

$\Delta_j$ = the slack of trip $j$, where $\Delta_j = U_j - L_j - t_r - \tau$.

Our second objective function term, $F_r^{flex}$, is defined as follows:

$$F_r^{flex} = \sum_{j \in J^r} \sum_{s \in S_j^r | \delta_s > 0} \frac{\delta_s - \Delta_j}{\delta_s}. \tag{5}$$

$F_r^{flex}$ is the sum of the reduction (in percentages) in the slack of all tasks that belong to origin–destination pair $r \in R$, as a consequence of assigning the tasks together into trips as specified by the solution $J^r$ (and considering the slack of a trip as the new slack of all tasks that belong to it). Note that by feasibility, when $s \in S_j^r$ it is true that $\delta_s \geqslant \Delta_j$ and hence each term in $F_r^{flex}$ is non-negative. Minimizing $F_r^{flex}$ motivates assigning together tasks with similar time window parameters, which may allow forming efficient trips in the STV problem. Note, however, that minimizing this term by itself is not desirable since its minimum occurs when each trip consists of one task only, or when each trip consists of tasks that all have the exact same time window parameters. Such solutions are likely to incur high traveling costs and are not desirable as far as the $F_r^{util}$ term is concerned, hence the use of both objective function terms balances all cost components.

Based on the above derivations we now present a mathematical formulation of $ATT_r$ which clarifies the exact problem and is used later in the description of the TS method. In the formulation we as-

sign weights $\alpha$ and $1-\alpha$ to $F_r^{util}$ and $F_r^{flex}$, respectively, for varying values of $\alpha$ between zero and one. Ideally, the entire efficient frontier of the multi-objective function should be constructed. However, a separate efficient frontier exists for each origin–destination pair and it is unclear how (or very expensive computationally) to combine them when forming the input to the STV problem.

We define binary decision variables $x_{sj} = 1$ if task $s \in S^r$ is assigned to trip $j \in J^r$, and 0 otherwise. Although the number of trips is not known in advance, an upper bound on it (e.g., the number of tasks) may be used. Note also that the number of terms in (4) equals twice the number of trips, and the number of terms in (5) equals the number of tasks, which is usually higher. Moreover, the actual scale of these terms may not be the same, even though they are all between zero and one. Thus, the weight coefficient $\alpha$ in the multi objective cost function also serves as a normalizing factor. We formulate the problem as follows:

$$\min \alpha F_r^{util} + (1-\alpha) F_r^{flex} \tag{6}$$

$$\text{s.t.} \quad \sum_{j \in J^r} x_{sj} = 1 \quad \forall s \in S^r, \tag{7}$$

$$\sum_{s \in S^r} w_s x_{sj} \leqslant Q^W \quad \forall j \in J^r, \tag{8}$$

$$\sum_{s \in S^r} v_s x_{sj} \leqslant Q^V \quad \forall j \in J^r, \tag{9}$$

$$\max_{s \in S^r} \{l_s x_{sj}\} + t_r + \tau \leqslant \min_{s \in S^r} \{u_s x_{sj} + T(1 - x_{sj})\} \quad \forall j \in J^r, \tag{10}$$

$$x_{sj} \in \{0,1\} \quad \forall j \in J^r \quad \forall s \in S^r. \tag{11}$$

The objective function (6) aims at minimizing the criteria discussed above. Constraints (7) require each task to be assigned to exactly one vehicle trip. Constraints (8) and (9) require that the vehicle's weight and volume capacities, respectively, are not exceeded. Constraints (10) ensure that the time window of each trip is feasible. Finally, constraints (11) impose binary restrictions. While the constraints in (10) may be linearized, the objective function (6) cannot be represented linearly using the $x_{sj}$ decision variables, therefore this formulation is not used for solving the problem. However, a relaxed version of it, with an additional term in the objective function, is used in the TS (Tabu Search) algorithm, described next.

For given $r$ and $\alpha$ values, we obtain an approximate solution to the above problem through a TS method. The results of $ATT_r$ for all $r \in R$ (each with a certain $\alpha$ value, as explained below) are considered in the STV problem. In addition to providing a good input to the STV problem, this process is also useful in analyzing the importance of each of the two objective function terms considered, as discussed in Section 6. Several TS algorithms have been proposed in the literature for assignment problems, including algorithms for the *Generalized Assignment Problem* (GAP), see, for example, Osman (1995). The GAP consists of assigning each of a given number of tasks to agents, where each task requires a certain amount of a resource from the agent, and each agent has a limited amount of the resource. A TS algorithm presented by Glover (1977) for the GAP was used by several authors, including Laguna et al. (1995)and Díaz and Fernandez (2001). Problem $ATT_r$, which includes time window constraints, can be viewed as an extension of the GAP, where trips in the $ATT_r$ correspond to agents in the GAP. Therefore, we modify existing TS algorithms proposed for the GAP. The detailed description of the adapted implementation is provided in Appendix A.

Finally, recall that the above TS algorithm for $ATT_r$ assumes a certain $\alpha$ value in the bi-criterion objective function. Initially the same $\alpha$ value is used for every origin–destination pair $r$ and the results of the associated $ATT_r$ problems are used as an input to the STV problem. The process repeats itself for every $\alpha$ value, so that the STV problem is solved once for every $\alpha$ value. To allow for STV instances which use inputs from $ATT_r$ problems with a possibly different $\alpha$ value for each, we also consider an optional phase. In the optional phase we define a set of $\alpha$ values whose associated $ATT_r$ solutions (when $\alpha$ was identical) produced good results for the STV problem. Then, the STV problem is solved for each alpha combination from the chosen set. Fig. 3 illustrates this relationship between the ATT and STV problems. An evaluation of the proposed TS algorithm is presented in Section 6, together with an evaluation of the entire solution method to the ASTV problem.

### 4.3. Solving the STV problem

The trips created in the ATT problem by the tasks assignments are treated as given full-load trips in the STV problem. Thus, the STV problem is concerned with combining and scheduling full-load trips into vehicle routes, given the number of available vehicles in each DC. The problem is similar to the MDVSPTW (multi depot vehicle scheduling problem with time windows) discussed in the Introduction, however the objective in the STV problem is minimizing total cost and not traveling distance. Thus, existing methods from the literature cannot be used to solve it. Accordingly, our solution method consists of two phases:

Phase 1: For each DC $i \in D$ we define problem $STV_i$ which considers trips destined to DC $i$. $STV_i$ is the problem of combining and scheduling round-trips, which originate and terminate at DC $i$, assuming that enough vehicles are available there. In this phase all $STV_i$ problems are solved separately.

Phase 2: In this phase we consider modifying or combining routes constructed in phase 1. This may result in routes which originate from DCs that are possibly different than the destinations of some of the trips included in those routes. This may occur: (a) if the solution of $STV_i$ for some $i$ is using more vehicles than available there, in which case some routes may be modified so that some vehicles change their originating DCs; (b) if it is less costly to combine routes that originate from different $STV_i$ problems, than to keep them separate, in which case the combined route is performed by a vehicle from one of the involved DCs.

We now turn to solving phase 1 for a given $i \in D$. We note that an IP formulation may be adapted from Chuzhoy and Naor (2004) for the special case in which the traveling and waiting costs are zero, i.e., only fixed vehicle costs exist. This IP formulation to the $STV_i$ is presented in Appendix B. This formulation is quite efficient, and according to our experience could be solved using the Cplex solver in most practical cases. However, since it includes only fixed vehicle costs, it serves as an approximation to the original $STV_i$ problem, which may produce good results only when the fixed vehicle cost component is dominant. Otherwise, an alternative method is desirable. Therefore, we develop a polynomial time heuristic, denoted as algorithm *Combine*, which aims at minimizing the number of routes as a primary goal, and minimizing the waiting periods as a secondary goal. Our proposed heuristic is a greedy single-pass algorithm, however, with a sophisticated and designated pre-processing step and scheduling rules.

We use the following notation, some of which was defined in Section 4.2, and become parameters in this section:

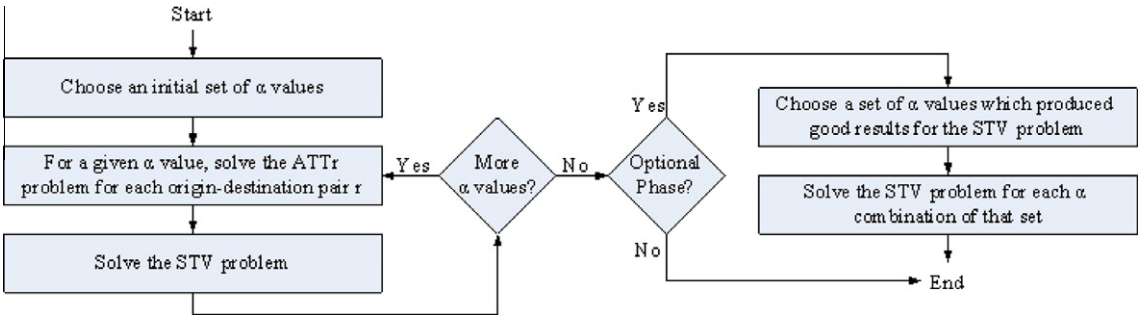


**Fig. 3.** The relationship between the ATT and STV problems.

$\widetilde{J}^i$ the set of trips sharing the same destination $i \in D$

$N^i$ the number of trips sharing destination $i \in D$, $(N^i = |\widetilde{J}^i|)$

$L_j$ earliest pickup time at the origin of tasks assigned to trip $j \in \widetilde{J}^i$

$U_j$ latest delivery time at the destination of tasks assigned to trip $j \in \widetilde{J}^i$

$T_j$ execution time of trip $j \in \widetilde{J}^i$, $(T_j = t_r + \tau$ for $j \in \widetilde{J}^i \cap J^r)$

Since in phase 1 all trips are round trips, we use the following adjusted parameters.

$\widehat{L}_j = L_j - t_r$ = earliest time the vehicle can leave the destination (DC $i$) before traveling to the origin of the trip (a certain factory) to pick up tasks assigned to trip $j \in \widetilde{J}^i \cap J^r$.

$\widehat{T}_j = T_j + t_r$ = *extended* execution time of trip $j \in \widetilde{J}^i \cap J^r$, where the extension includes the traveling time of the first part of the round trip, that is, from the destination to the origin.

Recall that $\Delta_j = U_j - L_j - t_r - \tau = U_j - \widehat{L}_j - \widehat{T}_j$ denotes the slack of trip $j$, which is also the length of time trip $j$ can move within its time window. A legitimate *placement* $\phi_j \in \{0, \ldots, \Delta_j\}$ needs to be selected for each trip $j \in \widetilde{J}^i$, corresponding to time slots $[\widehat{L}_j + \phi_j, \ldots, \widehat{L}_j + \phi_j + \widehat{T}_j)$ that are assigned to that trip (we use the convention that time slot $t$ is associated with the interval $[t, t+1]$).

In each iteration of phase 1 of algorithm Combine, one trip is scheduled in an existing or in a new route. We first describe the pre-processing step in which the order of trips to be scheduled is determined. Then we describe the criteria by which the placement (in time) of a trip is determined.

In the pre-processing step we define for each trip $j \in \widetilde{J}^i$ an index, referred to as the *extended flexibility index* $\widehat{I}_j$ which is computed by $(U_j - \widehat{L}_j - \widehat{T}_j)/\widehat{T}_j$. The trips are considered and scheduled in ascending order of their extended flexibility indices, that is: $\widehat{I}_1 \leqslant \widehat{I}_2 \leqslant \ldots \leqslant \widehat{I}_{N^i}$. Note that for trips with an equal slack, the above order reduces to the well known Longest Processing Time (LPT) heuristic in the scheduling literature. In case of equality in the flexibility index, the trip with the smaller execution time is considered first. Using the above order of trips, we specify next how the heuristic chooses the placement, i.e., the time intervals, in which the trips are scheduled.

In a given iteration, let $\Psi^i_t$ denote the number of trips from the yet unscheduled trips of $\widetilde{J}^i$ that can be performed during time slot $t$. $\Psi^i_t$ represents the potential future use of time slot $t$ and may be interpreted as the potential demand for time slot $t$. Let $a_{jt} = 1$ if trip $j \in \widetilde{J}^i$ can be executed during time slot $t$, i.e., if $\widehat{L}_j \leqslant t \leqslant U_j$, and 0 otherwise. $\Psi^i_t$ is initially calculated for each $t$ by (12) and then updated after each placement decision for trip $j \in \widetilde{J}^i$ by (13).

$$\Psi^i_t = \sum_{j \in \widetilde{J}^i} a_{jt} \quad \widehat{L}_j \leqslant t < U_j, \tag{12}$$

$$\Psi^i_t = \begin{cases} \Psi^i_t - 1 & \widehat{L}_j \leqslant t \leqslant U_j, \\ \Psi^i_t & \text{otherwise.} \end{cases} \tag{13}$$

Denote by $K^i$ the already constructed set of routes performed by vehicles housed in DC $i$. As explained above, the algorithm selects the lowest extended flexibility indexed trip among the remaining trips to be scheduled and attempts to schedule it, if feasible, in an existing route $k \in K^i$. Otherwise, if this is not feasible, it creates a new route for the trip.

Denote by $\Omega^i_t$ the number of routes from set $K^i$ in which time slot $t$ is unassigned. $\Omega^i_t$ can be interpreted as the supply of time slot $t$ in $K^i$ and is initialized at zero for each time slot $t$. Then, $\Omega^i_t$ is updated by (14) when trip $j$ is scheduled in a new route in placement $\phi_j$, and by (15) when it is scheduled in an existing route in placement $\phi_j$.

$$\Omega^i_t = \begin{cases} \Omega^i_t + 1 & t < \widehat{L}_j + \phi_j, \quad t \geqslant L_j + \phi_j + \widehat{T}_j, \\ \Omega^i_t & \text{otherwise,} \end{cases} \tag{14}$$

$$\Omega^i_t = \begin{cases} \Omega^i_t - 1 & \widehat{L}_j + \phi_j \leqslant t < L_j + \phi_j + \widehat{T}_j, \\ \Omega^i_t & \text{otherwise.} \end{cases} \tag{15}$$

The idea of algorithm Combine is to attempt to schedule trips in time slots with low demand and/or high supply. Thus, the selection of placement $\phi_j$ in which trip $j \in \widetilde{J}^i$ is performed in an existing or in a new route, depends on the supply and demand of time slots $[\widehat{L}_j + \phi_j, \ldots, \widehat{L}_j + \phi_j + \widehat{T}_j)$, $\phi_j \in \{0, \ldots, \Delta_j\}$. A preference function of the demand and supply of those time slots can be represented by:

$$h(\phi_j) = \beta[f(\Psi^i_{\widehat{L}_j+\phi_j}, \ldots, \Psi^i_{\widehat{L}_j+\phi_j+\widehat{T}_j-1})] - (1-\beta)[f(\Omega^i_{\widehat{L}_j+\phi_j}, \ldots, \Omega^i_{\widehat{L}_j+\phi_j+\widehat{T}_j-1})], \tag{16}$$

where $f(\cdot)$ is a function of the relevant demands and supplies, assumed to be increasing in its elements, and $\beta$ and $(1-\beta)$ denote the relative weights given to the demand and supply functions, respectively. Function $h(\phi_j)$ in (16) represents a measure of the attractiveness of placement $\phi_j$, where a low $h(\phi_j)$ value is preferred. Various $f(\cdot)$ functions may be used, including different functions for supply and demand. We consider the following:

$$h_1(\phi_j) = \beta\left(\sum_{t=0}^{\widehat{T}_j-1} \Psi^i_{\widehat{L}_j+\phi_j+t}\right) - (1-\beta)\left(\sum_{t=0}^{\widehat{T}_j-1} \Omega^i_{\widehat{L}_j+\phi_j+t}\right), \tag{17}$$

and

$$h_2(\phi_j) = \beta[\max(\Psi^i_{\widehat{L}_j+\phi_j}, \ldots, \Psi^i_{\widehat{L}_j+\phi_j+\widehat{T}_j-1})] - (1-\beta)[\max(\Omega^i_{\widehat{L}_j+\phi_j}, \ldots, \Omega^i_{\widehat{L}_j+\phi_j+\widehat{T}_j-1})]. \tag{18}$$

Algorithm Combine solves the $STV_i$ problem for several values of $\beta$ between 0 and 1 for each of the objective functions (17) and (18). The cost of each solution is calculated in terms of the original objective function (sum of fixed, traveling and waiting costs of all routes), and the lowest cost solution for each DC is considered in the integration phase. Additional scheduling criterions were used in the implementation of the algorithm to tie break alternative scheduling options and an improvement stage was performed to further minimize the possible vehicle idle time. The result of this phase is a set of routes for each DC $i \in D$ separately. Recall that by solving the IP formulation (referred to as the IP method) we obtain an additional solution. At the end of phase 1 the solutions resulting from the two methods (algorithm Combine and the IP method) are compared, and the better result in terms of the original objective function is chosen.

The algorithm continues to phase 2 of the STV problem, in which we develop a saving based algorithm, inspired by the saving algorithm of Clarke and Wright (1964) for the vehicle routing problem. Note that this is the first time in the entire solution method for the ASTV problem where routing elements, i.e., the $t_{ij}$ parameters, are considered. Route integration may save a vehicle fixed cost at the expense of possible higher traveling and waiting costs. This phase consists of considering possible integration of routes from different DCs, by computing their associated cost savings, and selecting the integration with the highest cost saving, if it is positive. Specifically, we consider a possible integration of routes $k \in K^i$ and $l \in K^j$ when $i, j \in D$, $i \neq j$, and possible execution of the combined route by a vehicle from DC $i$ or from DC $j$. Note that routes from the same DC may not be combined since phase 1 constructed routes from the same DC in a way that they cannot be further integrated.

Similarly, the integration phase addresses situations where the number of vehicles housed in a certain DC is smaller than the number of routes constructed for that DC i.e., when $V^i < |K^i|$ for a certain DC $i \in D$ (recall that $V^i$ denotes the number of vehicles located in DC $i$). This case is handled by assigning an infinite operational cost to the first $|K^i| - V^i$ routes which results in a positive saving for any feasible combined route. When a certain DC has extra vehicles, this is represented by empty routes with zero operational costs which are assigned to the unused vehicles in that DC. We omit the details of the saving calculations, since they are relatively standard. At the end of the integration phase, the resulting tours are the final solution of the ASTV problem.

## 5. Lower bounds

In this section we present a lower bound on the value of the optimal solution to the ASTV problem which enables us to evaluate the effectiveness of our heuristic solution method. We develop a lower bound for each of the ASTV cost components, i.e., on the traveling costs, the waiting costs, and the fixed cost. These lower bounds are achieved by considering relaxed problems of the original problem, which are formulated and solved as integer programs. These formulations employ binary decision variables $x_{st}$ for each task $s \in S$ and each time slot $t \in \{0, \ldots, T-1\}$, where $x_{st} = 1$ if task $s$ is assigned to a trip that starts at time slot $t$, and 0 otherwise. In addition, decision variables $h_{rt}$ denote the number of trips delivering tasks between origin–destination pair $r$, that start at time slot $t$.

### 5.1. Traveling costs

As mentioned in Section 4, when assuming that enough vehicles are available at each DC, the total traveling cost is minimized when all trips are round-trips and their total number is minimized. Furthermore, the problem of finding the minimum number of trips is then separable by origin–destination pairs $r \in R$. Thus, for each $r \in R$ a lower bound on the traveling cost can be calculated by multiplying the traveling cost per time slot, $\lambda^a$, by a lower bound on the number of round-trips between this pair, denoted $z_r$, and by the traveling time of a round trip between the DC and factory. Therefore, the lower bound on the traveling cost for an ASTV instance equals $\lambda^a \sum_{r \in R} (z_r \cdot 2t_r)$.

To find $z_r$ we formulate an integer program, $LB_r^1$. Note that formulation $LB_r^1$ considers a single origin–destination pair, therefore the decision variables $h_{rt}$ can be represented with only a single index $t$. We nevertheless use the above variable representation for consistency.

$$\left(LB_r^1\right) \quad z_r = \min \left\{ \sum_{t=0}^{T-(2t_r+\tau)} h_{rt} \right\}, \tag{19}$$

$$\text{s.t.} \quad \sum_{t=l_s-t_r}^{u_s-(2t_r+\tau)} x_{st} = 1 \quad \forall s \in S^r, \tag{20}$$

$$h_{rt} \geqslant \frac{\sum_{s \in S^r} w_s x_{st}}{Q^W} \quad t = 0, \ldots, T-(2t_r+\tau), \tag{21}$$

$$h_{rt} \geqslant \frac{\sum_{s \in S^r} v_s x_{st}}{Q^V} \quad t = 0, \ldots, T-(2t_r+\tau), \tag{22}$$

$$x_{st} \in \{0,1\} \ \forall s \in S^r, \ t = l_s - t_r, \ldots, u_s - (2t_r+\tau), \tag{23}$$

$$h_{rt} \geqslant 0 \quad \text{integer } t = 0, \ldots, T-(2t_r+\tau). \tag{24}$$

Objective function (19) minimizes the total number of round-trips starting at all time slots. Constraints (20) ensure that each task $s \in S^r$ is assigned to exactly one round-trip that starts within task $s$'s time window. Constraints (21) (and (22)) state that the number of round-trips that start on time slot $t$ must be greater than or equal to the sum of weights (volumes) of all tasks assigned to trips that start on time slot $t$ divided by the vehicle weight (volume) capacity. Note that $T-(2t_r+\tau)$ is the latest possible round trip starting time for trips that deliver tasks between origin–destination pair $r$. Finally, constraints (23) and (24) impose binary and integrality restrictions on the $x_{st}$ and $h_{rt}$ variables, respectively.

Using the $x_{st}$ variables in $LB_r^1$ enables us to deal with the "difficult" time window constraints encountered in the $ATT_r$ problem by defining the $x_{st}$ variables only for time slots $t = l_s - t_r, \ldots, u_s - (2t_r+\tau)$, so that the time window constraints are "automatically" satisfied. This approach is possible in the lower bound context since the $x_{st}$ variables indicate the starting time $t$ of the trip to which task $s$ is assigned, but they do not indicate the specific trip to which it is assigned. Thus, the difficult "bin-packing" element is not resolved in $LB_r^1$.

In most cases, an optimal solution to $LB_r^1$ can be reached in a reasonable amount of time for instances of practical size. As the number of tasks between each origin–destination pair gets larger, and as the flexibility of the tasks gets larger (both effects increase the number of $x_{st}$ variables), it becomes harder to solve the problem optimally, in which case we remove the integrality requirement in (24), resulting in a weaker lower bound.

### 5.2. Waiting costs

The lower bound on the waiting costs uses the lower bound on the minimum number of round-trips between each origin–destination pair, found by formulation $LB_r^1$. There are two vehicle waiting situations in the ASTV problem: waiting while loading and unloading commodity (denoted by $z^{w1}$) and idle time between trips (denoted by $z^{w2}$). The loading plus unloading times are identical for all trips, and equal $\tau$ per trip. Thus, $z^{w1}$ is minimized when the total number of trips is minimal. Using the lower bound on the number of trips from the previous section, $z_r$, we evaluate $z^{w1}$ by

$\lambda^w \cdot \tau \cdot \sum_{r\in R} z_r$. Idle time between different trips in the same route results from scheduling trips to routes. These waiting times can be avoided by ordering a separate vehicle for each trip. Thus, $z^{w2} = 0$, and the overall lower bound on the waiting cost is $\lambda^w \cdot \tau \cdot \sum_{r\in R} z_r$.

### 5.3. Fixed costs

The total vehicle fixed cost is minimized when the number of vehicle routes is minimal. Therefore, it may be evaluated by multiplying the vehicle fixed cost, $f$, by a lower bound on the minimum number of routes in the ASTV problem. We first calculate this lower bound for a special case of the ASTV problem in which there exists a single DC, and denote this problem by ASTV1. Note that in ASTV1, tasks destined to the DC are delivered by vehicles housed in that DC, and thus all trips are round-trips (although each round trip may involve a different factory). Let $Z$ be a lower bound on the minimum number of routes in ASTV1, which may be found by solving the integer program $LB2$ presented below. While $LB2$ is an extension of $LB_r^1$, it is no longer separable by origin–destination pairs.

$$(LB2) \quad \min\{Z\} \tag{25}$$

$$\text{s.t.} \quad Constraints(20) - (24) \quad \forall r \in R,$$

$$Z \geqslant \sum_{r\in R} \sum_{t'=t-(2t_r+\tau)+1}^{t} h_{rt'} \quad t = 0,\ldots,T. \tag{26}$$

Objective function (25) minimizes the number of routes, while constraints (26) ensure that for each time slot the number of routes is at least as large as the sum of trips executed during that time slot, since trips that are executed during the same time slot cannot belong to the same route. Again, for larger instances, the integrality requirement in (24) is removed.

However, formulation $LB2$ cannot be used to obtain a lower bound on the number of routes for ASTV problem instances with several DCs, since then some routes may no longer involve only round-trips. In other words, trips of different origin–destination pairs may be combined together, which may result in a lower trip time than two separate round trips. Therefore, for instances with more than one DC we find a lower bound on the minimum number of routes by computing a lower bound on the total time required to perform all trips, and dividing it by the length of the planning period $T$. A lower bound on the total time required to perform all trips is obtained by summing two previously computed lower bounds: the lower bound on the number of trips and the lower bound on the waiting time. Thus, the minimum number of routes must be greater than or equal to $(\sum_{r\in R}(z_r \cdot 2t_r) + z^{w1})/T = \sum_{r\in R}(z_r \cdot (2t_r + \tau))/T$ and hence $f \cdot (\sum_{r\in R}(z_r \cdot 2t_r) + z^{w1})/T$ is the lower bound on the total fixed costs.

We expect this lower bound to be less tight than the lower bound computed by formulation $LB2$, but it is computationally easier, since $z^r$ is solved for each $r$ separately, and therefore, can be estimated for relatively large problems. The performance of the above lower bound is good for ASTV instances, in which most of the planning period is occupied.

## 6. Numerical study

In this section we present a numerical study, designed to assess the performance of the proposed heuristic solution method for the ASTV problem as well as to obtain insights on properties of efficient solutions. The heuristic was coded in C and executed on a Pentium 4 computer with 3.6 GHz and 4 GB RAM. The performance of the heuristic is evaluated by comparing the value of its solution to the lower bound discussed in Section 5. We report in Sections 6.1, 6.2, 6.3 on problems of various sizes, and present a summary of their performance in Section 6.4. An explanation on how the basic input parameters were generated is provided in Appendix C. Other parameter values are specified within this section, as needed.

### 6.1. Single DC and factory instances

First we consider instances with a single DC and a single factory in which round-trips are performed. We first present results obtained by the TS algorithm for the $ATT_r$ problem with $\alpha = 1$. We use $\alpha = 1$ since $\alpha$ denotes the weight given to the utilization term of the objective function, $F^{util}$, and minimizing $F^{util}$ is equivalent to minimizing the number of trips between the origin and the destination (Lemma 1). Thus, the lower bound on the minimum number of trips, presented in Section 5.1, is also a lower bound on the optimal value of $ATT_r$, which enables us to evaluate the performance of the TS algorithm. For other values of $\alpha$ it is unclear how to obtain a lower bound on the solution value of $ATT_r$.

The TS algorithm was tested on various problem sizes (number of tasks), and for each size 10 instances were generated. Fig. 4 illustrates the average gap from the lower bound for each problem size. The average gap over all instances was 1.96% with a variance of 0.02. Note that there is no clear trend in the results as the number of tasks increases. The algorithm ran between two to four minutes, depending on the problem size, thus providing good and fast solutions.

Subsequently, the entire solution method for the problem was tested on five problem sizes ranging from 50 to 250 tasks with 10 instances for each problem size. For each ASTV problem instance, the $ATT_r$ problem was solved for 11 different values of alpha between zero and one and six different values of the fixed cost parameter, $f$, ranging from 50 to 1,000. Thus, 3300 instances of $ATT_r$ were solved in this experiment. The STV was solved twice for each solution of $ATT_r$, once by the IP method and once by the heuristic method, and the better result was chosen. The traveling cost parameter, $\lambda^a$, and the waiting cost parameter, $\lambda^w$, were set to 15 and 10, respectively. We refer to $\lambda^a$ and $\lambda^w$ as the variable cost parameters and to $f/\lambda^w$ as the parameters cost ratio.

We observed that small values of alpha did not produce good results for the entire problem, since the solution had a large number of poorly utilized trips, which resulted in a high number of routes. This was the case across all problem sets considered and therefore these $\alpha$ values are not shown in subsequent figures. In fact, the results demonstrated that middle values of alpha generated the best heuristic solutions in most cases, implying that both the utilization and the flexibility terms are important when considering the initial assignment decisions. In particular, it is not always beneficial to utilize the capacity of the vehicles as much as possible.

The optimality gap of the heuristic, as a function of the cost ratio $f/\lambda^w$, is presented in Fig. 5. It is evident that the average gap decreases as the cost ratio decreases, i.e., when the variable costs become more dominant. We believe that this is due to the better

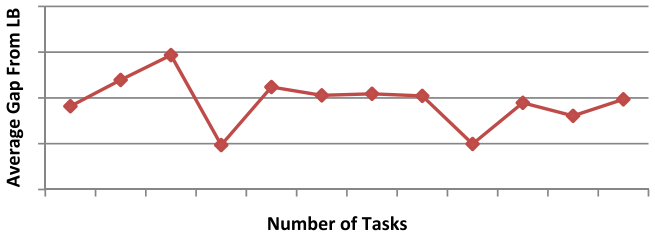


**Fig. 4.** TS algorithm evaluation with $\alpha = 1$.

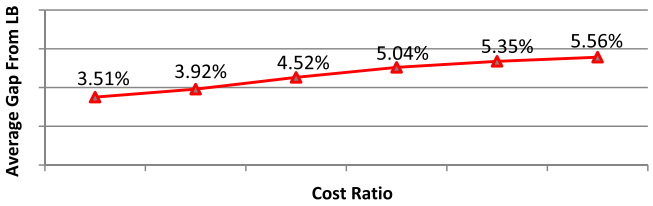


**Fig. 5.** Average gap from the lower bound as a function of the cost ratio for single DC and factory instances.

quality of the travel cost lower bound, compared to the fixed cost lower bound (as shown in subsequent figures) and the fact that the travel costs become a dominant component in the total cost when the cost ratio decreases. In all cases, the performance of the heuristic is very good, with gaps ranging from 3.51% to 5.56% for the lowest and highest cost ratio, respectively, and with an overall average gap over all instances of 4.65%.

We further consider the performance of a restricted version of the heuristic, when only a single value of $\alpha$ is used for all instances. Fig. 6 presents the optimality gaps of the travel and the fixed cost components from their respective lower bounds (the lower bound on the waiting cost was zero), as well as the distance of the total cost from its lower bound, as a function of alpha, for instances with cost ratio of $f/\lambda^w = 50$. This cost ratio represents cases where none of the fixed or variable cost components are dominant. We observe, as expected, that the gap of the traveling cost component from its lower bound increases as alpha decreases. The fixed cost graph is not monotone and is always higher than the travel cost graph in ranges of alpha where good solutions are obtained. The graph representing the gap of the total cost from its lower bound is rather flat for $\alpha$ values between 0.5 and 0.99. Note, however, that the graph represents average results and different instances may achieve their minimum at varying alpha values, and the graphs, in general, are more variable. However, the gap rapidly increases for $\alpha$ values lower than 0.5, as a result of poorly utilized trips.

Thus, for this set of experiments, even if a single alpha value is used for all instances, the total costs are higher than their respective lower bound by 7.5–8% on average, provided that the chosen alpha value is between 0.5 and 0.99. This is compared to 5.04% when choosing the best alpha value for each instance. Overall, these results demonstrate the efficiency of the general solution method (even for a restricted version), as well as the added value of solving for several alpha values and choosing the best.

For instances with dominating fixed or variable cost parameters the results for the restricted version of the heuristic are similar, with slight differences. For instances with dominating fixed (variable) costs the minimum average gap was 7.96% (4.23%) for an alpha value of 0.5 (0.99). Thus, when the variable costs are dominating and a single alpha value is used, considering the vehicle utilization factor alone provides the best results. When this is not the case, the utilization and the flexibility cost factors should be balanced by using an intermediate value of alpha. Finally, we performed sensitivity analysis on the tasks input parameters, i.e., on the weight, volume, and time window parameters, and observed that the solution method is quite robust to such changes.

### 6.2. Single DC – Multiple factories instances

In this section we examine ASTV problem instances including a single DC and three factories, i.e., three origin–destination pairs. The solution consists of round-trips from the DC, where the vehicles are housed, to one of the factories and back. The solution method was tested on three problem sizes ranging from a total of 150 tasks to a total of 450 tasks.

Overall, the average gap of the total cost from its lower bound is 7.42% where again the gap of the fixed cost from its lower bound is significantly larger than that of the travel cost. The gap is significantly affected by the cost ratio, with an average gap of 3.5% and 10.97% for cost ratios of 5 and 100, respectively. These gaps are larger than those in the previous section since in these larger instances the integrality requirement (24) in LB2 is removed (due to computational limitations), which results in a less tight lower bound on the fixed cost.

Here we were also able to test the optional solution method in which the STV problem is solved for each alpha combination from a set of good alpha values. We considered six values of alpha between 0.5 and 0.99 that produced the best results for these and previously solved instances. Thus, each of the three DC-factory ATT problems was solved six times, and the STV problem was solved $6^3 = 216$ times (once for each alpha combination of the three ATT solutions). The resulting gaps, as a function of the cost ratio, are presented in Fig. 7.

The results indicate that while the optional method clearly dominates the original method, the largest gap between them is only 0.75% or 0.77% (when $f/\lambda^w = 100$ or 75, respectively), and the gap narrows as the significance of the variable costs increases. This trend occurs since the optional solution method cannot improve the travel cost determined by the ATT, it can only improve the fixed cost and waiting cost. Therefore, the lower the importance of the variable cost component, the higher the improvement in percentage of the optional method. One should also note that the improvement of the optional method comes at the cost of a significantly larger computational effort.

### 6.3. Multiple DCs and factories instances

In this section we examine larger ASTV problem instances including three DCs and six factories. Using these practical instances we can thoroughly assess the entire proposed solution method which includes three main phases. The first is the assignment of tasks to trips for each origin–destination pair. The second is the scheduling of trips to routes for each DC separately, while the

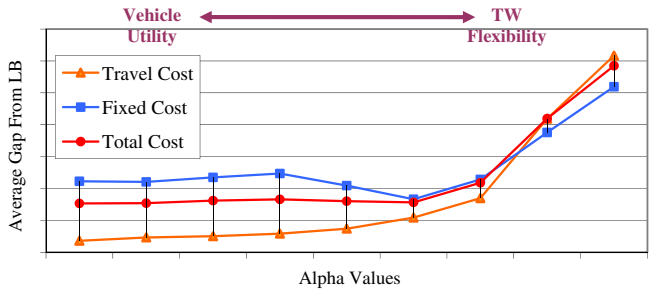


**Fig. 6.** Average gap from lower bound for each cost component as a function of alpha for single DC and factory instances with no dominating cost component.

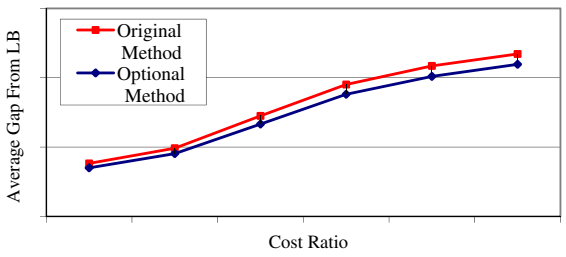


**Fig. 7.** Performance of the original and optional solution methods as a function of the cost ratio for single DC and three factories instances.

**Table 1**
Computational results summary.

| DCs | Fac | Tasks | AGDF (%) | AGND (%) | AGDV (%) | OAG (%) |
|---|---|---|---|---|---|---|
| 1 | 1 | 50 | 7.08 | 6.09 | 3.78 | 5.49 |
| | | 100 | 5.95 | 5.72 | 4.19 | 5.32 |
| | | 150 | 4.77 | 4.52 | 3.78 | 4.34 |
| | | 200 | 5.63 | 4.91 | 3.06 | 4.45 |
| | | 250 | 4.39 | 3.98 | 2.72 | 3.65 |
| 1 | 3 | 150 | 10.07 | 8.04 | 3.20 | 6.79 |
| | | 300 | 11.48 | 9.19 | 3.39 | 7.66 |
| | | 450 | 11.35 | 9.19 | 3.92 | 7.82 |
| 3 | 6 | 1800 | 6.67 | 5.55 | 3.24 | 4.97 |

third is the integration phase in which routes from different DCs are combined into routes performed by a vehicle from one of these DCs.

We consider a single scenario with a total of 1,800 tasks for the 18 origin–destination pairs. The $ATT_r$ problem was solved, for each origin–destination pair $r \in R$, for six different values of alpha between zero and one (0.99, 0.9, 0.8, 0.7, 0.6, and 0.5), which produced the best results for the previously solved instances. The average gap for these instances was 5% and we observed that even the gap of the fixed cost component was minimized in most cases for an alpha value of 0.99. The above small gap is due to the large number of tasks (600 tasks per DC). The set of utilized trips, obtained by solving the ATT with an alpha value of 0.99, is large enough and diverse enough (with respect to the time window parameters) for creating a set of laden vehicle routes. This is as opposed to a small set of non-flexible trips created with an alpha value of 0.99 in smaller ASTV problem instances, which are harder to combine into laden routes and therefore result in higher fixed costs. When the construction of laden vehicle routes is possible, minimizing the number of trips also minimizes the number of routes.

Thus, for instances in which the number of tasks destined to each DC is large, it is more important to create highly utilized vehicle trips than to create highly time window flexible vehicle trips. As explained in Section 5, the average gap from the lower bound on the fixed cost for multiple DCs instances is evaluated by summing the minimum traveling and waiting times, and dividing them by the length of the planning period. Therefore, the relatively small average gap on the fixed cost component can also be explained by the ability of the IP method to create laden vehicle routes which occupy most of the length of the planning period.

### 6.4. Computational results summary

In this section we present a summary of the computational results for the problems discussed in the previous sections. Table 1 reports the following for each scenario: number of DCs (*DCs*); number of factories (*Fac*); number of tasks (*Tasks*); average gap of the total cost from the lower bound, achieved by the solution method for instances with dominating fixed cost factor (*AGDF*) (cost ratio = 100), no-dominating cost factors (*AGND*) (cost ratio = 50), and dominating variable costs factors (*AGDV*) (cost ratio = 5). Finally, the last column reports on the overall average gap (*OAG*) of instances of all six values of cost ratio considered. For each instance the best result achieved either by the IP or by algorithm Combine is considered. In all problem categories except the single DC and single factory, the results obtained from the optional solution method are reported.

We conclude from these results that as the variable costs become dominant, the average gap of the total cost from the lower bound decreases. This is due to the performance of the TS algorithm for the ATT, which achieves a relatively low average gap between the traveling cost and its lower bound.

Running times greatly depend on the number of ATT problems solved in the first phase. On average, each ATT problem is run for 2.5–3 minutes, and the total number of ATT executions is the number of origin–destination pairs, multiplied by the number of alpha values (or combinations) used. The execution of the second phase (STV and the integration phase) require no longer than a few seconds, so they do not affect the run time significantly. Thus, the running time for our problems ranged from about half an hour for the small problems with one origin–destination pair (Section 6.1) and up to 4.8 hours for the large problems with 18 origin–destination pairs (Section 6.3).

## 7. Conclusions

We have identified and formulated a new transportation and scheduling problem in a distribution system, inspired by an application observed at a food manufacturer, with unique characteristics that were not identified and analyzed previously. We developed a heuristic solution method, based on a new decomposition/partitioning idea, which may be useful to other problems with a structure that enables such partitioning. Our heuristic includes detailed procedures for each of the resulting problems, and a detailed analysis of a lower bound to the entire problem. An extensive computational study indicates that our heuristic is very efficient.

## Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2011.06.013.

## References

Bent, R., Van Hentenryck, P., 2006. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. Computers and Operations Research 33 (4), 875–893.
Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. TOP 15, 1–31.
Bertossi, A.A., Carraresi, P., Gallo, G., 1987. On some matching problems arising in vehicle scheduling models. Networks 17 (3), 271–281.
Bodin, L., Golden, B., 1981. Classification in vehicle routing and scheduling. Networks 11, 97–108.
Brãndao, J., Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European Journal of Operational Research 100, 180–191.
Chuzhoy, J., Naor, S., 2004. New hardness results for congestion minimization and machine scheduling. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC), pp. 28–34.
Clarke, G., Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581.
Cordeau, J.-F., Laporte, G., Ropke, S., 2008. Recent models and algorithms for one-to-one pickup and delivery problems. In: Golden, B., Raghavan, R., Wasil, E. (Eds.), The vehicle routing problem: latest advances and new challenges. Springer, pp. 327–357.

Desaulniers, G., Lavigne, J., Soumis, F., 1998. vehicle scheduling problems with time windows and waiting costs. European Journal of Operational Research 111, 479–494.

Desrosiers, J., Laporte, G., Sauve, M., Soumis, F., Taillefer, S., 1988. Vehicle routing with full loads. Computers and Operations Research 15 (3), 219–226.

Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., 1995. Time constrained routing and scheduling. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Handbook in Operation Research and Management Science 8: Network Routing. North-Holland, Amsterdam, The Netherlands, pp. 35–139.

Díaz, J.A., Fernandez, E., 2001. A tabu search heuristic for the generalized assignment problem. European Journal of Operational Research 132 (1), 22–38.

Dondo, R., Cerdá, J., 2007. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. European Journal of Operational Research 176, 1478–1507.

Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. European Journal of Operational Research 54 (1), 7–22.

Federgruen, A., Tzur, M., 1999. Time-partitioning heuristics: application to one warehouse, multi-item, multi-retailer lot-sizing problems. Naval Research Logistics 46, 463–486.

Garey, M., Johnson, D., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. CA.W.H. Freeman and Company, San Francisco.

Glover, F., 1977. Heuristics for integer programming using surrogate constraints. Decision Sciences 8 (1), 156–166.

Jarrah, A.I., Johnson, E., Neubert, L.C., 2009. Large-scale, less-than-truckload service network design. Operations Research 57 (3), 609–625.

Karp, R.M., 1977. Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. Mathematics of Operations Research 2 (3), 209–224.

Laguna, M., Keely, J.P., González-Velarde, J.L., Glover, F., 1995. Tabu search for the multilevel generalized assignment problem. European Journal of Operational Research 82 (1), 176–189.

Lu, Q., Dessouky, M., 2004. An exact algorithm for the multiple vehicle pickup and delivery problem. Transportation Science 38 (4), 503–514.

Osman, I.H., 1995. Heuristics for the generalized assignment problem, simulated annealing and tabu search approaches. OR Spektrum 17 (4), 211–225.

Pankratz, G., 2005. A grouping genetic algorithm for the pickup and delivery problem with time windows. OR Spectrum 27 (1), 21–41.

Potvin, J.-Y., Rousseau, J.-M., 1993. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research 66, 331–340.

Ropke, S., Cordeau, J.-F., Laporte, G., 2007. Models and a branch-and-cut algorithm for pickup and delivery problems with time windows. Networks 49 (4), 258–272.

Sandhu, R., Klabjan, D., 2007. Integrated airline fleeting and crew-pairing decisions. Operations Research 55 (3), 439–456.

Savelsbergh, M.W.P., Sol, M., 1995. The general pickup and delivery problem. Transportation Science 29 (1), 17–29.

Toth, P., Vigo, D., (Eds.), 2002. The Vehicle routing problem. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA 19104-2688, USA.

Tung, D.V., Pinnoi, A., 2000. Vehicle routing-scheduling for waste collection in Hanoi. European Journal of Operational Research 125, 449–468.