An Inventory-Routing Problem with the Objective of Travel Time Minimization

Kunpeng Li[a,*], Bin Chen[a], Appa Iyer Sivakumar[b], Yong Wu[c]

[a] School of Management, Huazhong University of Science & Technology, Wuhan, Hubei, P.R. China. 430074.

[b] School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798.

[c] Department of International Business & Asian Studies, Griffith University, Gold Coast Campus, QLD 4222, Australia

* Corresponding author.

Tel.:+86-135-5443-6029.

Fax: +86-27-87556437.

E-mail: likp@mail.hust.edu.cn.

Postal address: School of Management, Huazhong University of Science & Technology, 1037 Luoyu Road, Wuhan, Hubei, P. R. China. 430074.

- We consider an inventory and routing problem to minimize the maximum route travel time;
- We present a tabu search algorithm with efficient improvement method in each iteration;
- Lagrangian relaxation is applied to obtain the lower bounds of the problem;
- Numerical results demonstrate the tabu search provide solutions close to the lower bounds.

An Inventory-Routing Problem with the Objective of Travel Time Minimization

Abstract: In this paper, we consider an inventory-routing problem (IRP) in a large petroleum and petrochemical enterprise group. Compared to many other IRPs, the problem in this paper includes some special aspects due to some operational constraints, such as hours-of-service regulations of the company and the industry. Also, in some cases, it is more important to avoid stock out for any station, rather than purely focusing on transportation cost minimization. The objective is to minimize the maximum of the route travel time, which is not addressed in the literature so far. We present a tabu search algorithm to tackle the problem, which builds in an efficient and effective procedure to improve the search quality in each iteration. Moreover, lower bounds of reasonable sized problems, which are intractable in the formulated mathematical model by existing optimization software, are obtained via Lagrangian relaxation technique. Computational results indicate that the lower bounds are tight and the tabu search is capable of providing near optimal, close-to-lower-bound solutions in a computational time effective manner.

Keywords: Logistics; Distribution; Inventory Routing; Tabu Search; Lagrangian Relaxation

1. Introduction

This paper studies a problem which is observed in a large petroleum and petrochemical enterprise group in China. The company relies on the product of gasoline for the majority of its sales and owns almost 40,000 gas stations spread all over China. Hundreds of oil depots are also maintained to replenish gasoline to the gas stations. Considering the number of depots and gas stations involved, the daily operations of gasoline distribution within this network is particularly complex. The problem can be formulated as an inventory-routing problem (IRP) with the following features:

1. The distribution of gasoline to gas stations is operated independently by each province. Therefore, the overall network is divided into sub-networks. For each province, there might be dozens of oil depots and thousands of gas stations. The network within each province is further divided into different districts according to the company’s current operational policy. Normally, each district has several oil depots and hundreds of gas stations. This is the basic operating unit for gasoline distribution within the company.
2. For each operating unit, the depots are regularly replenished to maintain a proper level of inventory to avoid stock outs. A central scheduling office determines the time and the amount to be delivered to each gas station, which has limited storage capacity with the daily demand represented by a consumption rate.
3. The product is distributed by a fleet of tank trucks, which belongs to a third party logistics (3PL) service provider. The 3PL previously belonged to the enterprise but now operates as an independent company to serve the function of transportation department of the enterprise.
4. When a truck arrives at a gas station, the gasoline is delivered at a delivery rate, i.e., the delivery cannot be instantaneous. An order up to level policy is applied to each gas station for inventory replenishment. Therefore, the maximum inventory level will be and should be reached once the gas station is visited by a truck.
5. In order to improve the utilization of transportation resources, a minimum delivery quantity (Campbell, 2004a) is specified for each station. This subsequently

specifies the earliest time a station should be visited. Moreover, the central scheduling office should guarantee that each gas station will not run out of stock, which specifies the latest time that the station should be visited given the consumption rate. These two constraints naturally pose a delivery time window for each gas station.

Each day, a fleet of vehicles depart from the oil depot and visit the gas stations following the routes designated by the central scheduling office. At the end of the day, the fleet returns to the depot or a specified parking area. For IRPs, one objective is to minimize the total transportation cost, which has received much attention in the literature (Andersson et al. 2010). However, the consideration of time is also very important in this problem.

In this paper, we address the IRPs from the perspective of travel time minimization. The objective is to minimize the maximum route travel time among all vehicles due to two reasons. Firstly, vehicles should deliver products to gas stations earlier than the latest allowable visiting time. If a gas station runs out of stock, it might not only be viewed as a business problem, but also a social problem by the local community. Therefore, it can be argued that in some cases, it is more important to avoid stocking outs, rather than purely focusing on transportation cost minimization. The similar priority rule is also applicable to the distribution of some scarce resources such as blood, and critical rescue goods stored for use in emergencies. The delivery of these resources has a strong focus on speed of delivery. Secondly, in the petrol delivery network, all vehicles should return to the depot within the work shift so that the total driving hours are maintained at a safe level and workers' preference on getting off duty on time are catered for. Furthermore, the maximum amount of accumulated driving time is limited by the regulations of the company. Similar regulations can be observed in the U.S., Europe and many other countries (e.g., Goel, 2010; Goel and Kok, 2012). Hence, the hours-of-service of each vehicle should be leveraged so that each driver has similar working hours, while the hours-of-service regulation is complied with at the same time. Although there is a large body of research addressing IRPs with objective of transportation cost minimization (Ng et al., 2008; Coelho et al., 2012a), there is almost no work considering time in the objective function.

The IRPs are difficult to solve (Campbell and Savelsbergh, 2004a). According to the published work, it is not easy to develop an exact algorithm for solving IRP with reasonable sizes (Archetti et al., 2007; Engineer et al., 2012). In practice, the tank of the gas station in urban area is relatively small due to limited space within the city and safety requirements. Thus, the delivery quantity for the gas station in the urban area is relatively small compared to vehicle capacity. As a first step, we concentrate on a simplified version of the problem, assuming that the vehicle capacity is un-constrained, which is similar to the uncapacitated assumption in Valle et al. (2011). However, we are well aware that in practice, more often than not, vehicles have limited capacity. We believe our contribution will serve as a basis for solving more realistic cases.

The remainder of this paper is organized as follows. We review the related literature in Section 2. The problem definition and the mathematical model are presented in Section 3 and a tabu search algorithm is proposed in Section 4. Section 5 employs the Lagrangian relaxation technique to obtain the lower bound for the problem. Section 6 presents computational results of the tabu search algorithm and the lower bounds, followed by the concluding Section 7.

2. Literature Review

The IRPs are observed in various industries and have received the attention of the academic communities since 1980, such as supermarket chains (Gaur and Fisher, 2004), industrial gas industry (Bell et al., 1983; Campbell and Savelsbergh, 2004a, 2004b), vending machine chain (Huang and Lin, 2010), automobile industries (Blumenfeld et al., 1987; Alegre et al., 2007; Ohlmann et al., 2008), meat industry (Oppen et al., 2010), oil refineries (Persson and Göthe-Lundgren, 2005), frozen food distribution companies (Custódio and Oliveira, 2006), blood distribution (Hemmelmayr et al., 2009) and maritime transportation industry (Al-Khayyal and Hwang, 2007; Dauzère-Pérès et al., 2007), etc. Recently, Andersson et al. (2010)

reviewed related literature published until 2008, where more than 100 papers or book chapters are classified accordingly to three types of planning horizon: instant time horizon, finite time horizon, and infinite planning horizon. IRPs are solved exactly by Archetti et al. (2007), Solyali and Süral (2011), Coelho and Laporte (2012), Adulyasak et al. (2012), etc. Also, efficient heuristic algorithms are proposed by Bertazzi et al. (2002), Zhao et al. (2007), Campbell and Savelsbergh (2004), Archetti et al. (2011), Coelho et al. (2012b, 2012c), Michel and Vanderbeck (2012), and Archetti et al. (2012), etc. In this paper, we only review the work published after 2008, which is not covered by the survey of Andersson et al. (2010).

Considering the infinite planning horizon, Bertazzi (2008) determines shipping policies that minimize the sum of transportation cost and inventory cost both at the supplier and at the retailers. The bounds of direct shipping over shipping with routing are developed subject to given conditions. Archibald (2009) also considers direct deliveries. In the work of Li et al. (2010), the effectiveness of the direct shipping strategy is evaluated. Raa and Aghezzaf (2008, 2009) assume that the demand rate is constant.

The IRPs with finite planning periods are addressed by Bard and Nananukul (2009a, 2009b), Kang and Kim (2010), Toriello et al. (2011) and Solyali and Süral (2011), etc., where both exact and heuristic algorithms are presented.

There is a set of work which considers different network structures in the IRPs as well. For example, Ohlmann et al. (2008) address the inbound vehicle routing problem, with the constraints that limit the amount of inventory in the logistical network and present a two phase solution procedure. Moin et al. (2011) address the multi-product IRP in a many-to-one distribution network with finite planning periods. Zhao et al. (2008) and Li et al. (2011a) address an integrated IRP in a three-echelon logistics system, which consists of a supplier, a central warehouse and a group of retailers.

Variations of IRPs can be found in the literature by considering different constraints or assumptions. Abdelmaguid et al. (2009) investigate the IRP with backlogging. Stochastic IRPs are considered by Chen and Lin (2009), Hvattum et al.

(2009) and Hvattum and Løkketangen (2009). Grønhaug et al. (2010) address a maritime IRP in the liquefied natural gas business. Huang and Lin (2010) study the multi-item IRPs with demand uncertainty. Benoist et al. (2011) present a randomized local search method for real-life IRP, which takes into account pickups, time windows, drivers' safety regulations, and orders. Li et al. (2011b) consider the IRPs with split deliveries while Liu and Lee (2011) address the IRP with time windows.

Recently, there emerged a set work which focuses on the petrol delivery problems. Various formulations and solution methodologies are proposed. Avella et al. (2004) study the case of a company that delivers different types of fuel to a set of fuel pumps. The objective is to minimize total travel cost. A branch-and-price algorithm is proposed. Cornillier et al. (2008) investigate a multi-period petrol station replenishment problem with the objective to maximize the total profit which equals to the revenue minus the sum of routing costs and of regular and overtime costs. Heuristic algorithm is proposed. Ng et al. (2008) present a case study on a single period tanker assignment and routing problem for petrol products in Hong Kong and develop a decision support system. Multiple objectives are considered simultaneously such that the number of tankers used and the number of drops in trips are minimized, profit in terms of total products delivered and utilization of resources are maximized. Muyldermans and Pang (2010) formulate the petrol delivery problem as the Multi-Compartment Vehicle Routing Problem with the objective of travel distance minimization and provide a local search procedure for this problem. Derigs et al. (2011) also formulate the petrol distribution problem as VRP with compartments. A heuristic algorithm is proposed, which combines local search and meta-heuristics. Cornillier et al. (2012) consider the multi-depot petrol station replenishment problem with time windows. Stations specify their demand by indicating the minimum and maximum quantities to be delivered for each ordered product and require the delivery within a predetermined time window. A heuristic algorithm is proposed to maximize the overall daily net revenue. Popovic et al. (2012) present a variable neighborhood search heuristic for solving a multi-product multi-period IRP in fuel delivery with multi-compartment homogeneous vehicles, and deterministic consumption that varies with each petrol station. The objective is to minimize the total cost of vehicle routing and inventory management.

According to the classification of the published work in Andersson et al. (2010), our problem focuses on the instant time horizon with deterministic demand. Routing of the fleet with homogeneous vehicles should be determined. Also, our problem is a variation of IRPs by considering a different objective function while most researchers take cost minimization as the objective. We would like to re-emphasize that in some real-life cases, while cost is still important, time has higher priority in the network of IRPs. Thus, our work attempts to take travel time minimization as the objective function.

3. Problem Formulation

We consider a logistic network in which the product of gasoline is distributed from the oil depot to a set of gas stations each day using a fleet of homogenous vehicles. For each gas station, an initial inventory level is given and the maximum inventory level is determined by its tank capacity. The problem is formulated based on the following assumptions:

- The oil depot is assumed to have infinite inventory.
- Each gas station has a product consumption rate.
- The order up to level policy is applied for each visit. That is, if a station is visited at time *t*, then the delivered product quantity is such that the inventory level will be brought to its maximum value (tank capacity).
- The vehicle’s delivery rate is the amount of product that can be delivered per hour. It is obvious that the delivery rate should be greater than the consumption rate for any gas station.
- A station should only be visited if it meets the minimum delivery quantity requirement. This improves the utilization of delivery vehicles and creates better insurance against running out of product (Campbell and Savelsbergh, 2004a). This constrains the earliest time that a station should be visited. Conversely, stock out is not allowed for each station, which specifies the

latest time that a station should be visited. Thus, each station has a specific delivery time window.

As mentioned in Section 1, higher priority is assigned to route travel time. Therefore, this IRP is investigated with the objective of minimizing the maximum of vehicle travel time within the network. The decision is to determine the route for each vehicle such that each gas station is visited once within its delivery time windows. For each route, it is interesting that the visiting sequence of gas stations not only affects the vehicle travel time of this route, but also determines the delivery time and delivery quantity for the visited gas stations.

Before presenting the mathematical model, the following set of indices and parameters are defined.

i	index of the gas station, which belongs to the gas station set $N = \{1, 2, ..., n\}$
k	index of vehicles, which belongs to the vehicle set $V = \{1, 2, ..., K\}$
o	the depot
I_i	the initial inventory level of gas station i
C_i	The tank capacity at gas station i
u_i	The product consumption rate in gas station i
R	the vehicle's delivery rate
$q_i^{\min}$	the minimum delivery quantity when station i is visited.
$[t_i^a, t_i^b]$	the delivery time windows of station i, where $t_i^a = \max(0, (q_i^{\min} - (C_i - I_i)) / u_i)$ and $t_i^b = I_i / u_i$.
TI_{ij}	the travel time between station i and station j.

The variables that need to be determined are as follows:

x_{ijk}	equals to 1 if vehicle k visits arc (i, j), 0 otherwise.
d_{ik}	the delivery quantity to station i by vehicle k.
t_{ik}	the delivery start time of vehicle k at station i

t_{ik}^{d} the departure time of vehicle *k* from station *i*.

t_{0k} the time the vehicle *k* reaches the depot at the end of the route.

t_{0k}^{d} the time the vehicle *k* departs from the depot at the beginning of the route.

The mathematical model is presented below:

$$\min(\max_{k} t_{0k}) \tag{1}$$

Subject to:

$$\sum_{k=1}^{K} \sum_{j=0, j\neq i}^{n} x_{ijk} = 1, \qquad i = 1,2,...,n \tag{2}$$

$$\sum_{j=1}^{n} x_{0jk} = 1, \qquad k = 1,2,...,K \tag{3}$$

$$\sum_{j=1}^{n} x_{j0k} = 1, \qquad k = 1,2,...,K \tag{4}$$

$$\sum_{i=0}^{n} x_{ihk} - \sum_{j=0}^{n} x_{hjk} = 0\ , \quad h = 1,2,....,n \qquad k = 1,2,...,K \tag{5}$$

$$t_{jk} \geq t_{ik}^{d} + TI_{ij} - M(1 - x_{ijk}) \quad i = 0,1,2...,n \quad j = 0,1,2,...,n \quad k = 1,2,...,K \tag{6}$$

$$\sum_{i=0}^{n} x_{ijk} t_{j}^{a} \leq t_{jk} \leq \sum_{i=0}^{n} x_{ijk} t_{j}^{b} \qquad j = 1,2,...n \quad k = 1,2,...,K \tag{7}$$

$$d_{ik} = \sum_{j=0}^{n} x_{jik}(C_i - I_i + u_i t_{ik}) \quad i = 1,2,...,n \qquad k = 1,2,...,K \tag{8}$$

$$t_{0k}^{d} = 0 \qquad\qquad k = 1,2,...,K \tag{9}$$

$$t_{ik}^{d} = t_{ik} + d_{ik} / R \qquad i = 1,2,...,n \quad k = 1,2,...,K \tag{10}$$

$$t_{ik} \geq 0\, , t_{ik}^{d} \geq 0\, , \qquad i = 0,1,2...,n \quad k = 1,2,...,K \tag{11}$$

The objective is to minimize the maximum route travel time among all the vehicles. Constraints (2) ensure that each gas station is visited only once. Constraints (3)-(5) characterize the flow on the path to be followed by each vehicle. Constraints

(6), which contain a large number *M*, eliminate sub-tours in each route. Constraints (7) specify the time window constraint for each gas station. Constraints (8) determine delivery quantity for each gas station, when it is visited by vehicle *k*. Constraints (9) and (10) determine the time that vehicle *k* leaves the depot and the stations.

4. The Tabu Search Algorithm

If we assume that there is only one vehicle in this problem, the problem is obviously more complex than a Traveling Salesman Problem (TSP). Therefore, this problem is a NP-Hard problem. Due to the high complexity of the problem, exact methods become inapplicable for solving large scale instances that often arise in practice. Attention can be paid to meta-heuristic methods that are able to produce near optimal solutions for such problems within reasonable computing time. Tabu search is a local search meta-heuristic proposed independently by Glover (1986) and Hansen and Jaumard (1990). According to recent literature, tabu search is widely applied for solving vehicle routing problems (Cordeau, et al., 1997; Archetti et al., 2006; Zachariadis et al., 2009). It is proved to be an effective meta-heuristic method for large scale VRP problems. Thus, as a first attempt for solving the problem in this paper, a tabu search algorithm is proposed.

4.1 Initial solution construction

To construct an initial feasible solution, we first sequence the gas stations in non-decreasing order of t_i^b. Gas stations are subsequently inserted into the vehicle routes in a greedy manner. The obtained initial solution consists of several (vehicle) routes. Each route can be treated as the route of a TSP and therefore can be improved by a method adapted from the Unstring and String (US) algorithm proposed by Gendreau et al. (1992). The steps of constructing the initial solution are presented below.

Step 1: sequence the gas stations in non-decreasing order of t_i^b. In case there is a tie of value t_i^b, stations are sequenced in non-increasing order of available empty capacity, i.e., $C_i - I_i$.

Step 2: take the first station in the sequence as current station, append the station to the route that the travel time of the route with current station appended is minimized.

Step 3: repeat Step 2 until all stations are inserted into vehicle routes.

4.2 Adapted US algorithm to improve the initial and neighbor solutions

Gendreau et al. (1992) propose the US algorithm to improve the solution for the TSP. The algorithm first deletes selected nodes from the route and then re-inserts these nodes into the route by some pre-designed rules. The work shows that this algorithm is a very efficient method for the TSP. This procedure is widely employed in tabu search algorithm to improve the neighborhood solutions (Cordeau et al., 1997; Archetti et al., 2006; etc.). We modify the standard US algorithm in Gendreau et al. (1992) to improve the solutions of the tabu search algorithm in this paper according to the problem features. In the adapted US algorithm, the search is more extensive in some steps but less in other steps. The detailed differences are listed below:

1. For each route, there are time window constraints for customers and maximum operation hour constraint for the vehicle. Thus, there might be fewer customers in each route, compared to standard VRP for the similar sized problems. Hence, only type 1 unstring step and type 1 string step in Gendreau et al. (1992) are employed in the adapted algorithm.
2. In the standard US algorithm, the two adjacent nodes of the insertion point are selected from the *p*-neighborhood of the insertion point. Here we simplify it by selecting the two nearest nodes within the *p*-neighborhood of the insertion point as the two adjacent nodes.
3. For a given route in the solution of the tabu search, we enumerate all the possible sequences and select the best one if the number of nodes does not exceed 7 in the route considering the computational power advancement in the past two decades.

From preliminary experiments, we set p = 3 in the adapted US algorithm. In

Gendreau et al. (1992), the complexity of the original procedure is $o(np^4 + n^2)$. As mentioned by the above item 2, the selection of the insertion point is simplified. Thus, the complexity of the corresponding step is decreased from $o(p^4)$ to $o(p^3)$. Therefore, the complexity of the adapted US algorithm is $o(np^3 + n^2)$.

4.3 The tabu search algorithm

The tabu search algorithm starts from the initial feasible solution and improves the solution quality by generating a set of neighbors in the hope that better solutions will be found. A tabu list is maintained during the search to prevent the search from getting stuck at local optimal solutions.

4.3.1 Neighbor generation

A neighbor can be generated by swapping pieces of any two routes in the current solution. For any two routes, once a cutting point is specified in each route, the two pieces from the cutting point to the end of the route can be exchanged to generate one neighbor solution.

Another way to generate neighbor solutions is to move one node in the longest route and insert it into the remaining routes with less travel time. This neighbor generation operation reflects the objective of min-max the route time.

As mentioned earlier, each neighbor consists of several routes. These routes can be improved by the adapted US algorithm. Prins (2004) shows that for searching algorithm, better results can be obtained if each randomly generated neighbor can be improved to be optimal. Though it is generally not possible to obtain the optimal solution for each generated neighbor due to computational time constraint, it is still desirable to apply the adapted US algorithm to obtain better TSP tours within allocated time.

4.3.2 The tabu list

We define each item in the tabu list by two parts. The first part is the two routes, from which the two neighbors can be generated. The second part is the cutting points in the routes with maximum route travel time. When generating a neighbor by switching the two pieces, if the selected two routes and the cutting point are identical to an item in the tabu list, no neighbor will be generated. On the other hand, when generating a neighbor by inserting a node from a route to another route, if the selected two routes and the cutting point, which is on the left side of the selected node, are identical to an item in the tabu list, this move is forbidden. According to preliminary experiments, the tabu list length is set to $\sqrt{n}$, where n is the number of nodes in the network.

4.3.3 The tabu search algorithm

Before presenting the detailed steps, we first define the notations. Let s^* denotes the initial solution. *CS* and *BS* indicate current solution and the best solution, respectively. It is obvious that a solution consists of several routes. The route with longest travel time is denoted by Ω. Y denotes the index of the remaining routes in current solution. Let X denote the position index in Ω, the counter "*Non_I*" to record the number of times the solution has not been improved and *Div_Num* to denote the number of times that diversification is executed. The detailed steps are illustrated in Algorithm 1.

Algorithm 1: The tabu search algorithm

```
Generate $s^*$ and apply the adapted US algorithm to improve $s^*$;
Initialize the tabu list; $CS = s^*$; $BS = s^*$; $Div\_Num = 0$;
while $Div\_Num \leqslant 25$ do
    $X = 1$, $Y = 1$, $Non\_I = 0$;
    while $Y$ does not exceed the index of the last route in $CS$ do
        while $X$ is less than the last customer position index of $\Omega$ or $Y$ do
            The two pieces are cut from position $X$ to end in $\Omega$ and route $Y$ in $CS$;
            A candidate neighbor is generated by switching the two pieces;
            Apply the adapted US algorithm to improve the candidate neighbor;
            Record the candidate neighbor, $X = X + 1$;
        $X = 1$;
        while $X$ is less than the last customer position index of $\Omega$ or $Y$ do
            Select node $X$ in $\Omega$ and insert it into all possible positions in route $Y$;
            Apply the adapted US algorithm to improve the candidate neighbors;
            Record the candidate neighbors, $X = X + 1$;
        $Y = Y + 1$, $X = 1$;
        Sort the generated neighbors in non-decreasing order of objective value;
        Select the best generated neighbor;
        $Accept\_flag$ = FALSE;
        while Accept_flag is FALSE do
            if the selected neighbor is non-tabu then
                if the selected neighbor is better than $BS$ then
                    Update the best solution; clear tabu list (intensification),
                    $Non\_I = 0$;
                    $CS$ = the selected neighbor; $Accepted\_flag$ = TRUE;
                else
                    $CS$ = the selected neighbor; $Accepted\_flag$ = TRUE; update the
                    tabu list;
                    $Non\_I = Non\_I + 1$;
                    if $Non\_I >$ pre-specified value then
                        Clear the tabu list (diversification);
                        $Div\_Num = Div\_Num + 1$; $Non\_I = 0$;
            else
                if the selected neighbor is better than $BS$ then
                    Update the best solution; clear tabu list (aspersion);
                    $Non\_I = 0$;
                    $CS$ = the selected neighbor, $Accept\_flag$ = TRUE; update tabu
                    list;
                else
                    Select the next generated neighbor, $Accept\_flag$ = FALSE;
Report the best solution $BS$;
```

5. Lower Bound by Lagrangian Relaxation Technique

In order to validate the performance of the tabu search algorithm, lower bounds of the problem are developed via Lagrangian relaxation technique. By relaxing some constraints, the original complex problem can be converted to a tractable one which then can yield a lower bound of the optimal solution to the original problem. The foundations of the Lagrangian relaxation technique and related applications can be referred to Fisher (1981, 1985) and Geoffrion (1974). Recent successful applications can be found, for example, in Kallehauge et al. (2006) and Imai et al. (2007), etc.

We select Constraints (2) in the original model to be relaxed with Lagrangian multipliers α_i. Thus, the model can be presented as:

$$LR = \min((\max_k t_{0k}) + \sum_{i=1}^{n} \alpha_i (\sum_{k=1}^{K} \sum_{j=0, j \neq i}^{n} x_{ijk} - 1)) \tag{12}$$

Subject to constraints (3)-(11).

5.1 Problem decomposition

The purpose of the Lagrangian relaxation technique is to create a problem that can be decomposed into a number of subproblems with known usable structures (Jörnsten et al., 1985; Guignard and Kim, 1987). Normally, the resulting model decomposes the original problem into one subproblem for each vehicle (Kohl and Madsen, 1997). However, the min-max format in objective function indicates that the relaxed model cannot be easily decomposed at the first glance. Fortunately, the following property of the relaxed model implies that the relaxed model can be decomposed.

Property 1. For the optimal solution of the relaxed problem, each vehicle's route can be identical.

Proof: We assume that the routes are not identical in the optimal solution of the relaxed problem. The routes can be denoted by $(i_1^1, i_1^2, ... i_1^{l_1})$,…, $(i_k^1, i_k^2, ..., i_k^{l_k})$,…, $(i_K^1, i_K^2, ..., i_K^{l_K})$. i_k^l indicates the l^{th} gas station in vehicle *k*'s route, where $k = 1, 2, ..., K, l = 1, 2, ..., l_k$. l_k indicates the number of gas station in vehicle *k*'s route. Assume that vehicle *k** has the maximum travel time among the routes of the optimal solution. Thus, $t_{0k^*} = \max_k \{t_{0k}\}$. In (12), the term $\sum_{i=1}^{n} \alpha_i \sum_{j=0, j \neq i}^{n} x_{ijk}$ denotes the sum of α_i on vehicle *k's* route. Thus, the objective function can be written as:

$$LR = t_{0k^*} + \sum_{k=1}^{K} \sum_{l=1}^{l_k} \alpha_{i_k^l} - \sum_{i=1}^{n} \alpha_i \,. \tag{13}$$

Let s_k denote part of the second term, i.e., $s_k = \alpha_{i_k^1} + \alpha_{i_k^2} + \ldots + \alpha_{i_k^{l_k}}$. The second term can be written as $\sum_{k=1}^{K} s_k$. For the route of any vehicle $p\,(p \neq k^*)$ in the solution, we have $t_{0k^*} \geq t_{0p}$. If there is only one route has the value of t_{0k^*}, we prove that each vehicle's route can be identical.

(1) $s_{k^*} < s_p$

If vehicle *p* abandons its route and take the route that is identical to *k**, we have $t_{0p} = t_{0k^*}$, $s_p = s_{k^*}$. Thus, the value is not changed for the first term in (13), while the value of the second term is decreased. Therefore, we obtain a better solution by change vehicle *p*'s route. It is straightforward that we can replace the routes of any vehicle *k* with $s_{k^*} < s_k$ to improve the solution.

(2) $s_{k^*} = s_p$

This condition indicates that the sum of the multipliers for each route is identical. Probably, the set of gas stations is identical for the two routes. The only difference is the visiting sequence of gas stations. Let $t_{p^*} = \min_p \{t_{0p}\}$ for all routes *p* with $s_p = s_{k^*}$ Replacing the routes of all vehicle *p* and vehicle *k** by the route of vehicle *p**, the solution of (12) can be improved.

(3) $s_{k^*} > s_p$

In this case, it is obvious that a better solution will be obtained by replacing the route of vehicle k^* by the best of the remaining routes. Similar to condition (2), let $t_{p^*} = \min_p \{t_{0p}\}$ for all routes *p* with $s_p < s_{k^*}$ Then, replace the routes of all vehicle *p* and vehicle k* by the route of vehicle *p**, the solution of (12) can be improved.

Therefore, if there is only one route has the value of t_{0k^*}, each vehicle's route should be identical in the optimal solution of the relaxed problem. On the other hand, if there is more than one route have the value of t_{0k^*}, such as $t_{0k1^*} = t_{0k2^*} = t_{0k^*}$ we prove that these routes *k1** and *k2** can be converted to only one route.

(4) $s_{k1^*} \neq s_{k2^*}$

Similar to case (1), the route with larger value of *s* will be replaced with the other route with a smaller value of *s*. Therefore, we obtain a better solution by reducing the value of the second term in (13).

(5) $s_{k1^*} = s_{k2^*}$

Since the route travel time and the sum of the multipliers of the two routes are identical, any route can be replaced by the other route without changing the value of (13).

Thus, for the optimal solution of the relaxed problem, each vehicle's route can be identical. □

According to Property 1, the route of each vehicle should have identical travel time. Let t_{n+1} represents all t_{0k}. Thus, (12) can be re-written as:

$$LR = \min t_{n+1} + \sum_{i=1}^{n} \alpha_i \left(\sum_{k=1}^{K} \sum_{j=0, j\neq i}^{n} x_{ijk} - 1\right) \qquad (14)$$

By Property 1, each vehicle's route is identical. Thus, the original problem can be decomposed into sub-problems for each vehicle. Also, the variable x_{ijk} can be replaced by x_{ij}. The second term of (14) consists of the total of the corresponding term in each sub-problem. By decomposing (14), the model of each subproblem is presented in (15) to (25). For the convenience of presentation, the depot that each vehicle returns to is represented by *n* + 1 in the model.

$$LR_k = \min(t_{n+1} + K\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n+1} \alpha_i x_{ij} - \sum_{i=0}^{n} \alpha_i) / K \qquad (15)$$

Subject to：

$$\sum_{j=1}^{n} x_{0j} = 1, \tag{16}$$

$$\sum_{j=1}^{n} x_{j,n+1} = 1, \tag{17}$$

$$\sum_{i=0}^{n} x_{ih} - \sum_{j=0}^{n} x_{hj} = 0 \qquad h = 1, 2, ..., n \tag{18}$$

$$t_0^d = 0 \tag{19}$$

$$\sum_{i=0}^{n} x_{ij} t_j^a \le t_j \le \sum_{i=0}^{n} x_{ij} t_j^b \qquad j = 1, 2, ..., n \text{ ,} \tag{20}$$

$$d_i = \sum_{j=0}^{n} x_{ji} (C_i - I_i + u_i t_i) \qquad i = 1,2,...,n \tag{21}$$

$$t_i^d = t_i + d_i / R \qquad i = 1,2,...,n \tag{22}$$

$$t_j \ge t_i^d + TI_{ij} - M(1 - x_{ij}) \qquad i = 0,1,2...,n \qquad j = 1,2,...,n,n+1 \tag{23}$$

$$t_i \ge 0 \text{ , } t_i^d \ge 0 \qquad i = 0,1,2...,n \tag{24}$$

$$x_{ij} = 0,1 \qquad i = 0,1,2...,n \qquad j = 1,2,...,n,n+1 \tag{25}$$

The objective is the minimization of the travel time of each route. If we re-write t_{n+1} as $t_{n+1} = \sum_{i=0}^{n} \sum_{j=1}^{n+1} (t_j - t_i) x_{ij}$, thus, the objective function can be re-written as $LR_k = \min(\sum_{i=0}^{n} \sum_{j=1}^{n+1} (t_j - t_i + \alpha_i K) x_{ij} - \sum_{i=1}^{n} \alpha_i) / K$. Since K and α_i are constants, the subproblem can be treated as a shortest path problem with time constraints (Desrosiers et al., 1995).

5.2 Dynamic programming algorithm for solving the sub-problem

We develop a dynamic programming (DP) algorithm to solve the subproblem. The algorithm extends the dynamic algorithm in Desrosiers et al. (1995) to avoid cyclic routes in the network. Before presenting the algorithm, the following notation is explained.

Let $c_{ij} = t_j - t_i + \alpha_i K$, which denotes the travel cost from node *i* to node *j*. For node *i*, there is a set of labels. Each label can be denoted by $(T_i^k, C_i^k, Front(v_i^k, vt_i^k))$. T_i^k and C_i^k indicate the travel time and cost of the k^{th} path from *o* to *i*, respectively. T_i^k also indicates the delivery start time of the vehicle for customer *i*. On the k^{th} path, v_i^k denotes the immediate predecessor of *i*. vt_i^k denotes the route index from *o* to v_i^k. Q_i is the set of labels for node *i*. P_i refers the set of labels that is exploited in the DP procedure. N denotes the set of nodes. o and d denote the depot that the vehicle departs and returns to, respectively. V denotes nodes including N and *d*. $\Gamma(i)$ denotes successors of node *i*. To avoid cyclic routes, if a node is included in the route from *o* to *i*, the node cannot be included in $\Gamma(i)$.

5.2.1 Label domination condition

Similar to Desrosiers et al. (1995), among the set of labels of node *i*, we define the condition that one label dominates another label.

Definition. $(T_i^{k1}, C_i^{k1}, Front(v_i^{k1}, vt_i^{k1}))$ dominates $(T_i^{k2}, C_i^{k2}, Front(v_i^{k2}, vt_i^{k2}))$ if and only if $T_i^{k1} \le T_i^{k2}$, $C_i^{k1} \le C_i^{k2}$, $C_i^{k1} - T_i^{k1} \le C_i^{k2} - T_i^{k2}$, and at least one of the inequalities holds strictly.

If we extend the sub-route 0-*i* to node *j*, the labels are updated, so that $(T_j^{k1}, C_j^{k1}, Front(v_j^{k1}, vt_j^{k1}))$ and $(T_j^{k2}, C_j^{k2}, Front(v_j^{k2}, vt_j^{k2}))$ are obtained. The first sub-route is better than the second if $T_j^{k1} \le T_j^{k2}$ and $C_j^{k1} \le C_j^{k2}$. We now show that why $C_i^{k1} - T_i^{k1} \le C_i^{k2} - T_i^{k2}$ is appended compared to the domination definition in Desrosiers et al. (1995).

When we extend the sub-route *0-i* to node *j*, $T_j^{k1} = \max\{t_j^a, T_i^{k1} + T_{ij}\}$, $T_j^{k2} = \max\{t_j^a, T_i^{k2} + T_{ij}\}$, $C_j^{k1} = C_i^{k1} + c_{ij}, C_j^{k2} = C_i^{k2} + c_{ij}$. According to the definition of T_j^k and t_j^k, it is obvious that $T_j^k = t_j^k$. If c_{ij} is constant, the domination condition should be $T_i^{k1} \le T_i^{k2}$, $C_i^{k1} \le C_i^{k2}$, which is given by Desrosiers et al. (1995). However, $c_{ij} = t_j - t_i + \alpha_i K$ in this sub-problem. Let AT_j denote the arrival time of the vehicle to node j. Thus,

$$AT_j = t_i + \frac{(C_i - I_i + u_i t_i)}{R} + TI_{ij} \tag{26}$$

Then, $$c_{ij} = \begin{cases} \dfrac{(C_i - I_i + u_i t_i)}{R} + TI_{ij} + \alpha_i K, & if \quad AT_j > t_j^a \\ t_j^a - t_i + \alpha_i K, & if \quad AT_j \le t_j^a \end{cases} \tag{27}$$

(1). If $AT_j^{k1} \le AT_j^{k2} \le t_j^a$.

According to (26) and (27), $C_j^{k1} = C_i^{k1} + t_j^a - T_i^{k1} + \alpha_i K$,$C_j^{k2} = C_i^{k2} + t_j^a - T_i^{k2} + \alpha_i K$.

If $C_i^{k1} - T_i^{k1} \le C_i^{k2} - T_i^{k2}$, $C_j^{k1} \le C_j^{k2}$

(2). If $AT_j^{k1} \le t_j^a \le AT_j^{k2}$

$$C_j^{k1} = C_i^{k1} + t_j^a - t_i^{k1} + \alpha_i K\,,\ C_j^{k2} = C_i^{k2} + \frac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij} + \alpha_i K$$

Since $AT_j^{k2} = t_i^{k2} + \dfrac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij}$, we obtain $t_j^a \le t_i^{k2} + \dfrac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij}$.

Then, $$C_j^{k1} = C_i^{k1} - T_i^{k1} + t_j^a + \alpha_i K \le C_i^{k1} - T_i^{k1} + t_i^{k2} + \frac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij} + \alpha_i K \tag{28}$$

If $C_i^{k1} - T_i^{k1} \le C_i^{k2} - T_i^{k2}$,

$$C_j^{k1} = C_i^{k1} - T_i^{k1} + t_i^{k2} + \frac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij} + \alpha_i K \le C_i^{k2} - T_i^{k2} + t_i^{k2} + \frac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij} + \alpha_i K$$

As T_i^k also indicates the delivery start time of vehicle *k* for customer *i*,

$$C_i^{k2} - T_i^{k2} + t_i^{k2} + \frac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij} + \alpha_i K = C_i^{k2} + \frac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij} + \alpha_i K = C_j^{k2}$$

Thus, we obtain $C_j^{k1} \le C_j^{k2}$.

(3). If $t_j^a \le AT_j^{k1} \le AT_j^{k2}$

$$C_j^{k1} = C_i^{k1} + \frac{(C_i - I_i + u_i t_i^{k1})}{R} + TI_{ij} + \alpha_i K \text{ , } C_j^{k2} = C_i^{k2} + \frac{(C_i - I_i + u_i t_i^{k2})}{R} + TI_{ij} + \alpha_i K$$

If $T_i^{k1} \le T_i^{k2}$, and $C_i^{k1} \le C_i^{k2}$, we obtain $C_j^{k1} \le C_j^{k2}$.

Therefore, the condition of $T_i^{k1} \le T_i^{k2}$, $C_i^{k1} \le C_i^{k2}$ and $C_i^{k1} - T_i^{k1} \le C_i^{k2} - T_i^{k2}$ ensure that $T_j^{k1} \le T_j^{k2}$ and $C_j^{k1} \le C_j^{k2}$, which indicate that $(T_i^{k1}, C_i^{k1}, Front(v_i^{k1}, vt_i^{k1}))$ dominates $(T_i^{k2}, C_i^{k2}, Front(v_i^{k2}, vt_i^{k2}))$.

5.2.2 The dynamic programming algorithm

Let $EFF(T_i^{k1}, C_i^{k1}, Front(v_i^{k1}, vt_i^{k1}))$ denotes the set of non-dominated labels of node *i*. The algorithm consists of the following steps.

Step 1. Initialization

$$Q_0 = \{(T_0^1 = 0, C_0^1 = 0, Front(0,0))\};\ Q_i = \varnothing, \forall i \in N \cup \{d\};\ P_i = \varnothing, \forall i \in V$$

Step 2.selection of node $i*$

From the set $U_i \in (Q_i \setminus P_i), \forall i \in N \cup \{o\}$, select the label $(T_{i^*}^k, C_{i^*}^k, Front(v_{i*}^k, vt_{i*}^k))$ with minimum value of T_i^k . If $U_i \in (Q_i \setminus P_i) = \varnothing, \forall i \in N \cup \{o\}$, stop， go to step 4. Otherwise, go to next step.

Step 3： treatment of label $(T_{i^*}^k, C_{i^*}^k, Front(v_{i*}^k, vt_{i*}^k))$

For all $j \in \Gamma(i^*)$,

$Q_j = EFF(f_{i^*j}(T_{i^*}^k, C_{i^*}^k, Front(v_{i*}^k, vt_{i*}^k)) \cup Q_j)$, $P_i = P_i \cup \{(T_{i^*}^k, C_{i^*}^k, Front(v_{i*}^k, vt_{i*}^k))\}$.

$f_{i^*j}(T_{i^*}^k, C_{i^*}^k, Front(v_{i*}^k, vt_{i*}^k))$ is the treatment of labels of the node *j*, which is the successor of node $i*$. If $T_{i^*}^k + (C_{i^*} - I_{i^*} + u_{i^*} T_{i^*}^k)/R + TI_{i^*j} \le t_j^b$,

$$f_{i^*j}(T_{i^*}^k, C_{i^*}^k, Front(v_{i*}^k, vt_{i*}^k)) = \\ (\max\{t_j^a, T_{i^*}^k + (C_{i^*} - I_{i^*} + u_{i^*} T_{i^*}^k)/R + TI_{i^*j}\}, C_{i^*}^k + c_{i^*j}, Front(i^*, k)) \quad (29)$$

The term $T_{i^*}^k + (C_{i^*} - I_{i^*} + u_{i^*} T_{i^*}^k)/R + TI_{i^*j}$ denotes the sum of delivery start time at node *i*, the delivery time of the vehicle at node *i* and the travel time between node *i* and *j*. $[t_j^a, t_j^b]$ is the time windows for delivery at node *j*. It is obvious that if $T_{i^*}^k + (C_{i^*} - I_{i^*} + u_{i^*} T_{i^*}^k)/R + TI_{i^* j} > t_j^b$, $f_{i^* j}(T_{i^*}^k, C_{i^*}^k, Front(v_{i*}^k, vt_{i*}^k))$ is empty.

When the treatment is executed, return to step 2.

Step 4: from all the labels of {*d*}, select the label $T_d^{k^*}$ with the minimum value of T_d^k. Then, find the shortest path from this label.

5.3 Lower bound calculation

The subgradiant method is used to find the appropriate multipliers for the Lagrangian relaxation technique. In the subgradient method, a sequence of Lagrangian multipliers is generated by the formula:

$$\alpha_i^{s+1} = \alpha_i^s + t_s * (K \sum_{j=1}^{n+1} x_{ij} - 1), \tag{30}$$

Where t_s is a positive scalar step size defined as:

$$t_s = \frac{u_s * (Z - LB)}{\sum_{i=1}^{n} (K \sum_{j=1}^{n+1} x_{ij} - 1)^2} \tag{31}$$

The initial multipliers can be α_i =0. μ_s is a scalar satisfying $0 < \mu_s \le 2$. *Z* is an upper bound on the optimal solution, which can be obtained by heuristics to the original problem. *LB* is the lower bound, which is obtained by the dynamic programming algorithm.

The steps of the Lagrangian relaxation technique are:

Step 1: initialization. Set α_f =0, μ_s =2, Maximum iteration *kmax*=1000, initial lower bound *LB*=0, initial upper bound *UB*=20000, ε =0.05, a parameter named *Max_No_Improve* is set to 4, which will be used in the following step.

Step 2：calculate relaxed lower bound *LR* and upper bound *Z*, LB=Max(LR, LB). if *LB* is not increased in *Max_No_Improve* iterations, take μ_n to its half.

Step 3：calculate t_s and α_i

Step 4：if iteration number reaches its maximum or $100 \times \frac{(Z - LB)}{Z} \leq \varepsilon$, stop. Return *LB*. Otherwise, return to Step1.

The upper bound in Step 2 is generated by the tabu search algorithm. When applying the Lagrangian relaxation technique, we observe that the iteration is not always necessary to be 1000. In most of the cases, when μ_n is reduced to a certain value, *LR* cannot be increased. Thus, if *LR* stops increasing for some value of μ_n and the subsequent value of $\mu_n / 2$, we terminate the algorithm to save computing time.

6. Computational Results

In this section, the performance of the tabu search algorithm is evaluated by computational experiments based on randomly generated instances. Campbell and Savelsbergh (2004a) work on an industrial gas delivery problem, which is similar to the problem studied in this paper. Therefore, we generate our data in the way that is similar to the procedure described in Campbell and Savelsbergh (2004a). However, the ranges of the randomly generated data are slightly adjusted according to the assumptions in this paper. Three sets of data are generated based on 5–9 gas stations (first set S1), 10–100 gas stations (second set S2) and 60–200 gas stations (third set S3), respectively. The storage capacity of the gas stations is generated in [288, 312] with an average of 300, which is a unitless measure without losing generality. The same applies to other similar parameters such as consumption and vehicle delivery rates. The average consumption rate of each gas station is 15 per hour. We allow the value to vary by 20% below or above the average value. It is safe to assume that the consumption rate in the day time is higher than that during the night. Based on observation and communications with employees from the company, the consumption rate during the 8-hour delivery time (day time) period is set to 2 times over the consumption rate of the 16-hour low demand time period (non-delivery time period, i.e., night time). In our model, the decision only cares about the operations within an 8-hour delivery time period. Thus, in order to simplify the model, the low demand time period can be condensed to 8 hours with identical consumption rate within the delivery time period. The initial inventory at each gas station is set to be the consumption quantity between 4 to 16 hours at the delivery time period consumption rate. Also, the minimum delivery quantity at each gas station should satisfy at least one day's consumption. It is set to be 20 hours consumption quantity minus the initial inventory. The position of the depot and each gas station is represented by the X and Y coordinates, which are generated in [0, 70]. There are 2–20 vehicles in each problem, with the travel speed of 40 per hour. The delivery rate of each vehicle to each gas station is set to 900 per hour. For each problem configuration, 5 instances are generated and the average is presented as the reported results.

The tabu search algorithm and the Lagrangian relaxation technique were coded in C++. The computational experiments are conducted on a personal computer with Intel Pentium dual-core 2.8 GHz CPU and 2 GB RAM. The test instances with 5–9 gas stations are employed to validate that whether the tabu search can obtain the optimal solution for small sized problems, which are solved by CPLEX 12.0.

The computational results are shown in Tables 1 to 3. Table 1 provides optimal solutions obtained by CPLEX and the results of the tabu search algorithm for the five instances generated for each combination of gas station number and vehicle number. It is shown that the tabu search algorithm can obtain the optimal solutions for all of the small sized instances. It is also observed from Table 1 that the tabu search algorithm spends shorter time for most of the problems. The gap of CPU time increases as the problem size increases. Note that CPLEX can only solve the problem with 9 gas stations and 2 vehicles. For larger sized problem, CPLEX cannot provide the optimal solution within 8 hours, which is a shift time period in reality. Thus, we can find that for small-scale test instances the tabu search algorithm dominates CPLEX.

[Insert Table 1 here]

In order to evaluate the performance of the presented tabu search algorithm for solving larger sized problem, we compare the results with the lower bound obtained using the procedure described in Section 5. The results of the average travel time, lower bounds and computational time for five instances of each problem in the second set are reported in Table 2.

[Insert Table 2 here]

For small to medium sized problems in the second set, with the exception of problem configuration number 17 which is the largest instance in the test set and has a gap of 17.9%, the maximum of the average gap between the solution of the tabu search algorithm and the lower bound provided by the Lagrangian relaxation technique is less than 8.1%. It implies that the tabu search algorithm can obtain solutions which are quite close to the optimal ones. Also, the results indicate that the lower bound obtained by the Lagrangian relaxation technique is rather tight for most of the test instances in the set. It is worthwhile to mention that the computational time of the tabu search is very competitive compared to that of the Lagrangian relaxation technique.

As discussed earlier, an adapted US algorithm is applied in each iteration of the tabu search algorithm. To evaluate the effectiveness of the adapted US algorithm, the tabu search algorithm is compared to its duplication where the adapted US algorithm is removed (i.e., by removing Steps 9 and 14 from Algorithm 1). It should be mentioned that if the adapted US algorithm is totally removed from the tabu search algorithm, it is hard to find a feasible solution. Therefore, the adapted US algorithm is still applied to the selected best neighbor (i.e., applied to the solution in Step 18 in Algorithm 1) in each iteration of the tabu search. Meanwhile, we apply the standard US algorithm, i.e., without the adaptation, to the tabu search to see whether the adapted US algorithm contribute to the solutions. The computational results are shown in Table 3, where the three approaches are marked as “without”, “with standard” and “with adapted” which are linked with the US algorithm. The columns of “Gap (%)” below “without” and “with standard” measure the relative difference of the particular approach to the tabu search with the adapted US algorithm.

[Insert Table 3 here]

The results indicate that the tabu search with the adapted US algorithm obtain much better results compared to the tabu search without the adapted US algorithm. The average gap is about 12.1%. Although the computational time of the former algorithm is longer than the latter, the time is still acceptable in practice. The results indicate that the adapted US algorithm enhance the performance of the tabu search algorithm considerably.

Since standard the US algorithm is quite time consuming, it is also only applied for the selected neighbor in each iteration of the tabu search algorithm. It is shown that the tabu search with standard US algorithm spent less time for smaller problem. However, the CPU time increases quickly when the problem size increases. It is also shown that the tabu search with adapted US algorithm performs better than tabu search with the standard US algorithm. For all of the 15 problems, tabu search with adapted US algorithm is slightly inferior to tabu search with standard US algorithm for 7 problems. However, for the remaining 8 problems, tabu search with adapted US algorithm clearly outperforms. The last row in Table 3 summarizes the performance of the three approaches across all the test instances in the third test set, which clearly shows that the tabu search with adapted US algorithm outperforms the one without by 12.1%, and slightly outperforms the one with standard US algorithm with a performance gap of 1.0%.

The results in Table 3 indicate that it is preferable to apply the adapted US algorithm or the standard US algorithm for each generated neighbor. However, as the standard US algorithm is quite time consuming, it can only be applied to one selected neighbor in each iteration. The adapted US algorithm can be applied to each generated neighbor to enhance the performance of the tabu search algorithm while maintain the time-effectiveness of the approach.

7. Conclusion

In this paper, we investigate a new variant of the inventory-routing problem in the gasoline distribution industry. We consider a couple of special constraints in this industry, which motivate us to concentrate on the vehicle travel time minimization as the objective. For this NP-hard problem, we have proposed a tabu search algorithm and developed an adapted US algorithm as part of the solution which improves the solution quality of the tabu search. For evaluating the performance of the proposed approach, we employ the Lagrangian relaxation technique to obtain the lower bound of the problem which decomposes the relaxed problem by exploiting its properties. Computational results show that the solution of the tabu search algorithm is quite close to the lower bounds for small to medium sized problems. It indicates that the tabu search algorithm can provide near optimal solution for these problems and the Lagrangian relaxation technique provides tight lower bounds. Computational results also demonstrate that the adapted US algorithm can improve the performance of the tabu search considerably.

Although the lower bound provided by the Lagrangian relaxation technique is tight, the computational time is relatively long. Thus, it is an interesting direction to reduce the computational complexity of the lower bound while not sacrificing the quality. From a practical perspective, a natural next step to tackle the problem is to consider the vehicle capacity constraint while routing the fleet of distribution vehicles. How to balance the maximum travel time and the overall distribution system cost would also be interesting and the trade-off between time and cost needs to be examined as future work as well.

Acknowledgements

This research was supported by the National Nature Science Foundation of China (No. 70972019 and 71131004).

References:

Abdelmaguid, T.F., Dessouky, M.M., Ordóñez, F. (2009). Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering*, 56, 1519-1534.

Adulyasak, Y., Cordeau, J.-F., Jans, R. (2012). *Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems*. Technical Report G-2012-14, GERAD, Montreal, Canada.

Alegre, J., Laguna, M., Pacheco, J. (2007). Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*, 179, 736-746.

Al-Khayyal, F., Hwang, S-J. (2007). Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: applications and model. *European Journal of Operational Research*, 176, 106-113.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37, 1515-1536.

Archetti, C., Speranza, M.G., Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40(1), 64-73.

Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41 (3), 382–391.

Archetti, C., Bertazzi, L., Paletta, G., Speranza, M.G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 12(38), 1731–46.

Archetti, C., Bertazzi, L., Hertz, A., Speranza, M.G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing,* 24(1), 101-116.

Archibald, T.W., Black, D.P., Glazebrook, K.D. (2009). Indexability and index heuristics for a simple class of inventory routing problems. *Operations Research*, 57(2), 314-326.

Avella, P., Boccia, M., Sforza, A. (2004). Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research,* 152(1), 170-179.

Bard, J.F., Nananukul, N. (2009a). Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, 57, 713-723.

Bard, J.F., Nananukul, N. (2009b). The integrated production inventory distribution routing problem. *Journal of Scheduling*, 12, 257-280.

Bell, W.J., Dalberto, L.M., Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13, 4-23.

Benoist, T., Gardi, F., Jeanjean, A. (2011). Randomized local search for real-life inventory routing. *Transportation Science*, 45(3), 381-398.

Bertazzi, L., Paletta, G., Speranza, M.G. (2002). Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1), 119–32.

Bertazzi, L. (2008). Analysis of direct shipping policies in an inventory-routing problem with discrete shipping times. *Management Science*, 54(4), 748-762.

Blumenfeld, D.E., Burns, L.D., Daganzo, C.F., Frick, M.C., Hall, R.W. (1987). Reducing logistic costs at General Motors. *Interfaces*, 17, 26-47.

Campbell, A.M., Savelsbergh, M.W.P. (2004a). Delivery volume optimization. *Transportation Science*, 38(2), 210-223.

Campbell, A.M., Savelsbergh, M.W.P. (2004b). A decomposition approach for the inventory-routing problem. *Transportation Science*, 38(4), 488-502.

Chen, Y.M., Lin, C.T. (2009). A coordinated approach to hedge the risks in stochastic inventory-routing problem. *Computers & Industrial Engineering*, 56, 1095-1112.

Coelho, L.C., Cordeau, J.-F., Laporte, G. (2012a). *Thirty Years of Inventory-Routing,* Technical Report.

https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2012-52.pdf.

Coelho, L.C., Cordeau, J.-F., Laporte, G. (2012b). The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11), 2537-2548.

Coelho, L.C., Cordeau, J.-F., Laporte, G. (2012c). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C,* 24(1), 270-287.

Ceolho, L.C., Laporte, G. (2012). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research.* 40(2), 558–565.

Cordeau, J.-F., Gendreau, M., Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30, 105-119.

Cornillier, F., Boctor, F.F., Laporte, G., Renaud, J. (2008). A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research,* 191, 295 ‒ 305

Cornillier F., Boctor F.F., Renaud J. (2012). Heuristics for the multi-depot petrol station replenishment problem with time windows. *European Journal of Operational Research,* 220 361 ‒ 369.

Custódio, A.L., Oliveira, C. (2006). Redesigning distribution operations: a case study on integrating inventory management and vehicle routes design. *International Journal of Logistics: Research and Applications*, 9, 169-187.

Dauzère-Pérès, S., Nordli, A., Olstad, A., Haugen, K., Koester, U., Olav, M. P., Teistklub, G., Reistad, A. (2007). Omya Hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to European paper manufactures. *Interfaces,* 37, 39-51.

Derigs, U., Gottlieb, J., Kalkoff., Piesche, M., Rothlauf, F., Vogel, U. (2011). Vehicle routing with compartments: applications, modelling and heuristics. *OR Spectrum*. 33:885–914

Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F. (1995). Time constraints routing and scheduling. In: M.O. Ball, T.L. Magnanti, C.L. Momma, G.L. Nemhauser, (Eds.), Network Routing, *Handbooks in Operations Research and Management Science 8,* (pp. 35-139), North Holland, Amsterdam.

Engineer, F.G, Furman, K.C., Nemhauser, G.L., Savelsbergh, M.W.P., Song, J.H. (2012). A branch-price-and-cut algorithm for single-product maritime inventory routing, *Operations Research*, 60(1), 106-122.

Fisher, M.L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1), 1-18.

Fisher, M.L. (1985). An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15(2), 10-21.

Gaur, V., Fisher, M.L. (2004). A periodic inventory routing problem at a supermarket chain. *Operations Research,* 52(6), 813-822.

Gendreau, M., Hertz, A., Laporte, G. (1992). New insertion and post-optimization procedures for the traveling salesman problem. *Operations Research,* 40(6), 1086-1095.

Geoffrion, A.M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study,* 2, 82-114.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research,* 13(5), 533-549.

Goel, A. (2010). Truck driver scheduling in the European Union. *Transportation Science*, 44(4), 429-441.

Goel, A., Kok, L. (2012). Truck driver scheduling in the United States. *Transportation Science,* 46(3), 317–326.

Grønhaug, R., Christiansen, M. (2010). A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science,* 44(3), 400-415.

Guignard, M., Kim, S. (1987). Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming,* 39, 215-228.

Hansen, P., Jaumard, B. (1990). Algorithms for the maximum satisfiability problem. *Computing*, 44(4), 279-303.

Hemmelmayr, V., Doerner, K.F., Hartl, R.F., Savelsbergh, M.V.P. (2009). Delivery strategies for blood products supplies. *OR Spectrum*, 31(4), 707-725.

Huang, S.H., Lin, P.C. (2010). A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E,* 46, 598-611.

Hvattum, L.M., Løkketangen, A. (2009). Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics,* 15, 527-557.

Hvattum, L.M., Løkketangen, A., Laporte, G. (2009). Scenario tree-based heuristics for stochastic invenroty-routing problems. *INFORMS Journal on Computing,* 21(2), 268-285.

Imai, A., Nishimura, E., Current, J. (2007). A Lagrangian relaxation-based heuristic for the vehicle routing with full container load. *European Journal of Operational Research*, 176, 87-105.

Jörnsten, K., Näsberg, M., Smeds, P. (1985). *Variable splitting—A new Lagrangean approach to some mathematical programming models*. Report LITHMAT-85-04, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden.

Kallehauge, B., Larsen, J., Madsen, O.B.G. (2006). Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research*, 33, 1464-1487.

Kang, J.-H., Kim, Y.-D. (2010). Coordination of inventory and transportation managements in a two-level supply chain. *International Journal of Production Economics,* 123, 137-145.

Kohl, N., Madsen, O.B.G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. *Operations Research,* 45(3), 395-406.

Li, J.X., Chu, F., Chen, H.X. (2011a). A solution approach to the inventory routing problem in a three-level distribution system. *European Journal of Operational Research*, 210, 736-744.

Li, J.X., Chu, F., Chen, H.X. (2011b). Coordination of split deliveries in one-warehouse multi-retailer distribution systems. *Computers & Industrial Engineering,* 60.291-301

Li, J.X., Chen, H.X., Chu, F. (2010). Performance evaluation of distribution strategies for the inventory routing problem. *European Journal of Operational Research*, 202, 412-419.

Liu, S.-C., Lee, W.-T. (2011). A heuristic method for the inventory routing problem with time windows. *Expert Systems with Applications,* 38, 13223-13231.

Michel, S., Vanderbeck, F. (2012). A column-generation based tactical planning method for inventory-routing. *Operations Research,* 60(2), 382-397.

Moin, N.H., Salhi, S., Aziz, N.A.B. (2011). An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics,* 133, 334-343.

Muyldermans, L., Pang, G. (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research*. 206. 93 – 103

Ng, W.L., Leung, S.C.H., Lam, J.K.P., Pan, S.W. (2008) Petrol delivery tanker assignment and routing: a case study in Hong Kong. *Journal of the Operational Research Society*. 59, 1191 -1200.

Ohlmann, J.W., Fry, M.J., Thomas, B.W. (2008). Route design for lean production systems. *Transportation Science,* 42(3), 352-370.

Oppen, J., Løkketangen, A., Desrosiers, J. (2010). Solving a rich vehicle routing and inventory problem using column generation. *Computers & Operations Research,* 37, 1308-1317.

Persson, J.A, Göthe-Lundgren, M. (2005). Shipment planning at oil refineries using column generation and valid inequalities. *European Journal of Operational Research,* 163, 631-652.

Popović, D., Vidović, M., Radivojević, G. (2012). Variable neighborhood search heuristic for the inventory routing problem in fuel delivery, *Expert Systems with Applications,* 39(18), 13390-13398.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research,* 31, 1985–2002.

Raa, B., Aghezzaf, E.-H. (2008). Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics,* 112, 255-263.

Raa, B., Aghezzaf, E.-H. (2009). A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192, 429-441.

Solyali, O., Süral, H. (2011). A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation Science*, 45(3), 335-345.

Toriello, A., Nemhauser, G.L., Savelsbergh, M. (2011). Decomposing inventory routing problems with approximate value functions. *Naval Research Logistics*, 57(8), 718-727.

Valle, C.A., Martinez, L.C., da Cunha, A.S., Mateus, G.R. (2011). Heuristic and exact algorithms for a min–max selective vehicle routing problem. *Computers & Operations Research*, 38, 1054-1065.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195, 729-743.

Zhao, Q.H., Chen, S., Zang, C.X. (2008). Model and algorithm for inventory/routing decision in a three-echelon logistics system. *European Journal of Operational Research*, 191, 623-635.

Zhao, Q.H., Wang, S.Y., Lai, K.K. (2007). A partition approach to the inventory /routing problem, *European Journal of Operational Research*, 177(2), 786-802.

Table 1. Computational results of CPLEX and tabu search algorithm

Prob	n	K	CPLEX		Tabu Search		
			$\max_k t_{0k}$	CPU (s)	$\max_k t_{0k}$	CPU (s)	Optimal?
S1.1	5	2	190.6	3.1	190.6	3.7	Yes
S1.2	6	2	239.9	3.4	239.9	4.0	Yes
S1.3	7	2	252.4	6.9	252.4	4.7	Yes
S1.4	8	2	249.6	47.5	249.6	5.5	Yes
S1.5	9	2	265.5	640.5	265.5	8.8	Yes

Table 2. Computational results of Lagrangian relaxation technique and tabu search algorithm

Prob	n	K	Lagrangian Relaxation		Tabu Search		Gap (%)
			LB	CPU (s)	$\max_k t_{0k}$	CPU (s)	
S2.1	10	2	194.5	4.5	207.5	10.4	6.2
S2.2	11	2	204.1	3.7	218.3	11.3	6.5
S2.3	12	2	212.0	5.8	230.7	11.0	8.1
S2.4	13	2	227.5	5.1	245.5	12.9	7.3
S2.5	14	2	246.7	5.5	264.3	11.6	6.6
S2.6	20	2	301.8	9.3	311.8	14.1	3.3
S2.7	25	2	374.3	12.7	390.9	14.2	4.4
S2.8	30	3	390.1	22.4	412.4	15.1	5.7
S2.9	35	3	390.8	38.1	410.6	20.6	5.1
S2.10	40	3	346.9	41.5	366.2	32.3	5.6
S2.11	45	3	378.9	120.9	398.5	36.1	5.2
S2.12	50	4	323.5	66.3	347.3	44.0	7.3
S2.13	60	4	389.5	127.3	412.4	80.2	5.9
S2.14	70	5	346.8	129.3	373.8	107.3	7.8
S2.15	80	5	395.0	331.8	421.9	124.3	6.8
S2.16	90	6	363.5	799.3	389.8	145.9	7.2
S2.17	100	7	325.6	886.8	384.0	171.3	17.9

Table 3. Computational results of the tabu search algorithm without, with the standard, and with the adapted US algorithm

Prob	***n***	***K***	**without**			**with standard**			**with adapted**	
			$\max_k t_{0k}$	CPU (s)	Gap (%)	$\max_k t_{0k}$	CPU (s)	Gap (%)	$\max_k t_{0k}$	CPU (s)
S3.1	60	4	455.4	59.1	10.4	**410.8**	129.3	–0.4	412.4	80.2
S3.2	70	5	394.6	78.8	5.6	**369.1**	127.5	–1.3	373.8	107.3
S3.3	80	6	461.8	94.7	9.5	433.5	114.3	2.7	**421.9**	124.3
S3.4	90	7	432.0	109.6	10.8	**389.0**	126.7	–0.2	389.8	145.9
S3.5	100	8	423.5	126.0	10.3	**383.6**	161.6	–0.1	384.0	171.3
S3.6	110	8	486.5	128.5	18.6	422.2	172.9	2.9	**410.2**	196.1
S3.7	120	9	429.3	151.5	9.1	395.9	200.0	0.6	**393.6**	210.6
S3.8	130	10	436.1	103.3	14.0	387.1	230.0	1.2	**382.5**	244.5
S3.9	140	11	460.4	119.9	18.3	405.8	228.0	4.3	**389.2**	267.8
S3.10	150	12	430.3	129.2	12.6	**381.8**	274.3	–0.1	382.1	292.1
S3.11	160	13	429.2	125.7	15.3	383.5	279.7	3.0	**372.4**	327.3
S3.12	170	14	400.0	143.3	12.9	358.1	321.1	1.1	**354.2**	339.3
S3.13	180	15	406.7	161.3	11.8	**361.7**	417.7	–0.6	363.8	352.4
S3.14	190	16	393.9	189.2	12.1	356.9	367.0	1.5	**351.5**	392.7
S3.15	200	17	403.8	174.5	9.9	**366.5**	516.4	–0.2	367.3	427.9
Total			*6443.5*		*12.1*	*5805.5*		*1.0*	***5748.7***	