Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **20XX**

Website: http://AIMsciences.org

pp. **X–XX**

OPTIMAL FLEET COMPOSITION VIA DYNAMIC PROGRAMMING AND GOLDEN SECTION SEARCH

Ryan Loxton

Department of Mathematics and Statistics
Curtin University
GPO Box U1987 Perth, Western Australia 6845

Qun Lin

Department of Mathematics and Statistics
Curtin University
GPO Box U1987 Perth, Western Australia 6845

(Communicated by the associate editor name)

Abstract. In this paper, we consider an optimization problem arising in vehicle fleet management. The problem is to construct a heterogeneous vehicle fleet in such a way that cost is minimized subject to a constraint on the overall fleet size. The cost function incorporates fixed and variable costs associated with the fleet, as well as hiring costs that are incurred when vehicle requirements exceed fleet capacity. We first consider the simple case when there is only one type of vehicle. We show that in this case the cost function is convex, and thus the problem can be solved efficiently using the well-known golden section method. We then devise an algorithm, based on dynamic programming and the golden section method, for solving the general problem in which there are multiple vehicle types. We conclude the paper with some simulation results.

1. **Introduction.**

Purchasing a vehicle fleet is one of the most expensive capital investments a company or organization can make. Accordingly, both the size and composition of such a fleet need to be carefully considered. Purchasing too few vehicles will result in excessive hiring costs, as additional vehicles will need to be hired whenever vehicle requirements exceed fleet capacity. On the other hand, purchasing too many vehicles will result in a massive opportunity cost.

Many optimization problems related to fleet composition have been discussed in the literature; see, for example, [3, 6, 7, 8, 11] and the references cited therein. A recent survey of optimization models combining fleet composition with vehicle routing is given in [5]. These combined models typically assume that the vehicle fleet is homogeneous—that is, the fleet contains only one type of vehicle. However, this assumption is unrealistic, as heterogenous fleets are usually preferred in practice because of their flexibility. In this paper, we consider a generalization of the fleet composition problem formulated in [4]. This problem does not incorporate vehicle routing, but it does allow for heterogeneous fleets consisting of different types of vehicles.

2000 *Mathematics Subject Classification.* Primary: 90B06, 90C25, 90C39; Secondary: 90C10.

Key words and phrases. Fleet Composition, Convex Optimization, Dynamic Programming, Golden Section Method.

We formulate the problem as follows. First, define the following quantities:

$$
\begin{aligned}
m &= \text{number of vehicle types.} \\
n &= \text{number of periods in the planning horizon.} \\
v_{it} &= \text{number of type-}i\text{ vehicles required during period } t \in \{1, \dots, n\}. \\
p_{\max} &= \text{maximum fleet size.} \\
\alpha_i &= \text{fixed cost per period of a type-}i\text{ vehicle.} \\
\beta_i &= \text{variable cost per period of a type-}i\text{ vehicle.} \\
\gamma_i &= \text{hiring cost per period of a type-}i\text{ vehicle.}
\end{aligned}
$$

A vehicle's fixed cost includes the initial cost of purchasing the vehicle, as well as other costs such as insurance premiums and registration fees. Variable costs are generally due to maintenance (servicing, replacing tires, etc.).

We assume that the cost of operating an owned vehicle for one period is less than the cost of hiring the same vehicle for one period—otherwise, there would be no reason to own a vehicle fleet. Hence,

$$\alpha_i + \beta_i < \gamma_i.$$

Let p_i denote the number of type-i vehicles in the fleet. Then

$$p_1 + \cdots + p_m \le p_{\max}.$$

Furthermore,

$$\text{Total fixed cost for type-}i\text{ vehicles} = n\alpha_i p_i.$$

Now, if $v_{it} > p_i$, then during period t all p_i of the fleet's type-i vehicles will be used, and an additional $v_{it} - p_i$ type-i vehicles will need to be hired to meet the vehicle requirements. On the other hand, when $v_{it} \le p_i$ only v_{it} of the fleet's type-i vehicles will be used, and no type-i vehicles will be hired. Thus,

$$\text{Number of (owned) type-}i\text{ vehicles used during period } t \;=\; \min\{v_{it}, p_i\}$$

and

$$\text{Number of type-}i\text{ vehicles hired during period } t \;=\; \max\{v_{it} - p_i, 0\}.$$

Consequently,

$$\text{Total variable cost for type-}i\text{ vehicles} \;=\; \beta_i \sum_{t=1}^{n} \min\{v_{it}, p_i\}$$

and

$$\text{Total hiring cost for type-}i\text{ vehicles} \;=\; \gamma_i \sum_{t=1}^{n} \max\{v_{it} - p_i, 0\}.$$

Therefore, the total cost associated with purchasing p_i vehicles of type i is:

$$C_i(p_i) = \underbrace{n\alpha_i p_i}_{\text{Fixed cost}} + \underbrace{\beta_i \sum_{t=1}^{n} \min\{v_{it}, p_i\}}_{\text{Variable cost}} + \underbrace{\gamma_i \sum_{t=1}^{n} \max\{v_{it} - p_i, 0\}}_{\text{Hiring cost}}. \tag{1}$$

The key question that now arises is: what values of p_i, $i = 1, \dots, m$ minimize the overall cost? We formulate this question mathematically as follows.

Problem P. Choose non-negative integers $p_1, \ldots, p_m$ to minimize the cost function

$$C(p_1, \ldots, p_m) = \sum_{i=1}^{m} C_i(p_i)$$

subject to the inequality constraint

$$\sum_{i=1}^{m} p_i \le p_{\max}.$$

When $m = 1$, Problem P is the same as the one-dimensional fleet composition problem described in reference [4]. According to [4], the optimal number of vehicles p^* satisfies the following equation:

$$|\mathcal{R}(p^*)| = \frac{n\alpha_1}{\gamma_1 - \beta_1}, \tag{2}$$

where $\mathcal{R}(p^*) = \{\, t : \ v_{1t} > p^* \,\}$ and $|\cdot|$ denotes set cardinality. Since $\mathcal{R}(p^*)$ is a discrete set, equation (2) is only valid when $n\alpha_1$ is divisible by $\gamma_1 - \beta_1$. This is often not the case. For example, if $n = 52$, $\alpha_1 = 50$, $\beta_1 = 40$, and $\gamma_1 = 100$, then equation (2) gives $|\mathcal{R}(p^*)| = 43.3333$, which is impossible. In this case, equation (2) cannot be used to solve Problem P. The purpose of this paper is to develop a new method for solving Problem P that does not rely on equation (2). Our new method is applicable to Problem P of any dimension, not just $m = 1$.

2. Preliminary Results.

Throughout this section, we assume that $i \in \{1, \ldots, m\}$ is arbitrary but fixed.

For each $p \in \mathbb{R}$, let

$$\mathcal{R}_i(p) \triangleq \{\, t : \ v_{it} > p \,\},$$
$$\mathcal{S}_i(p) \triangleq \{\, t : \ v_{it} < p \,\},$$
$$\mathcal{T}_i(p) \triangleq \{\, t : \ v_{it} = p \,\}.$$

It's clear that $\mathcal{R}_i(p)$, $\mathcal{S}_i(p)$, and $\mathcal{T}_i(p)$ constitute a partition of $\{1, \ldots, n\}$.

Recall that $C_i(p_i)$ is the total cost of purchasing p_i vehicles of type i. Although in this context p_i is an integer, the formula for $C_i(p_i)$ in equation (1) still makes sense when $p_i \in \mathbb{R} \setminus \mathbb{Z}$. Thus, we can extend the domain of C_i to $\mathbb{R}$. The following result then follows immediately.

Theorem 2.1. *The cost function C_i is continuous on $\mathbb{R}$.*

We now show that C_i is differentiable from the left.

Theorem 2.2. *The left derivative of C_i at $p \in \mathbb{R}$ is given by*

$$D_{-}C_i(p) = n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(p)| + (\beta_i - \gamma_i)|\mathcal{T}_i(p)|,$$

where $|\cdot|$ denotes set cardinality.

Proof. Define

$$\epsilon' \triangleq \begin{cases} \max\{\, v_{it} - p : \ t \in \mathcal{S}_i(p) \,\}, & \text{if } \mathcal{S}_i(p) \neq \emptyset, \\ -\infty, & \text{if } \mathcal{S}_i(p) = \emptyset. \end{cases}$$

Since $v_{it} < p$ for each $t \in \mathcal{S}_i(p)$, we have $\epsilon' < 0$.

Now, let $\epsilon \in (\epsilon', 0)$. Then

$$C_i(p+\epsilon) = n\alpha_i p + n\alpha_i\epsilon + \beta_i \sum_{t=1}^{n} \min\{v_{it}, p+\epsilon\} + \gamma_i \sum_{t=1}^{n} \max\{v_{it} - p - \epsilon, 0\}. \tag{3}$$

If $t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p)$, then $v_{it} \geq p > p + \epsilon$. Hence,

$$\min\{v_{it}, p+\epsilon\} = p+\epsilon, \qquad t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p), \tag{4}$$

and

$$\max\{v_{it} - p - \epsilon, 0\} = v_{it} - p - \epsilon, \qquad t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p). \tag{5}$$

On the other hand, if $t \in \mathcal{S}_i(p)$ then

$$p + \epsilon > p + \epsilon' \geq p + v_{it} - p = v_{it}.$$

Hence,

$$\min\{v_{it}, p+\epsilon\} = v_{it}, \qquad t \in \mathcal{S}_i(p), \tag{6}$$

and

$$\max\{v_{it} - p - \epsilon, 0\} = 0, \qquad t \in \mathcal{S}_i(p). \tag{7}$$

By equations (4) and (6),

$$\begin{aligned}
\sum_{t=1}^{n} \min\{v_{it}, p+\epsilon\} &= \sum_{t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p)} \min\{v_{it}, p+\epsilon\} + \sum_{t \in \mathcal{S}_i(p)} \min\{v_{it}, p+\epsilon\} \\
&= \sum_{t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p)} (p+\epsilon) + \sum_{t \in \mathcal{S}_i(p)} v_{it} \\
&= \epsilon|\mathcal{R}_i(p)| + \epsilon|\mathcal{T}_i(p)| + \sum_{t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p)} p + \sum_{t \in \mathcal{S}_i(p)} v_{it} \\
&= \epsilon|\mathcal{R}_i(p)| + \epsilon|\mathcal{T}_i(p)| + \sum_{t=1}^{n} \min\{v_{it}, p\}. \qquad (8)
\end{aligned}$$

Similarly, by equations (5) and (7),

$$\begin{aligned}
\sum_{t=1}^{n} \max\{v_{it} - p - \epsilon, 0\} &= \sum_{t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p)} \max\{v_{it} - p - \epsilon, 0\} \\
&\qquad + \sum_{t \in \mathcal{S}_i(p)} \max\{v_{it} - p - \epsilon, 0\} \\
&= \sum_{t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p)} (v_{it} - p - \epsilon) \\
&= -\epsilon|\mathcal{R}_i(p)| - \epsilon|\mathcal{T}_i(p)| + \sum_{t \in \mathcal{R}_i(p) \cup \mathcal{T}_i(p)} (v_{it} - p) \\
&= -\epsilon|\mathcal{R}_i(p)| - \epsilon|\mathcal{T}_i(p)| + \sum_{t=1}^{n} \max\{v_{it} - p, 0\}. \qquad (9)
\end{aligned}$$

Substituting equations (8) and (9) into equation (3) yields

$$\begin{aligned}
C_i(p+\epsilon) = n\alpha_i p + n\alpha_i\epsilon + \epsilon(\beta_i - \gamma_i)|\mathcal{R}_i(p)| + \epsilon(\beta_i - \gamma_i)|\mathcal{T}_i(p)| \\
+ \beta_i \sum_{t=1}^{n} \min\{v_{it}, p\} + \gamma_i \sum_{t=1}^{n} \max\{v_{it} - p, 0\}.
\end{aligned}$$

Thus,

$$C_i(p+\epsilon) = C_i(p) + n\alpha_i\epsilon + \epsilon(\beta_i - \gamma_i)|\mathcal{R}_i(p)| + \epsilon(\beta_i - \gamma_i)|\mathcal{T}_i(p)|.$$

Rearranging and then dividing both sides by ϵ gives

$$\frac{C_i(p+\epsilon) - C_i(p)}{\epsilon} = n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(p)| + (\beta_i - \gamma_i)|\mathcal{T}_i(p)|.$$

This equation holds for all $\epsilon \in (\epsilon', 0)$. By taking the limit as $\epsilon \to 0-$ we obtain

$$D_-C_i(p) = \lim_{\epsilon \to 0-} \frac{C_i(p+\epsilon) - C_i(p)}{\epsilon} = n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(p)| + (\beta_i - \gamma_i)|\mathcal{T}_i(p)|,$$

which completes the proof. □

We now show that C_i is also differentiable from the right.

Theorem 2.3. *The right derivative of C_i at $p \in \mathbb{R}$ is given by*

$$D_+C_i(p) = n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(p)|.$$

Proof. Define

$$\epsilon' \triangleq \begin{cases} \min\{\, v_{it} - p : \ t \in \mathcal{R}_i(p) \,\}, & \text{if } \mathcal{R}_i(p) \neq \emptyset, \\ +\infty, & \text{if } \mathcal{R}_i(p) = \emptyset. \end{cases}$$

Since $v_{it} > p$ for each $t \in \mathcal{R}_i(p)$, we have $\epsilon' > 0$.

Let $\epsilon \in (0, \epsilon')$. Then

$$C_i(p+\epsilon) = n\alpha_i p + n\alpha_i\epsilon + \beta_i \sum_{t=1}^{n} \min\{v_{it}, p+\epsilon\} + \gamma_i \sum_{t=1}^{n} \max\{v_{it} - p - \epsilon, 0\}. \quad (10)$$

If $t \in \mathcal{R}_i(p)$, then

$$p + \epsilon < p + \epsilon' \leq p + v_{it} - p = v_{it}.$$

Therefore,

$$\min\{v_{it}, p+\epsilon\} = p + \epsilon, \qquad t \in \mathcal{R}_i(p), \quad (11)$$

and

$$\max\{v_{it} - p - \epsilon, 0\} = v_{it} - p - \epsilon, \qquad t \in \mathcal{R}_i(p). \quad (12)$$

On the other hand, if $t \in \mathcal{S}_i(p) \cup \mathcal{T}_i(p)$ then $v_{it} \leq p < p + \epsilon$. Thus,

$$\min\{v_{it}, p+\epsilon\} = v_{it}, \qquad t \in \mathcal{S}_i(p) \cup \mathcal{T}_i(p), \quad (13)$$

and

$$\max\{v_{it} - p - \epsilon, 0\} = 0, \qquad t \in \mathcal{S}_i(p) \cup \mathcal{T}_i(p). \quad (14)$$

By equations (11) and (13),

$$\begin{aligned} \sum_{t=1}^{n} \min\{v_{it}, p+\epsilon\} &= \sum_{t \in \mathcal{R}_i(p)} \min\{v_{it}, p+\epsilon\} + \sum_{t \in \mathcal{S}_i(p) \cup \mathcal{T}_i(p)} \min\{v_{it}, p+\epsilon\} \\ &= \sum_{t \in \mathcal{R}_i(p)} (p+\epsilon) + \sum_{t \in \mathcal{S}_i(p) \cup \mathcal{T}_i(p)} v_{it} \\ &= \epsilon|\mathcal{R}_i(p)| + \sum_{t \in \mathcal{R}_i(p)} p + \sum_{t \in \mathcal{S}_i(p) \cup \mathcal{T}_i(p)} v_{it} \\ &= \epsilon|\mathcal{R}_i(p)| + \sum_{t=1}^{n} \min\{v_{it}, p\}. \end{aligned} \quad (15)$$

By equations (12) and (14),

$$\begin{aligned}\sum_{t=1}^{n}\max\{v_{it}-p-\epsilon,0\} &= \sum_{t\in\mathcal{R}_i(p)}\max\{v_{it}-p-\epsilon,0\}\\ &\qquad+\sum_{t\in\mathcal{S}_i(p)\cup\mathcal{T}_i(p)}\max\{v_{it}-p-\epsilon,0\}\\ &= \sum_{t\in\mathcal{R}_i(p)}(v_{it}-p-\epsilon)\\ &= -\epsilon|\mathcal{R}_i(p)|+\sum_{t\in\mathcal{R}_i(p)}(v_{it}-p)\\ &= -\epsilon|\mathcal{R}_i(p)|+\sum_{t=1}^{n}\max\{v_{it}-p,0\}. \qquad (16)\end{aligned}$$

Substituting equations (15) and (16) into equation (10) gives

$$\begin{aligned}C_i(p+\epsilon) &= n\alpha_i p+n\alpha_i\epsilon+\epsilon(\beta_i-\gamma_i)|\mathcal{R}_i(p)|\\ &\qquad+\beta_i\sum_{t=1}^{n}\min\{v_{it},p\}+\gamma_i\sum_{t=1}^{n}\max\{v_{it}-p,0\}.\end{aligned}$$

Therefore,

$$C_i(p+\epsilon)-C_i(p)=n\alpha_i\epsilon+\epsilon(\beta_i-\gamma_i)|\mathcal{R}_i(p)|.$$

Dividing both sides by ϵ gives

$$\frac{C_i(p+\epsilon)-C_i(p)}{\epsilon}=n\alpha_i+(\beta_i-\gamma_i)|\mathcal{R}_i(p)|.$$

Hence,

$$D_+C_i(p)=\lim_{\epsilon\to 0+}\frac{C_i(p+\epsilon)-C_i(p)}{\epsilon}=n\alpha_i+(\beta_i-\gamma_i)|\mathcal{R}_i(p)|,$$

as required. □

Now, by Theorems 2.2 and 2.3,

$$D_+C_i(p)-D_-C_i(p)=-(\beta_i-\gamma_i)|\mathcal{T}_i(p)|.$$

Hence, since $\beta_i-\gamma_i<0$ (recall that $\alpha_i+\beta_i<\gamma_i$), the left and right derivatives of C_i differ when $\mathcal{T}_i(p)\neq\emptyset$. It follows that C_i is not differentiable at points $p=v_{it}$, $t=1,\ldots,n$.

The following result is proved in Chapter 5 of [10].

Lemma 2.4. *Let $f:\mathbb{R}\to\mathbb{R}$ be a continuous function with right derivative D_+f. If D_+f is non-decreasing, then f is convex.*

We now use Lemma 2.4 to prove that C_i is a convex function.

Theorem 2.5. *The cost function C_i is convex.*

Proof. We already know that C_i is continuous and right differentiable (Theorems 2.1 and 2.3). We will now prove that D_+C_i is non-decreasing (the convexity of C_i then follows immediately from Lemma 2.4).

Let $x<y$. Then obviously

$$\mathcal{R}_i(y)\subset\mathcal{R}_i(x).$$

Thus,

$$|\mathcal{R}_i(y)| \leq |\mathcal{R}_i(x)|.$$

Since $\beta_i - \gamma_i < 0$,

$$(\beta_i - \gamma_i)|\mathcal{R}_i(y)| \geq (\beta_i - \gamma_i)|\mathcal{R}_i(x)|.$$

Combining this inequality with the formula in Theorem 2.3 gives

$$D_+C_i(x) = n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(x)| \leq n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(y)| = D_+C_i(y).$$

Since x and y were chosen arbitrarily, this argument shows that D_+C_i is non-decreasing. □

3. **Special Case: One Vehicle Type.** Suppose that we only consider vehicles of one type (say type i). Then Problem P takes the following form.

Problem $\mathbf{P}_i$. Choose an integer $p_i \in \{0, \ldots, p_{\max}\}$ to minimize the cost function

$$C_i(p_i) = n\alpha_i p_i + \beta_i \sum_{t=1}^{n} \min\{v_{it}, p_i\} + \gamma_i \sum_{t=1}^{n} \max\{v_{it} - p_i, 0\}.$$

By discarding the integer constraint on p_i, we obtain the following *continuous relaxation* of Problem P_i.

Problem $\tilde{\mathbf{P}}_i$. Choose a real number $p_i \in [0, p_{\max}]$ to minimize the cost function

$$C_i(p_i) = n\alpha_i p_i + \beta_i \sum_{t=1}^{n} \min\{v_{it}, p_i\} + \gamma_i \sum_{t=1}^{n} \max\{v_{it} - p_i, 0\}.$$

Since C_i is continuous and $[0, p_{\max}]$ is compact, Problem $\tilde{\mathrm{P}}_i$ always has a solution. In fact, it turns out that Problem $\tilde{\mathrm{P}}_i$ always has an *integer* solution. This suggests that the integer constraint in Problem P_i is actually redundant.

To prove that Problem $\tilde{\mathrm{P}}_i$ has an integer solution, we need the following result.

Lemma 3.1. *Let p^* be a solution of Problem $\tilde{P}_i$. If $p^* \notin \mathbb{Z}$, then*

$$|\mathcal{R}_i(p^*)| = \frac{n\alpha_i}{\gamma_i - \beta_i}.$$

Proof. Since $p^* \notin \mathbb{Z}$, we must have $\mathcal{T}_i(p^*) = \emptyset$. Hence, by Theorems 2.2 and 2.3,

$$\begin{aligned} D_-C_i(p^*) &= n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(p^*)| + (\beta_i - \gamma_i)|\mathcal{T}_i(p^*)| \\ &= n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(p^*)| \\ &= D_+C_i(p^*). \end{aligned}$$

This shows that C_i is differentiable at $p = p^*$. Thus, since p^* is a minimal point,

$$\frac{dC_i(p^*)}{dp} = n\alpha_i + (\beta_i - \gamma_i)|\mathcal{R}_i(p^*)| = 0.$$

Rearranging this equation gives

$$|\mathcal{R}_i(p^*)| = \frac{n\alpha_i}{\gamma_i - \beta_i},$$

which completes the proof. □

We now show that Problem $\tilde{\mathrm{P}}_i$ always has at least one integer solution.

Theorem 3.2. *If p^* is a solution of Problem $\tilde{P}_i$, then $\lfloor p^* \rfloor$ and $\lceil p^* \rceil$ are also solutions of Problem $\tilde{P}_i$.*

Proof. If $p^* \in \mathbb{Z}$, then $p^* = \lfloor p^* \rfloor = \lceil p^* \rceil$ and the result is immediate. Thus, assume that $p^* \notin \mathbb{Z}$. Then by Lemma 3.1,

$$|\mathcal{R}_i(p^*)| = \frac{n\alpha_i}{\gamma_i - \beta_i}. \tag{17}$$

Now,

$$\begin{aligned} C_i(\lfloor p^* \rfloor) &= n\alpha_i \lfloor p^* \rfloor + \beta_i \sum_{t=1}^{n} \min\{v_{it}, \lfloor p^* \rfloor\} + \gamma_i \sum_{t=1}^{n} \max\{v_{it} - \lfloor p^* \rfloor, 0\} \\ &= n\alpha_i \lfloor p^* \rfloor + \beta_i \sum_{t \in \mathcal{R}_i(\lfloor p^* \rfloor)} \lfloor p^* \rfloor + \beta_i \sum_{t \in \mathcal{S}_i(\lfloor p^* \rfloor) \cup \mathcal{T}_i(\lfloor p^* \rfloor)} v_{it} \\ &\qquad + \gamma_i \sum_{t \in \mathcal{R}_i(\lfloor p^* \rfloor)} (v_{it} - \lfloor p^* \rfloor). \end{aligned}$$

Clearly,

$$\mathcal{R}_i(\lfloor p^* \rfloor) = \mathcal{R}_i(p^*)$$

and

$$\mathcal{S}_i(\lfloor p^* \rfloor) \cup \mathcal{T}_i(\lfloor p^* \rfloor) = \mathcal{S}_i(p^*).$$

Hence,

$$\begin{aligned} C_i(\lfloor p^* \rfloor) &= n\alpha_i \lfloor p^* \rfloor + \beta_i \cdot \lfloor p^* \rfloor \cdot |\mathcal{R}_i(p^*)| + \beta_i \sum_{t \in \mathcal{S}_i(p^*)} v_{it} + \gamma_i \sum_{t \in \mathcal{R}_i(p^*)} (v_{it} - \lfloor p^* \rfloor) \\ &= n\alpha_i p^* - n\alpha_i (p^* - \lfloor p^* \rfloor) + \beta_i p^* |\mathcal{R}_i(p^*)| - \beta_i \cdot (p^* - \lfloor p^* \rfloor) \cdot |\mathcal{R}_i(p^*)| \\ &\quad + \beta_i \sum_{t \in \mathcal{S}_i(p^*)} v_{it} + \gamma_i \sum_{t \in \mathcal{R}_i(p^*)} (v_{it} - p^*) + \gamma_i \cdot (p^* - \lfloor p^* \rfloor) \cdot |\mathcal{R}_i(p^*)|. \end{aligned} \tag{18}$$

Since $p^* \notin \mathbb{Z}$, we have $\mathcal{T}_i(p^*) = \emptyset$. Thus,

$$\begin{aligned} C_i(p^*) &= n\alpha_i p^* + \beta_i \sum_{t=1}^{n} \min\{v_{it}, p^*\} + \gamma_i \sum_{t=1}^{n} \max\{v_{it} - p^*, 0\} \\ &= n\alpha_i p^* + \beta_i \sum_{t \in \mathcal{R}_i(p^*)} \min\{v_{it}, p^*\} + \beta_i \sum_{t \in \mathcal{S}_i(p^*)} \min\{v_{it}, p^*\} \\ &\qquad + \gamma_i \sum_{t \in \mathcal{R}_i(p^*)} \max\{v_{it} - p^*, 0\} + \gamma_i \sum_{t \in \mathcal{S}_i(p^*)} \max\{v_{it} - p^*, 0\} \\ &= n\alpha_i p^* + \beta_i p^* |\mathcal{R}_i(p^*)| + \beta_i \sum_{t \in \mathcal{S}_i(p^*)} v_{it} + \gamma_i \sum_{t \in \mathcal{R}_i(p^*)} (v_{it} - p^*). \end{aligned} \tag{19}$$

Substituting (19) into (18) yields

$$C_i(\lfloor p^* \rfloor) = C_i(p^*) - n\alpha_i (p^* - \lfloor p^* \rfloor) + (\gamma_i - \beta_i) \cdot (p^* - \lfloor p^* \rfloor) \cdot |\mathcal{R}_i(p^*)|.$$

Thus, by using (17) we obtain

$$C_i(\lfloor p^* \rfloor) = C_i(p^*),$$

which shows that $\lfloor p^* \rfloor$ is optimal for Problem $\tilde{\mathrm{P}}_i$. A similar proof shows that $\lceil p^* \rceil$ is also optimal. □

Obviously, any integer solution of Problem $\tilde{\mathrm{P}}_i$ is also a solution of Problem P_i. Thus, it follows from Theorem 3.2 that if p^* is a solution of Problem $\tilde{\mathrm{P}}_i$, then both $\lfloor p^* \rfloor$ and $\lceil p^* \rceil$ are solutions of Problem P_i.

Problem $\tilde{\mathrm{P}}_i$ is just a one-dimensional convex optimization problem that can be solved efficiently using the golden section method (see [1, 9]). The golden section method works by successively reducing the *interval of uncertainty*—a known interval that is guaranteed to contain at least one optimal solution. This is done by evaluating the cost function at certain *test points* and then exploiting convexity.

The initial interval of uncertainty for Problem $\tilde{\mathrm{P}}_i$ is

$$\mathcal{I}_0 = [a_0, b_0] = [0, p_{\max}].$$

The initial test points q_1^1 and q_2^1 are

$$q_1^1 = b_0 - r(b_0 - a_0) = p_{\max} - rp_{\max}$$

and

$$q_2^1 = a_0 + r(b_0 - a_0) = rp_{\max},$$

where

$$r = \frac{\sqrt{5}-1}{2} \approx 0.618.$$

Suppose that we are given the $(k-1)$th interval of uncertainty $\mathcal{I}_{k-1} = [a_{k-1}, b_{k-1}]$ and the corresponding test points q_1^k and q_2^k. If $C_i(q_1^k) < C_i(q_2^k)$, then the optimal solution must lie in $[a_{k-1}, q_2^k]$ because C_i is a convex function. Hence, the new interval of uncertainty is

$$\mathcal{I}_k = [a_k, b_k] = [a_{k-1}, q_2^k].$$

On the other hand, if $C_i(q_1^k) \geq C_i(q_2^k)$, then the optimal solution must lie in the interval $[q_1^k, b_{k-1}]$. Hence, the new interval of uncertainty is

$$\mathcal{I}_k = [a_k, b_k] = [q_1^k, b_{k-1}].$$

The new test points are

$$q_1^{k+1} = \begin{cases} q_2^k - r|\mathcal{I}_k|, & \text{if } \mathcal{I}_k = [a_{k-1}, q_2^k], \\ q_2^k, & \text{if } \mathcal{I}_k = [q_1^k, b_{k-1}], \end{cases}$$

and

$$q_2^{k+1} = \begin{cases} q_1^k, & \text{if } \mathcal{I}_k = [a_{k-1}, q_2^k], \\ q_1^k + r|\mathcal{I}_k|, & \text{if } \mathcal{I}_k = [q_1^k, b_{k-1}]. \end{cases}$$

Since one of the new test points coincides with an old test point, each subsequent iteration of the golden section method requires only one extra cost function evaluation. Furthermore,

$$|\mathcal{I}_k| = r|\mathcal{I}_{k-1}| < |\mathcal{I}_{k-1}|.$$

Consequently,

$$|\mathcal{I}_k| = b_k - a_k = r^k p_{\max}.$$

We now prove a key result.

Theorem 3.3. *Let N be an integer such that*

$$N > -\frac{\ln p_{\max}}{\ln r}. \tag{20}$$

Suppose that the golden section method is applied to Problem $\tilde{P}_i$ for N iterations, and let $\mathcal{I}_N = [a_N, b_N]$ denote the final interval of uncertainty. Then $\lceil a_N \rceil$ is a solution of both Problem $\tilde{P}_i$ and Problem P_i.

Proof. From (20), we obtain

$$r^N < \frac{1}{p_{\max}}.$$

Thus,

$$b_N - a_N = r^N p_{\max} < 1. \tag{21}$$

Now, the interval of uncertainty $\mathcal{I}_N$ must contain at least one solution of Problem $\tilde{\mathrm{P}}_i$. Let p^* denote such a solution. We know from Theorem 3.2 that both $\lfloor p^* \rfloor$ and $\lceil p^* \rceil$ are integer solutions of Problem $\tilde{\mathrm{P}}_i$. Thus, to complete the proof, it is sufficient to show that either $\lceil a_N \rceil = \lfloor p^* \rfloor$ or $\lceil a_N \rceil = \lceil p^* \rceil$.

Since $p^* \in \mathcal{I}_N$, we have

$$\lceil a_N \rceil \leq \lceil p^* \rceil.$$

Thus, either $\lceil a_N \rceil = \lceil p^* \rceil$ or $\lceil a_N \rceil < \lceil p^* \rceil$. If $\lceil a_N \rceil = \lceil p^* \rceil$, then the proof is complete. Therefore, we assume that $\lceil a_N \rceil < \lceil p^* \rceil$. By (21),

$$p^* \leq b_N < a_N + 1 \leq \lceil a_N \rceil + 1, \tag{22}$$

which implies that

$$\lceil a_N \rceil < \lceil p^* \rceil \leq \lceil a_N \rceil + 1.$$

Hence, we must have

$$\lceil p^* \rceil = \lceil a_N \rceil + 1. \tag{23}$$

Now, if p^* is an integer, then

$$p^* = \lceil p^* \rceil = \lceil a_N \rceil + 1,$$

which contradicts (22). Thus, p^* cannot be an integer, and so

$$\lceil p^* \rceil = \lfloor p^* \rfloor + 1. \tag{24}$$

Combining (23) and (24) yields $\lceil a_N \rceil = \lfloor p^* \rfloor$. □

By virtue of Theorem 3.3, we can solve Problem P_i using the following simple algorithm.

Algorithm 3.1. (Finds a solution p_i^* of Problem P_i)

1. Compute
$$N = \left\lceil -\frac{\ln p_{\max}}{\ln r} \right\rceil. \tag{25}$$
2. Apply the golden section method to Problem $\tilde{\mathrm{P}}_i$ for N iterations. Let $[a_N, b_N]$ denote the final interval of uncertainty.
3. Stop: $p_i^* = \lceil a_N \rceil$ is a solution of Problem P_i.

Note that Algorithm 3.1 performs $N+1$ cost function evaluations—a small number even when $p_{\max}$ is extremely large (see Table 1).

4. **Solving Problem P: A Dynamic Programming Approach.**

In Section 3 we considered Problem P for $m = 1$ (only one vehicle type). We now focus on the general case when $m > 1$ (multiple vehicle types).

For each $k = 1, \ldots, m$ and $\theta = 0, ..., p_{\max}$, consider the following subproblem of Problem P.

$p_{\max}$	$N+1$
10	6
10^2	11
10^3	16
10^4	21
10^5	25
10^6	30

Table 1. Algorithm 3.1 performs $N+1$ cost function evaluations, where N is given by equation (25).

Problem $\mathrm{Q}_k(\theta)$. Choose non-negative integers $p_1, \ldots, p_k$ to minimize

$$\sum_{i=1}^{k} C_i(p_i)$$

subject to the inequality constraint

$$\sum_{i=1}^{k} p_i \le \theta.$$

Let $f_k(\theta)$ denote the optimal cost of Problem $\mathrm{Q}_k(\theta)$. Furthermore, let $\hat{p}_1^*$ denote a solution of Problem P_1 (which can be computed efficiently using Algorithm 3.1). We have the following result.

Theorem 4.1. *For each $\theta = 0, \ldots, p_{\max}$, the value of $f_1(\theta)$ is given by*

$$f_1(\theta) = \begin{cases} C_1(\theta), & \text{if } \theta < \hat{p}_1^*, \\ C_1(\hat{p}_1^*), & \text{if } \theta \ge \hat{p}_1^*. \end{cases}$$

Proof. Since $\hat{p}_1^*$ is a solution of Problem P_1,

$$C_1(\hat{p}_1^*) \le C_1(p_1), \qquad p_1 = 0, \ldots, p_{\max}. \tag{26}$$

Suppose that $\theta \ge \hat{p}_1^*$. Then $\hat{p}_1^* \in \{0, \ldots, \theta\}$ and thus it follows from (26) that

$$f_1(\theta) = \min_{p_1 \in \{0, \ldots, \theta\}} C_1(p_1) = C_1(\hat{p}_1^*),$$

as required.

Now, suppose that $\theta < \hat{p}_1^*$. If $p_1 = \theta$ is not optimal for Problem $\mathrm{Q}_1(\theta)$, then there exists an integer $p' \in \{0, \ldots, \theta\}$ such that

$$C_1(p') < C_1(\theta). \tag{27}$$

Clearly, $\theta \in (p', \hat{p}_1^*)$. Hence, there exists a $\lambda \in (0,1)$ such that

$$\theta = \lambda p' + (1-\lambda)\hat{p}_1^*.$$

Since C_1 is convex,

$$C_1(\theta) = C_1(\lambda p' + (1-\lambda)\hat{p}_1^*) \le \lambda C_1(p') + (1-\lambda)C_1(\hat{p}_1^*).$$

Thus, from (26) and (27),

$$C_1(\theta) < \lambda C_1(\theta) + (1-\lambda)C_1(\theta) = C_1(\theta),$$

which is an obvious contradiction. Hence, θ must be optimal for Problem $\mathrm{Q}_1(\theta)$. That is,

$$f_1(\theta) = \min_{p_1 \in \{0,\ldots,\theta\}} C_1(p_1) = C_1(\theta),$$

as required. □

Theorem 4.1 gives a simple formula for computing $f_1(\theta)$. We will now use the *principle of optimality*, also called the *principle of dynamic programming* (see [2]), to derive a recurrence equation for computing the remaining f_k. The principle of optimality states that if $p_1^*, \ldots, p_k^*$ are optimal for Problem $\mathrm{Q}_k(\theta)$, then $p_1^*, \ldots, p_{k-1}^*$ are optimal for Problem $\mathrm{Q}_{k-1}(\theta - p_k^*)$. Mathematically,

$$\begin{aligned} f_k(\theta) &= \min_{\substack{p_1,\ldots,p_k \in \mathbb{Z}^+ \cup \{0\} \\ p_1+\cdots+p_k \le \theta}} \left\{ \sum_{i=1}^{k} C_i(p_i) \right\} \\ &= \min_{p_k \in \{0,\ldots,\theta\}} \min_{\substack{p_1,\ldots,p_{k-1} \in \mathbb{Z}^+ \cup \{0\} \\ p_1+\cdots+p_{k-1} \le \theta - p_k}} \left\{ C_k(p_k) + \sum_{i=1}^{k-1} C_i(p_i) \right\} \\ &= \min_{p_k \in \{0,\ldots,\theta\}} \left\{ C_k(p_k) + \min_{\substack{p_1,\ldots,p_{k-1} \in \mathbb{Z}^+ \cup \{0\} \\ p_1+\cdots+p_{k-1} \le \theta - p_k}} \sum_{i=1}^{k-1} C_i(p_i) \right\} \\ &= \min_{p_k \in \{0,\ldots,\theta\}} \big\{ C_k(p_k) + f_{k-1}(\theta - p_k) \big\}, \end{aligned} \tag{28}$$

where $f_0 \triangleq 0$. Equation (28) and the formula in Theorem 4.1 define a recurrence relationship for f_k, $k \ge 1$.

Now, let $z_k(\theta)$ denote the element of $\{0, \ldots, \theta\}$ that achieves the minimum in equation (28). That is,

$$z_k(\theta) = \operatorname*{arg\,min}_{p_k \in \{0,\ldots,\theta\}} \big\{ C_k(p_k) + f_{k-1}(\theta - p_k) \big\}.$$

By Theorem 4.1,

$$z_1(\theta) = \begin{cases} \theta, & \text{if } \theta < \hat{p}_1^*, \\ \hat{p}_1^*, & \text{if } \theta \ge \hat{p}_1^*. \end{cases}$$

Problem $\mathrm{Q}_m(p_{\max})$ is the same as Problem P. Thus, $f_m(p_{\max})$ is the optimal cost of Problem P, and $p_m^* = z_m(p_{\max})$ is the optimal number of type-m vehicles in Problem P. It then follows from the principle of optimality that the optimal number of type-$(m-1)$ vehicles is $p_{m-1}^* = z_{m-1}(p_{\max} - p_m^*)$. Similarly, the optimal number of type-$(m-2)$ vehicles is $p_{m-2}^* = z_{m-2}(p_{\max} - p_m^* - p_{m-1}^*)$, and so on. The following algorithm for solving Problem P is based on this dynamic programming approach.

Algorithm 4.1. (Finds a solution $p_1^*, \ldots, p_m^*$ of Problem P)

1. Use Algorithm 3.1 to solve Problem P_1. Let $\hat{p}_1^*$ denote the solution obtained.
2. If $m = 1$, then stop: $p_1^* = \hat{p}_1^*$ is the solution of Problem P. Otherwise, go to Step 3.
3. Compute $C_1(\hat{p}_1^*)$.
4. Use the formula in Theorem 4.1 to compute $f_1(\theta)$, $\theta = 0, \ldots, p_{\max}$.
5. For each $k = 2, \ldots, m-1$, compute

$$f_k(\theta) = \min_{p_k \in \{0,\ldots,\theta\}} \big\{ C_k(p_k) + f_{k-1}(\theta - p_k) \big\}, \qquad \theta = 0, \ldots, p_{\max}.$$

   Let $z_k(\theta)$ denote the p_k that achieves the minimum in this equation.

6. Compute
$$f_m(p_{\max}) = \min_{p_m \in \{0,\ldots,p_{\max}\}} \big\{C_m(p_m) + f_{m-1}(p_{\max} - p_m)\big\}.$$
Let $z_m(p_{\max})$ denote the p_m that achieves the minimum in this equation.
7. For each $k = m, \ldots, 1$, compute
$$\theta_k = p_{\max} - \sum_{j=k+1}^{m} p_j^*$$
and
$$p_k^* = z_k(\theta_k).$$
8. Stop: $p_1^*, \ldots, p_m^*$ is a solution of Problem P.

Note that Step 1 and Steps 3-6 of Algorithm 4.1 involve evaluating C_i. The cost function evaluations[1] required in these steps are:

- $N + 1$ cost function evaluations in Step 1 (see Section 3).
- 1 cost function evaluation in Step 3.
- $\hat{p}_1^* \leq p_{\max}$ cost function evaluations in Step 4.
- Less than $(m-2) \times (p_{\max}+1)^2$ cost function evaluations in Step 5.
- $p_{\max} + 1$ cost function evaluations in Step 6.

Thus, an upper bound for the total number of cost function evaluations performed by Algorithm 4.1 is:
$$\hat{N} \triangleq N + 1 + \varsigma_m \big[2p_{\max} + (m-2)(p_{\max}+1)^2 + 2\big], \tag{29}$$
where N is defined by equation (25) and
$$\varsigma_m = \begin{cases} 1, & \text{if } m \geq 2, \\ 0, & \text{if } m = 1. \end{cases}$$
We will see in the next section that $\hat{N}$ is much smaller than the total number of feasible points in Problem P. Thus, Algorithm 4.1 is very efficient.

5. **Numerical Results.**

For numerical testing, we wrote a Fortran program that uses Algorithm 4.1 to solve random instances of Problem P. This program verifies the solution from Algorithm 4.1 by *complete enumeration*—that is, by evaluating the cost function C at every feasible point.

The number of random problems solved during each run of the program is selected by the user. During each run, the problem dimensions m, n, and $p_{\max}$ are fixed, but the other parameters are chosen randomly to generate the different instances of Problem P. The random parameters are selected from the following sets:

- $\alpha_i \in [0, 50]$
- $\beta_i \in [0, 50]$
- $\gamma_i \in (\alpha_i + \beta_i, 2\alpha_i + 2\beta_i]$[2]
- $v_{it} \in \{0, \ldots, p_{\max}\}$

[1] "Cost function evaluation" refers to a single evaluation of C_i.
[2] Recall that α_i, β_i, and γ_i must satisfy $\alpha_i + \beta_i < \gamma_i$.

Dimensions						
m	n	$p_{\max}$	AC	$\hat{N}$	FP	ρ
1	50	300	13	13	301	0.0432
1	100	500	14	14	501	0.0279
2	20	50	86	112	1,326	0.0324
2	50	100	161	213	5,151	0.0156
3	30	80	3,453	6,734	91,881	0.0125
3	50	100	5,313	10,414	176,851	0.0100
4	50	30	1,048	1,993	46,376	0.0056
4	50	50	2,739	5,314	316,251	0.0022
5	60	20	733	1,373	53,130	0.0028
5	100	30	1,543	2,954	324,632	0.0010

TABLE 2. Results from using Algorithm 4.1 to solve 1000 random instances of Problem P.

Note that the inequality constraint in Problem P doesn't depend on α_i, β_i, γ_i, or v_{it} (it only depends on $p_{\max}$). Thus, the random problems generated by our program all have the same number of feasible points.

To test Algorithm 4.1, we ran the program with 10 different sets of problem dimensions. During each run, the program generated and solved 1000 random instances of Problem P. Each random problem was solved twice: once using Algorithm 4.1 and once using complete enumeration. For each problem, the solution obtained by Algorithm 4.1 agreed with the solution obtained by complete enumeration.

Table 2 provides a summary of our numerical results. The notation in this table is explained below:

- AC is the average number of cost function evaluations used by Algorithm 4.1 (rounded up to the nearest integer).
- $\hat{N}$ is an upper bound for the number of cost function evaluations used by Algorithm 4.1 (see equation (29)).
- FP is the number of feasible points in each random problem (recall that the problem dimensions are fixed during each run of the program).

As expected, AC is always less than or equal to $\hat{N}$ in Table 2.

In the last column of Table 2, we give the ratio ρ of the number of cost evaluations used by Algorithm 4.1 to the number of cost evaluations used by complete enumeration. This ratio is calculated according to the following formula:

$$\rho = \frac{AC}{m \times FP}.$$

The values of ρ in Table 2 are very small; the highest is 0.0432. This suggests that Algorithm 4.1 needs significantly fewer cost function evaluations than complete enumeration. Furthermore, Table 2 shows that ρ decreases as m increases, so it seems that Algorithm 4.1 becomes more efficient as the problem dimensions increase.

Dimensions					
m	n	$p_{\max}$	AC	$\hat{N}$	Time (secs)
10	20	20	1,887	3,578	0.1818
10	50	50	10,694	20,920	2.6277
20	20	20	4,197	7,988	0.4058
20	50	50	23,954	46,930	5.8588
30	20	20	6,507	12,398	0.5936
30	50	50	37,213	72,940	9.1121
40	20	20	8,818	16,808	0.8042
40	50	50	50,473	98,950	12.3617
50	20	20	11,127	21,218	1.0093
50	50	50	63,734	124,960	15.5507

TABLE 3. Time taken for Algorithm 4.1 to solve 1000 random instances of Problem P.

Since the problem dimensions in Table 2 are reasonably small, the solution obtained by Algorithm 4.1 could be easily verified by complete enumeration. This is not possible, however, for problems with large dimensions. To test Algorithm 4.1's performance on large-scale problems, we ran our program with 10 sets of "large" problem dimensions, each of which has $m \geq 10$. As before, the program generated and solved 1000 random instances of Problem P, but this time we did not use the program to verify the solution by complete enumeration. Table 3 records the program's running times on a MacBook Pro (2.66GHz Intel Core i7 processor). The efficiency of Algorithm 4.1 is clearly evident here: solving 1000 random problems with $m = 50$ takes only 15 seconds. Solving just one of these problems using complete enumeration would take hours, if not days.

Acknowledgements. We would like to thank the anonymous reviewers for their kind comments.

REFERENCES

[1] M. S. Bazaraa, H. D. Sherali and C. M. Shetty, "Nonlinear Programming: Theory and Algorithms," 3^{rd} edition, John Wiley, New Jersey, 2006.
[2] R. Bellman, "Dynamic Programming," Dover Publications, New York, 2003.
[3] J. Couillard, *A decision support system for vehicle fleet planning*, Decision Support Systems, **9** (1993), 149-159.
[4] G. Ghiani, G. Laporte and R. Musmanno, "Introduction to Logistics Systems Planning and Control," John Wiley, Chichester, 2004.
[5] A. Hoff, H. Andersson, M. Christiansen, G. Hasle and A. Løkketangen, *Industrial aspects and literature survey: Fleet composition and routing*, Computers & Operations Research, **37** (2010), 2041-2061.
[6] A. Imai and F. Rivera, *Strategic fleet size planning for maritime refrigerated containers*, Maritime Policy & Management, **28** (2001), 361-374.
[7] D. Kirby, *Is your fleet the right size?*, Operational Research Quarterly, **10** (1959), 252.
[8] M. Y. Lai, C. S. Liu and X. J. Tong, *A two-stage hybrid meta-heuristic for pickup and delivery vehicle routing problem with time windows*, Journal of Industrial & Management Optimization, **6** (2010), 435-451.
[9] D. G. Luenberger and Y. Ye, "Linear and Nonlinear Programming," 3^{rd} edition, Springer, New York, 2008.

[10] H. L. Royden, “Real Analysis,” 3^{rd} edition, Prentice Hall, New Jersey, 1988.
[11] I. F. A. Vis, R. B. M. de Koster and M. W. P. Savelsbergh, *Minimum vehicle fleet size under time-window constraints at a container terminal*, Transportation Science, **39** (2005), 249-260.

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: `R.Loxton@curtin.edu.au`
E-mail address: `Q.Lin@curtin.edu.au`