Simple Policies for Capacitated Resupply Problems

Mette Wagenvoort
Econometric Institute, Erasmus University Rotterdam, The Netherlands

Martijn van Ee
Faculty of Military Sciences, Netherlands Defence Academy, Den Helder, The Netherlands

Paul Bouman
Econometric Institute, Erasmus University Rotterdam, The Netherlands

Kerry M. Malone
Military Operations, TNO, The Hague, The Netherlands

Abstract

We consider the Capacitated Resupply Problem in which locations with a given demand rate should be resupplied by vehicles such that they do not run out of stock and the number of vehicles is minimised. Compared to related problems, we consider the scenario where the payload of the vehicles may not suffice to bring the stock level back to full capacity. We focus on the Homogeneous Capacitated Resupply Problem and present both simple policies that provide 2-approximations and an optimal greedy policy that runs in pseudo-polynomial time.

2012 ACM Subject Classification Applied computing → Transportation

Keywords and phrases resupply problems, periodic schedules, approximation guarantee, greedy policy

Digital Object Identifier 10.4230/OASIcs.ATMOS.2023.18

Category Short Paper

Funding This research was made possible by TNO in collaboration with Erasmus University Rotterdam and the Netherlands Defence Academy.

1 Introduction

There are numerous applications, such as disaster relief and expeditions in harsh environments, where people operating at different locations need periodic resupply of commodities such as food, fuel and medicines. Such resupplies can typically be performed using faster motorized (off-terrain) vehicles. Recent advances in drone technology make fast resupply at remote locations increasingly possible. In many of the mentioned applications, it is vital that the stock of the commodities never drops below a critical level for a sustained period of time. An interesting tactical question is how many vehicles are needed to sustain all the stocks of the commodities above the critical level.

We consider a periodic resupply problem for a single commodity. In this problem we have a set of locations with associated capacities and demand rates. The goal is to determine whether the stock levels can be indefinitely sustained above a critical threshold by a set of vehicles that have an associated maximum payload. Several variants of the problem can be considered. The locations and vehicles can either be homogeneous or heterogeneous and each vehicle may or may not have a sufficient payload to fully restock locations to maximum capacity. Hence, partial restocking or multi-vehicle convoys can be considered.

Our problem, which we formally define in the next section, has a relation with periodic scheduling problems, such as the Pinwheel Scheduling problem [4], the Windows Scheduling problem [2], and the Periodic Latency problem [3]. In each of these problems, we are given integers p_i, and we have to schedule job i at least once in any period of p_i consecutive time-units. In the Windows Scheduling problem, we seek the minimum number of vehicles that are needed for a feasible solution, whereas the other two problems are concerned with feasibility only. From our problem's perspective, they all implicitly assume that each time a visit to a location is scheduled, its inventory is restocked to its full capacity. In our problem, it is possible that a vehicle's payload is not sufficient to restock a location to its full capacity, introducing interplay between payload and capacity. Also, the UAV resupply scheduling problem considered in [1] has some similarities with our model, but again locations are restocked (recharged) to their full capacity at each visit. We leave a full comparison of our model with the related literature for the full version of this paper.

In this article, we discuss the general problem and show it is intractable. We then focus on the homogeneous variant and study several simple policies that we prove to give 2-approximations. Finally, we present a greedy policy that is able to find a feasible schedule for the optimal number of vehicles and explain how the optimal number of vehicles can be found in pseudo-polynomial time.

2 Problem Description and Complexity

In this article we consider a single commodity problem as there is often a commodity that is most important in terms of size and weight. We consider discrete time-units, and at time-unit zero, each location starts with initial stock at maximum capacity. We then apply the following procedure: each time-unit each vehicle resupplies at most one location by a round-trip from a depot. If a location is resupplied, the payload from the vehicle is added to the stock at the location up to its capacity within the time-unit. At the start of a time-unit, consumption decreases the stock level of locations by the demand rate. If the stock drops below 0 after consumption, we call this a *stock-out*, a situation we want to avoid. We formally define the decision variant of our problem as follows:

Capacitated Resupply Problem (CRP)
Instance: A set of $n \in \mathbb{N}$ locations $N = \{1, \ldots, n\}$ to be supplied, $m \in \mathbb{N}$ vehicles available for resupply, vectors $c \in \mathbb{N}^n$ and $r \in \mathbb{N}^n$ with the maximum capacity and the demand rates per location, and $p \in \mathbb{N}$ the maximum payload of the vehicles.
Question: Is it possible to perform resupply of the locations with the given vehicles, such that there is never a time-unit where a stock-out occurs?

We show that CRP is intractable by a reduction from the Pinwheel Scheduling problem. In this problem, we have n' jobs with periods $p_1, \ldots, p_{n'}$. In each time-unit, we can schedule one job. The question is whether we can construct a perpetual schedule such that job i is scheduled in any period of p_i consecutive time-units. It is not known whether the Pinwheel Scheduling problem is NP-complete, or even contained in NP. It was shown by [5] that there cannot be a polynomial time exact algorithm for the Pinwheel Scheduling problem, unless Satisfiability can be solved in expected time $O(n^{\log n \log\log n})$, which is deemed unlikely.

▶ **Theorem 1.** *There is no polynomial time exact algorithm for CRP, unless the Satisfiability problem can be solved in expected time $O(n^{\log n \log\log n})$.*

Proof. Given an instance of the Pinwheel Scheduling problem, i.e., $p_1, \ldots, p_{n'}$, create an instance of CRP as follows. Create a location for each job, where location i has $r_i = \prod_{j=1}^{n'} p_j / p_i$, and $c_i = c = \prod_{j=1}^{n'} p_j$. Furthermore, we set $p = \prod_{j=1}^{n'} p_j$, and $m = 1$. Now, we can verify there exists a feasible resupply schedule for this instance of CRP if and only if there is a feasible schedule for the original instance of the Pinwheel Scheduling problem. ◀

The proof above shows it is hard to distinguish instances of CRP where one vehicle is sufficient, and instances for which at least two vehicles are necessary. This implies that for any $\alpha < 2$, there is no α-approximation for minimizing the number of vehicles, unless we can solve the Pinwheel Scheduling problem in polynomial time. The same holds for the Windows Scheduling problem, which can be seen as the multi-vehicle version of the Pinwheel Scheduling problem. For the Windows Scheduling problem, a 2-approximation is known [2]. It is an open question whether a 2-approximation exists for the optimization version of CRP. For now, we focus on the following homogeneous special case of this challenging problem:

Homogeneous Capacitated Resupply Problem (HCRP)
Instance: $n \in \mathbb{N}$ locations to be supplied, $m \in \mathbb{N}$ vehicles available for resupply, $c \in \mathbb{N}$ the maximum capacity per location, $r \in \mathbb{N}$ the demand rates at the locations, and $p \in \mathbb{N}$ the maximum payload of the vehicles, where $p \leq c$.
Question: Is it possible to perform resupply of the locations with the given vehicles, such that there is never a time-unit where a stock-out occurs?

More specifically, we consider the optimization variant where we seek the smallest m such that no stock-out occurs. As the input of an instance consists of four numbers, a polynomial time algorithm must be polynomial in $\log n$, $\log c$, $\log r$, and $\log p$.

3 Resupply policies

Let us now define the structure of the policies that we consider for the HCRP.

▶ **Definition 2** (Policy). *A policy $\pi(x^t, t, j) : \mathbb{N}^n \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ takes a vector x^t of stock levels at the locations right after the demand of time-unit t, the current time-unit t, and a vehicle index j, and produces the index of the location it visits in time-unit t.*

In each time-unit t we apply the policy π to the current stock vector x^t, by adding the payloads of the m available visiting vehicles during time-unit t up to the capacity and subtracting the demand of time-unit $t + 1$ as follows:

$$x_i^{t+1} = \min\left\{c,\ x_i^t + p\left|V_i^t\right|\right\} - r, \quad \text{with } V_i^t = \left\{j : j \in \{1, \ldots, m\},\ \pi(x^t, t, j) = i\right\} \tag{1}$$

We are interested in when policies can guarantee a stock-out never occurs, i.e., $x^t \geq 0$ for any time-unit t. We define the minimum number of vehicles needed to operate policy π without a stock-out as $m^*(\pi)$ and m^* as the minimum number of vehicles needed by *any* possible policy. We observe that the total supply per time-unit (at most mp) should exceed the total demand per time-unit (nr), otherwise the stocks will keep decreasing in the long run. It follows that $m^* \geq \left\lceil \frac{nr}{p} \right\rceil$. Furthermore, we argue that interesting cases have $r < p$, i.e., the vehicle payload is greater than the demand at a location. If the demand exceeds the payload, we can assign dedicated vehicles to each location and derive an instance with $m' = m - n\left\lfloor \frac{r}{p} \right\rfloor$, $c' = c - p\left\lfloor \frac{r}{p} \right\rfloor$, $r' = r - p\left\lfloor \frac{r}{p} \right\rfloor$ which has $r' < p$.

We now introduce three simple policies with specific operational advantages and consider their performance. First, the Wrap Around Policy produces a schedule where vehicles visit the same location consecutively, resulting in a sense of short-term consistency of the operations. Second, the No Migration Policy assigns a single vehicle to each location. Finally, the Shift Policy has a single sequence of visits operated by all vehicles at different offsets in time, resulting in consistent operations for the vehicles.

▶ **Policy 1** (Wrap Around Policy). *The* Wrap Around Policy π_{WA} *assigns r consecutive visits to each location and divides these sequentially within a period of p time-units. For example, if $r = 3$ and $p = 5$, vehicle 1 visits location 1 in the first three time-units and location 2 in time-units 4 and 5, whereas vehicle 2 visits locations 2 in the first time-unit, location 3 in time-units 2, 3 and 4, and location 4 in time-unit 5, etcetera. Formally, this is defined as:*

$$\pi_{\mathrm{WA}}(x^t, t, j) = 1 + \left\lfloor \frac{(t - 1 \bmod p) + (j-1)p}{r} \right\rfloor \tag{2}$$

▶ **Theorem 3.** *Policy 1 is optimal with $m^*(\pi_{WA}) = \left\lceil \frac{nr}{p} \right\rceil$ vehicles if $\frac{c}{r} \geq p$.*

Proof. Consider the first time-unit of a location where the stock starts decreasing below the maximum stock level c. Since it takes at least $\frac{c}{r} \geq p$ time-units for a stock-out to occur, and there are fewer than p time-units until the next visit, no stock-out can occur. The total decrease in stock level is at most $(p - r + 1)r$. Then, the location is visited in the next r time-units where the stock level can increase by $rp - (r-1)r$. After these visits the location is at full capacity again. We establish that no location can run out of stock. ◀

▶ **Lemma 4.** *If each location is visited at intervals of at most $\left\lfloor \frac{p}{r} \right\rfloor$ time-units, no stock-out occurs.*

Proof. The ratio $\frac{p}{r}$ which is the number of time-units before stock-out after performing a resupply, assuming the stock does not exceed capacity. Since we assume $c \geq p$ it holds that $\frac{c}{r} \geq \left\lfloor \frac{p}{r} \right\rfloor$. If each location is visited at least every $\left\lfloor \frac{p}{r} \right\rfloor$ time-units, the total supply satisfies the total demand and no stock-outs can occur. ◀

▶ **Policy 2** (No Migration Policy). *The* No Migration Policy π_{NM} *divides all locations over all vehicles such that each location is always visited by the same vehicle. It repeats this assignment every $\left\lfloor \frac{p}{r} \right\rfloor$ time-units. This implies the first vehicle visits locations $1, \ldots, \left\lfloor \frac{p}{r} \right\rfloor$, the second vehicle locations $\left\lfloor \frac{p}{r} \right\rfloor + 1, \ldots, 2\left\lfloor \frac{p}{r} \right\rfloor$, and so on. Formally, this is defined as follows:*

$$\pi_{\mathrm{NM}}(x^t, t, j) = 1 + (j-1)\left\lfloor \frac{p}{r} \right\rfloor + \left((t-1) \bmod \left\lfloor \frac{p}{r} \right\rfloor\right) \tag{3}$$

▶ **Theorem 5.** *Policy 2 gives a 2-approximation with $m^*(\pi_{NM}) = \left\lceil \frac{n}{\lfloor p/r \rfloor} \right\rceil$ vehicles.*

Proof. From Lemma 4, it follows that no stock-out occurs with Policy 2. As each vehicle has a periodic schedule of length $\lfloor p/r \rfloor$, this results in $\left\lceil \frac{n}{\lfloor p/r \rfloor} \right\rceil$ vehicles. Using that $\lfloor x \rfloor \geq x/2$ for $x \geq 1$, and $\lceil nx \rceil \leq n\lceil x \rceil$ for $n \in \mathbb{N}$, we get an approximation guarantee of

$$\frac{m^*(\pi_{\mathrm{NM}})}{m^*} = \frac{\left\lceil \frac{n}{\lfloor \frac{p}{r} \rfloor} \right\rceil}{m^*} \leq \frac{\left\lceil \frac{n}{\lfloor \frac{p}{r} \rfloor} \right\rceil}{\left\lceil \frac{nr}{p} \right\rceil} \leq \frac{\left\lceil \frac{n}{\frac{1}{2}\frac{p}{r}} \right\rceil}{\left\lceil \frac{nr}{p} \right\rceil} \leq \frac{2\left\lceil \frac{nr}{p} \right\rceil}{\left\lceil \frac{nr}{p} \right\rceil} = 2. \tag{4}$$

◀

▶ **Policy 3** (Shift Policy). *The* Shift Policy π_{SH} *lets each vehicle visit the same sequence of locations $1, \ldots, n$, but varies the starting point of each vehicle within this sequence by increments of $\left\lfloor \frac{p}{r} \right\rfloor$. Thus this policy repeats after n time-units. For example, if $n = 10$, $p = 9$ and $r = 3$, vehicle 1 start at location 1, vehicle 2 starts at location 4, vehicle 3 starts at location 7 and vehicle 4 starts at location 10. Formally, this is defined as follows:*

$$\pi_{\mathrm{SH}}(x^t, t, j) = 1 + \left(t - 1 + (j-1)\left\lfloor \frac{p}{r} \right\rfloor\right) \bmod n \tag{5}$$

▶ **Theorem 6.** *Policy 3 gives a 2-approximation with* $m^*(\pi_{NM}) = \left\lceil \frac{n}{\lfloor p/r \rfloor} \right\rceil$ *vehicles.*

The proof of Theorem 6 is similar to the proof of Theorem 5. Note that both Policy 2 and Policy 3 are optimal if p/r is an integer, i.e., if p is an integer multiple of r. The next example shows that the analysis for both Policy 2 and Policy 3 is tight.

▶ **Example 7.** Consider an instance with $n = 10$, $r = 10$, $p = 19$, and $c = 30$. Then, Policy 2 and 3 both result in a schedule with $m = \left\lceil \frac{n}{\lfloor p/r \rfloor} \right\rceil = 10$ vehicles. However, the optimal number of vehicles is equal to 6 with the following schedule. Apply the idea of Policy 3 to 5 vehicles with a shift of $\lceil p/r \rceil = 2$. Then, add a sixth vehicle with a schedule that is out of sync with the other schedules. This implies that both policies are a factor $\frac{5}{3}$ off. In general, we can set $r = n$, $p = 2n - 1$, and $c = 3n$, with n even. Then, Policy 2 and 3 use n vehicles, whereas there is an optimal schedule that uses $n/2 + 1$ vehicles.

4 A Greedy Policy

Up until now we considered policies for which the required number of vehicles can be easily derived, as they are only dependent on the current time-unit, which provides operational benefits. It is also interesting to consider a policy that assigns vehicles based on the current stock level. We consider a greedy policy that aims to maximally postpone stock-outs.

▶ **Policy 4** (Greedy Policy)**.** *The greedy policy* π_{GR} *assigns the vehicles to the locations with the lowest current stock levels. We define the sequence* $\sigma(x)$ *as a permutation of the indices* $1, \ldots, n$ *in increasing lexicographic order of their stock levels and indices, i.e.,* $(x_{\sigma(x)_i}, i) \leq_{\mathrm{lex}} (x_{\sigma(x)_j}, j)$ *for any* $i < j$*. Now the greedy policy is defined as follows:*

$$\pi_{\mathrm{GR}}(x^t, t, j) = \sigma(x)_j \tag{6}$$

▶ **Theorem 8.** *The greedy policy* π_{GR} *is* optimal*, i.e.* $m^*(\pi_{\mathrm{GR}}) = m^*$*.*

Proof. Given the current stock level x_i^t of a location i and assuming no future restocks occur, the time until a stock-out occurs can be expressed as $\frac{x_i^t}{r}$. In order to detect a stock-out, we only need to be concerned with the lowest stock level $x_{\min}^t = \min_{i=1\ldots n} x_i^t$.

First, we argue that the total stock level over all locations increases by the maximum amount possible with the greedy policy. Consider another policy that prefers restocking a location i which has a higher stock level than a location j that is not restocked by that policy. We thus have $x_i^t > x_j^t$. It is clear to see that the maximum amount that can be restocked at location i is at most the amount that can be restocked at location j, as $c - x_i^t < c - x_j^t$.

Second, we argue that the greedy policy *maximally postpones* stock-outs. Consider another policy that at time-unit t prefers restocking a location i which has a higher stock level than a location j that is not restocked by that policy. In case $x_j^{t+1} > x_{\min}^{t+1}$, location j is not critical and $x_{\min}^{t+1}$ will be equal for both policies. In case $x_j^{t+1} = x_{\min}^{t+1}$, the $x_{\min}^{t+1}$ of the greedy policy will be greater than or equal to that value of the other policy.

We conclude that among all policies, the greedy policy maximally postpones stock-outs as it maximally increases the total stock at the most critical locations. As locations are otherwise identical, the greedy policy can avoid stock-outs with m^* vehicles. ◀

If n/m is integer, the greedy schedule visits each location exactly once in each period of n/m consecutive time-units. It is easy to check if no stock-out can occur. If n/m is not integer, note that the time between two visits to a location in the greedy schedule is either $\lfloor n/m \rfloor$ or $\lceil n/m \rceil$. Observe that for the HCRP the greedy policy coincides with a round-robin policy defined as $\pi_{RR}(x^t, t, j) = 1 + (j - 1 + mr) \mod n$. For each location this results in a schedule with n_s periods of $\lfloor n/m \rfloor$ time-units and n_l periods of $\lfloor n/m \rfloor + 1$ time-units.

▶ **Theorem 9.** *If $n/m \notin \mathbb{N}$, the following statements hold.*

- *If $\lceil n/m \rceil \leq \lfloor p/r \rfloor$, then the greedy schedule is feasible.*
- *If $\lceil n/m \rceil \geq \lfloor p/r \rfloor + 2$, then the greedy schedule is infeasible.*
- *Else, the greedy schedule is feasible if and only if*

$$\min\{c,\ (n_s+1)p - n_s\lfloor p/r \rfloor r\} \geq n_l\left(\lfloor p/r \rfloor + 1\right)r - (n_l - 1)p. \tag{7}$$

Proof. If $\lceil n/m \rceil \leq \lfloor p/r \rfloor$, then every location is visited at least once every $\lfloor p/r \rfloor$ time-units. Hence, by Lemma 4, the schedule is feasible.

If $\lceil n/m \rceil \geq \lfloor p/r \rfloor + 2$, then the number of time-units between any two visits to a location is at least $\lfloor p/r \rfloor + 1$ which therefore consumes more than p between any two consecutive visits. Thus, the stock will eventually drop below 0.

Else, the refill after n_s periods of length $\lfloor p/r \rfloor$ time-units should be at least the decrease in n_l periods of length $\lfloor p/r \rfloor + 1$ time-units. If this inequality holds, the greedy schedule is feasible. It is also necessary, since a violation will lead to a stock-out. ◀

We now argue that we can solve the optimisation variant of the HCRP in pseudo-polynomial time. By using a binary search, we can solve the optimization variant by solving the decision variant $O(\log n)$ times. For this, Theorem 9 can be used which takes at most $O(n)$ time as π_{RR} has a period of at most n time-units. More precisely, when $\lceil n/m \rceil = \lfloor p/r \rfloor + 1$, we can find n_s and n_l from the periodic schedule with length at most $\text{lcm}(n,m)/m$. Hence, we can find m^* for HCRP in $O(n \log n)$ time, which is pseudo-polynomial in the input size.

5 Conclusions and Future Research

We consider the capacitated resupply problem (CRP) where locations with a given demand rate should be resupplied to avoid a stock-out. We show the CRP to be intractable and consider the homogeneous variant (HCRP) for which we present several policies. We conclude with a greedy policy that can be used to find the optimal solution in pseudo-polynomial time.

The insights provided form the basis for extending the analysis to problems with more realistic characteristics, such as multi-commodity resupply, locations with differing capacity and (time-varying) demand, vehicles with different payloads, and travel times that may not be insignificant compared to the resupply time. We believe these extensions provide interesting opportunities for future research.

References

1. Edgar Arribas, Vicent Cholvi, and Vincenzo Mancuso. Optimizing uav resupply scheduling for heterogeneous and persistent aerial service. *IEEE Transactions on Robotics*, 2023.
2. Amotz Bar-Noy and Richard E Ladner. Windows scheduling problems for broadcast systems. *SIAM Journal on Computing*, 32(4):1091–1113, 2003.
3. Sofie Coene, Frits C. R. Spieksma, and Gerhard J. Woeginger. Charlemagne's challenge: The periodic latency problem. *Operations Research*, 59(3):674–683, 2011.
4. Robert Holte, Al Mok, Louis Rosier, Igor Tulchinsky, and Donald Varvel. The pinwheel: A real-time scheduling problem. In *Proceedings of the 22th Annual Hawaii International Conference on System Sciences*, volume 2, pages 693–702, 1989.
5. Tobias Jacobs and Salvatore Longo. A new perspective on the windows scheduling problem. *arXiv preprint arXiv:1410.7237*, 2014.