

Approximation algorithms for the joint replenishment problem with deadlines

Marcin Bienkowski · Jarosław Byrka · Marek Chrobak · Neil Dobbs ·
Tomasz Nowicki · Maxim Sviridenko · Grzegorz Świrszcz · Neal E. Young

Received: 17 May 2014 / Accepted: 29 July 2014 / Published online: 27 August 2014
© Springer Science+Business Media New York 2014

Abstract The Joint Replenishment Problem (JRP) is a fundamental optimization problem in supply-chain management, concerned with optimizing the flow of goods from a supplier to retailers. Over time, in response to demands at the retailers, the supplier ships orders, via a warehouse, to the retailers. The objective is to schedule these orders to minimize the sum of ordering costs and retailers' waiting costs. We study the approximability of JRP-D, the version of JRP with deadlines, where instead of waiting costs the retailers impose strict deadlines. We study the integrality gap of the standard linear-program (LP) relaxation, giving a lower bound of 1.207, a stronger, computer-assisted lower bound of 1.245, as well as an upper bound and approximation ratio of 1.574. The best previous upper bound and approximation ratio was 1.667; no lower bound was previously published. For the special case when all demand periods are of equal length, we give an upper bound of 1.5, a lower bound of 1.2, and show APX-hardness.

Keywords Joint replenishment problem · NP-completeness · APX-hardness · Approximation algorithms

M. Bienkowski · J. Byrka
Institute of Computer Science, University of Wrocław,
Wrocław, Poland

M. Chrobak (✉) · N. E. Young
Department of Computer Science, University of California
at Riverside, Riverside, CA, USA
e-mail: marek@cs.ucr.edu

N. Dobbs · T. Nowicki · G. Świrszcz
IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

M. Sviridenko
Department of Computer Science, University of Warwick,
Coventry, UK

1 Introduction

The *Joint Replenishment Problem with Deadlines* (JRP-D) is an optimization problem in supply-chain management concerned with scheduling shipments (orders) of a commodity from a supplier, via a shared warehouse, to satisfy prior demands at m retailers (cf. Fig. 1). The objective is to find a schedule of orders that satisfies all demands before their deadlines expire, while minimizing the total ordering cost.

Specifically, an instance of JRP-D is given by a tuple (C, c, D) where

- $C \in \mathbb{Q}$ is the *warehouse ordering cost*;
- c is the vector of *retailer ordering costs*, where for each retailer $\rho \in \{1, 2, \dots, m\}$ its ordering cost is $c_\rho \in \mathbb{Q}$;
- D is a set of n *demands*, with each demand represented by a triple (ρ, r, d) , where ρ is the retailer that issued the demand, $r \in \mathbb{Q}$ is the demand's release time and $d \in \mathbb{Q}$ is its deadline.

For a demand (ρ, r, d) , the interval $[r, d]$ is called the *demand period*.¹ In sections that prove upper bounds we assume (without loss of generality by time scaling) that $r, d \in [2n]$, where $[i]$ denotes $\{1, 2, \dots, i\}$.

A solution (also called a *schedule*) is a set of *orders*, each specified by a pair (t, R) , where t is the time of the order and R is a subset of the retailers. An order (t, R) *satisfies* those demands (ρ, r, d) whose retailer is in R and whose demand period contains t (that is, $\rho \in R$ and $t \in [r, d]$). A schedule is *feasible* if all demands are satisfied by some order in the schedule.

¹ Note: our use of the term “period” is different from its use in operations research literature on supply-chain management problems.

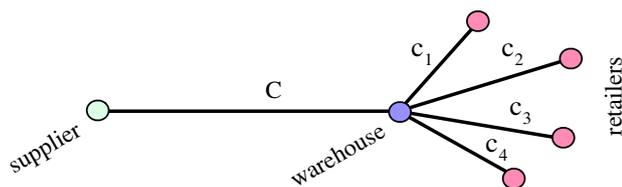


Fig. 1 An instance with four retailers, represented by a tree with ordering costs as weights assigned to its edges. The cost of an order is the total weight of the subtree connecting the supplier and the involved retailers

The *cost* of order (t, R) is the ordering cost of the warehouse plus the ordering costs of respective retailers, i.e., $C + \sum_{\rho \in R} c_{\rho}$. It is convenient to think of this order as consisting of a warehouse order of cost C , which is then joined by each retailer $\rho \in R$ at cost c_{ρ} . The cost of the schedule is the sum of the costs of its orders. The objective is to find a feasible schedule of minimum cost.

Previous results The decision variant of JRP-D was shown to be strongly NP-complete by Becchetti et al. (2009). (They considered an equivalent problem of packet aggregation with deadlines on two-level trees.) Nonner and Souza (2009) then showed that JRP-D is APX-hard, even if each retailer issues only three demands. Levi et al. (2006) gave a 2-approximation algorithm based on a primal-dual scheme. Using randomized rounding, Levi et al. (2008); Levi and Sviridenko (2006) (building on Levi et al. 2005) improved the approximation ratio to 1.8; (Nonner and Souza 2009) reduced it further to $5/3$. These results use a natural linear-program (LP) relaxation, which we use too.

The randomized-rounding approach from Nonner and Souza (2009) uses a natural rounding scheme whose analysis can be reduced to a probabilistic game. For any probability distribution p on $[0, 1]$, the integrality gap of the LP relaxation is at most $1/\mathcal{Z}(p)$, where $\mathcal{Z}(p)$ is a particular statistic of p (see Lemma 1). The challenge in this approach is to find a distribution where $1/\mathcal{Z}(p)$ is small. Nonner and Souza show that there is a distribution p with $1/\mathcal{Z}(p) \leq 5/3 \approx 1.67$. As long as the distribution can be sampled from efficiently, the approach yields a polynomial-time $(1/\mathcal{Z}(p))$ -approximation algorithm.

Our contributions We prove that there is a distribution p with $1/\mathcal{Z}(p) \leq 1.574$. We present this result in two steps: we show the bound $e/(e-1) \approx 1.58$ with a simple and elegant analysis, then improve it to 1.574 by refining the underlying distribution. This shows that the integrality gap is at most 1.574 and it gives a 1.574-approximation algorithm. We also prove that the LP integrality gap is at least 1.207 and we provide a computer-assisted proof that this gap is at least 1.245. As far as we know, no explicit lower bounds have been previously published.

For the special case when all demand periods have the same length (as occurs in applications where time-to-delivery

is globally standardized), we give an upper bound of 1.5, a lower bound of 1.2, and show APX-hardness.

Other related work JRP-D is a special case of the Joint Replenishment Problem (JRP). In JRP, instead of having a deadline, each demand is associated with a delay-cost function that specifies the cost for the delay between the time the demand is released and the time it is satisfied by an order. JRP is NP-complete, even if the delay cost is linear (Arkin et al. 1989; Nonner and Souza 2009). JRP is in turn a special case of the One-Warehouse Multi-Retailer (OWMR) problem, where the commodities may be stored at the warehouse for a given cost per time unit. The 1.8-approximation by Levi et al. (2006) holds also for OWMR. JRP was also studied in the online scenario: a 3-competitive algorithm was given by Buchbinder et al. (2008) (see also Brito et al. 2012).

The JRP model is an abstraction of a number of other optimization problems that arise in supply-chain management. It is often presented as an inventory-management problem, where all demands need to be satisfied immediately from the current inventory. In that scenario, orders are issued to replenish the inventory, ensuring that all future demands are met. (In contrast, in our model the orders are issued to satisfy *past* demands and there is no inventory.) Depending on the application, orders can represent deliveries (via a shared warehouse), or a manufacturing process that involves a joint set-up cost and individual set-up costs for retailers. The objective is to minimize the total cost, defined as the sum of ordering costs and inventory holding costs.

Another generalization of JRP involves a tree-like structure with the supplier in the root and retailers at the leaves, modeling control packet aggregation in computer networks. A 2-approximation is known for the variant with deadlines (Becchetti et al. 2009); the case of linear delay costs has also been studied (Khanna et al. 2002; Brito et al. 2012). Recently, Chaves (private communication) has shown that the generalization of JRP to arbitrary trees, even for arbitrary waiting cost functions, can be approximated within a factor of 2 through a reduction to the multi-stage assembly problem, see Levi et al. (2006).

2 Upper bound of 1.574

In this section, we derive our approximation algorithms for JRP-D, showing an approximation ratio of $e/(e-1) \approx 1.58$, which we then improve to 1.574. Both algorithms are based on randomized LP-rounding.

The LP relaxation For the rest of this section, fix an arbitrary instance $\mathcal{I} = (C, \mathbf{c}, \mathcal{D})$ of JRP-D. Let finite set $\mathcal{U} \subset \mathbb{Q}$ contain the release times and deadlines. Here is the standard LP relaxation of the problem:

$$\begin{aligned}
 &\text{minimize} && \text{cost}(\mathbf{x}) = \sum_{t \in \mathcal{U}} (\mathbf{C}x_t + \sum_{\rho=1}^m c_\rho x_t^\rho) \\
 &\text{subject to} && x_t, x_t^\rho \geq 0 && \text{for all } t \in \mathcal{U}, \rho \in \{1, \dots, m\} \\
 &&& x_t \geq x_t^\rho && \text{for all } t \in \mathcal{U}, \rho \in \{1, \dots, m\} \\
 &&& \sum_{t \in \mathcal{U} \cap [r,d]} x_t^\rho \geq 1 && \text{for all } (\rho, r, d) \in \mathcal{D}.
 \end{aligned}
 \tag{1}$$

The statistic $\mathcal{Z}(p)$. Let p be a probability distribution on $[0, 1]$. As we are about to show, the approximation ratio of algorithm Round_p (defined below) and the integrality gap of the LP are at most $1/\mathcal{Z}(p)$, where $\mathcal{Z}(p)$ is defined by the following so-called *tally game* (following Nonner and Souza 2009). To begin the game, fix any *threshold* $z \geq 0$, then draw a sequence of independent samples s_1, s_2, \dots, s_h from p , stopping when their sum exceeds z , that is when $s_1 + s_2 + \dots + s_h > z$. Call $z - (s_1 + s_2 + \dots + s_{h-1})$ the *waste*. Note that, since the waste is less than s_h , it is in $[0, 1]$. Let $\mathcal{W}(p, z)$ denote the expectation of the waste. Abusing notation, let $\mathbf{E}[p]$ denote the expected value of a single sample drawn from p . Then $\mathcal{Z}(p)$ is defined by

$$\mathcal{Z}(p) = \min \left\{ \mathbf{E}[p], 1 - \sup_{z \geq 0} \mathcal{W}(p, z) \right\}.$$

Note that the distribution p that chooses $\frac{1}{2}$ with probability 1 has $\mathcal{Z}(p) = \frac{1}{2}$. The challenge is to simultaneously increase $\mathbf{E}[p]$ and reduce the maximum expected waste.

A generic randomized-rounding algorithm The upper bound of 1.574 relies on a randomized-rounding algorithm, Round_p . The algorithm is parameterized by an arbitrary probability distribution p on $[0, 1]$ and gives a $(1/\mathcal{Z}(p))$ -approximation:

Lemma 1 *For any distribution p on $[0, 1]$ and fractional LP solution \mathbf{x} , if $\mathcal{Z}(p) > 0$, then with probability 1, Algorithm $\text{Round}_p(\mathbf{C}, \mathbf{c}, \mathcal{D}, \mathbf{x})$ returns a feasible schedule. The expected cost of the schedule is at most $\text{cost}(\mathbf{x})/\mathcal{Z}(p)$.*

The main ideas underlying Round_p and its analysis are from Nonner and Souza (2009); the presentation here (Sect. 2.1) addresses some technical subtleties. Subsequent sections (with a minor exception) use Lemma 1 as a black box; they can be read independently of the proof in Sect. 2.1.

2.1 The details of Round_p and proof of Lemma 1

Fix an arbitrary optimal fractional solution \mathbf{x} of the LP relaxation for instance \mathcal{I} . As previously discussed, without loss of generality we can assume that the given universe \mathcal{U} of release times and deadlines is $[U]$, where $U (\leq 2n)$ is the maximum deadline. We focus on producing a “rounded”

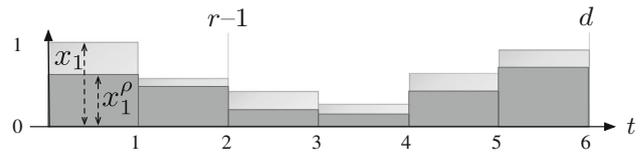


Fig. 2 The continuous solution $\bar{\mathbf{x}}$: the universe is $\bar{\mathcal{U}} = (0, U]$. At each time t in $\bar{\mathcal{U}}$, the solution ships at rate $\bar{x}_t = x_{\lceil t \rceil}$, while each retailer ρ takes at rate $\bar{x}_t^\rho = x_{\lceil t \rceil}^\rho$. During any demand period $(r - 1, d]$, the retailer’s cumulative take is at least 1

schedule S for \mathcal{I} with expected cost at most $1/\mathcal{Z}(p)$ times $\text{cost}(\mathbf{x}) = \sum_{j=1}^U (\mathbf{C}x_j + \sum_{\rho} c_\rho x_j^\rho)$.

Extend to continuous time To start, we recast the problem of rounding \mathbf{x} as a continuous-time problem. Extend the universe \mathcal{U} of times from the discrete set $\mathcal{U} = [U]$ to the continuous interval $\bar{\mathcal{U}} = (0, U]$ and relax each demand period $[r, d]$, replacing it with the half-open period $(r - 1, d]$. Let $\bar{\mathcal{I}}$ denote the modified instance. To find a schedule S for the given instance \mathcal{I} , we will find a schedule \bar{S} for $\bar{\mathcal{I}}$, then take $S = \{(\lceil t \rceil, R) : (t, R) \in \bar{S}\}$. S clearly has the cost not larger than that of \bar{S} and is also feasible (because the release times and deadline are integers).

For the algorithm, reinterpret the fractional solution \mathbf{x} as a *continuous-time* solution $\bar{\mathbf{x}}$ over universe $\bar{\mathcal{U}}$: as $t \in \bar{\mathcal{U}}$ ranges continuously from 0 to U , the continuous-time solution $\bar{\mathbf{x}}$ ships continuously at the *shipping rate* $\bar{x}_t = x_{\lceil t \rceil}$ and has each retailer ρ join at his *take rate* $\bar{x}_t^\rho = x_{\lceil t \rceil}^\rho$. The example at the top of Fig. 2 illustrates $\bar{\mathbf{x}}$ over time.

At each time t in $\bar{\mathcal{U}}$, define the *total shipped* up to time t to be $\sigma(t) = \int_0^t \bar{x}_t dt = \int_0^t x_{\lceil t \rceil} dt$. Define ρ ’s *take* up to time t to be $\tau_\rho(t) = \int_0^t \bar{x}_t^\rho dt = \int_0^t x_{\lceil t \rceil}^\rho dt$. Over any interval $(t, t']$, define the amount shipped to be $\sigma(t') - \sigma(t)$. Likewise define ρ ’s take to be $\tau_\rho(t') - \tau_\rho(t)$.

The algorithm Round_p draws samples (s_1, s_2, \dots, s_I) i.i.d. from the distribution p , stopping when the sum of the samples first exceeds $\sigma(U) - 1$. It creates orders at the times $\mathbf{t} = (t_1, t_2, \dots, t_I)$ such that, for each $i \in [I]$, the continuous-time solution $\bar{\mathbf{x}}$ ships s_i units in interval $(t_{i-1}, t_i]$ (interpreting t_0 as 0). By the choice of the number of samples I , $\bar{\mathbf{x}}$ ships strictly less than 1 unit in $(t_I, U]$.

After Round_p chooses \mathbf{t} , for each retailer ρ independently, it has ρ join a minimum-size subset of the orders that satisfies ρ ’s demands. (It computes this optimal subset using the standard earliest-deadline-first algorithm.) This determines the schedule \bar{S} for the continuous-time instance $\bar{\mathcal{I}}$. To get the schedule S for the original instance \mathcal{I} , the algorithm shifts each order time t_i to its ceiling $\lceil t_i \rceil$. The algorithm is shown in Fig. 3.

We remark that, in practice, modifying the algorithm to round up the times as soon as they are chosen might yield lower-cost solutions for some instances. That is, take each t_i to be the minimum integer such that the amount

Fig. 3 Round_p(C, c, D, x) randomly rounds the continuous-time fractional solution \bar{x}

```

1: Draw samples  $(s_1, s_2, s_3, \dots, s_I)$  i.i.d. from  $p$ , stopping when the sum of the samples first
   exceeds  $\sigma(U) - 1$ . Schedule orders at times in  $\mathbf{t} = (t_1, t_2, \dots, t_I)$  such that, for each  $i \in [I]$ ,
   each  $t_i$  is the minimum such that  $\bar{x}$  ships  $s_i$  units in the interval  $(t_{i-1}, t_i]$  (interpreting  $t_0$ 
   as 0). (By the choice of  $I$ ,  $\bar{x}$  ships strictly less than 1 unit in interval  $(t_I, U]$ .)
2: for each  $\rho \in [m]$  do
3:   Use the earliest-deadline-first algorithm to choose a minimum-size subset of the orders for  $\rho$ 
   to join to satisfy his demands. More explicitly:
4:   while retailer  $\rho$  has any not-yet-satisfied demand  $(\rho, r, d)$  do
5:     Let  $d^*$  be the earliest deadline of such a demand.
6:     Have  $\rho$  join the order at time  $T = \max\{t_i \in \mathbf{t} : t_i \leq d^*\}$ .
7:   end while
8: end for
9: Let  $\bar{S}$  denote the resulting schedule for  $\bar{X}$ . Return  $S = \{(\lceil t \rceil, R) : (t, R) \in \bar{S}\}$ .

```

shipped over interval $(t_{i-1}, t_i]$ is at least s_i (interpreting t_0 as 0). Stop when the amount shipped over $(t_i, U]$ is less than 1.

Proof (Lemma 1) Correctness and feasibility We claim first that the number I of order times in line 1 has finite expectation; in other words, with probability 1, I is finite—line 1 finishes. This follows from standard calculation: I is the number of samples taken before the sum of the samples exceeds $\sigma(U) - 1$. Since $\mathbf{E}[p] \geq \mathcal{Z}(p) > 0$, there exists $\epsilon > 0$ such that $\Pr[s \geq \epsilon] \geq \epsilon$. Thus, the expected number of samples needed to increase the sum by ϵ is at most $1/\epsilon$. By linearity of expectation, the expected number of samples needed to increase the sum by $\sigma(U)$ is at most $\sigma(U)/\epsilon^2$. Thus, $\mathbf{E}[I] \leq \sigma(U)/\epsilon^2 < \infty$.

In the argument below (for cost estimation) we show that each iteration of the inner loop on line 4 satisfies some not yet-satisfied demand (ρ, r, d) . Thus, with probability 1, Round_p terminates. By inspection, it does not terminate until all demands are satisfied. Thus, with probability 1, Round_p returns a feasible solution.

Cost of the schedule We use the following basic properties of \bar{x} .

1. Over any interval $(t, t'] \subseteq \bar{U}$, each retailer ρ 's take is at most the amount shipped. This follows directly from LP constraint (1).
2. For each demand (ρ, r, d) , retailer ρ 's take over the demand period $(r - 1, d]$ is at least 1. This holds because the take equals $\sum_{t=r}^d x_t$, which is at least 1 by LP constraint (2). Consequently, the amount shipped over $(r - 1, d]$ is also at least 1.

The given fractional solution \mathbf{x} has warehouse cost $C \sigma(U)$ and retailer cost $\sum_{\rho} c_{\rho} \tau_{\rho}(U)$. (Recall that $\sigma(U)$ is the amount \bar{x} ships up to time U while $\tau_{\rho}(U)$ is ρ 's take up to time U .)

The algorithm's schedule S has warehouse cost $C I$ and retailer cost $\sum_{\rho} c_{\rho} J_{\rho}$, where I is the number of orders placed in line 1 and J_{ρ} is the number of those orders joined by ρ in lines 3–7.

We show $\mathbf{E}[I] \leq \sigma(U)/\mathcal{Z}(p)$ and $\mathbf{E}[J_{\rho}] \leq \tau_{\rho}(U)/\mathcal{Z}(p)$. By linearity of expectation, these bounds imply that the schedule's expected cost is at most $1/\mathcal{Z}(p)$ times the cost of \mathbf{x} , proving the lemma.

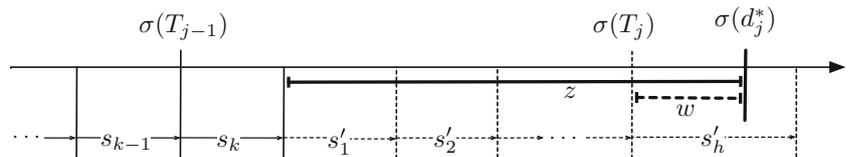
First analyze $\mathbf{E}[I]$, the expected number of samples until the sum of the samples exceeds $\sigma(U) - 1$. Clearly I is a stopping time.² As noted previously, I has finite expectation. So, by Wald's Lemma (see Appendix), the expectation of the sum of the first I samples is not smaller than the expectation of I times the expectation of each sample: $\mathbf{E}[\sum_{i=1}^I s_i] \geq \mathbf{E}[I] \mathbf{E}[p]$. The sum is at most $\sigma(U)$, because the sum of the first $I - 1$ samples is less than $\sigma(U) - 1$ and the last sample is at most 1. Thus, $\sigma(U) \geq \mathbf{E}[I] \mathbf{E}[p]$. Rearranging, $\mathbf{E}[I] \leq \sigma(U)/\mathbf{E}[p] \leq \sigma(U)/\mathcal{Z}(p)$, as desired.

Next analyze $\mathbf{E}[J_{\rho}]$. Fix a retailer $\rho \in [m]$. Focus on the inner loop, lines 3–7. For each iteration $j \in [J_{\rho}]$, let d_j^* and T_j denote, respectively, the value of d^* and T in iteration j . The order that ρ joins at time T_j indeed satisfies that iteration's unmet demand (ρ, r, d_j^*) , because ρ 's take over the period $(r - 1, d_j^*]$ is at least 1, so the amount shipped over the period is at least 1, so, by the choice of \mathbf{t} in line 1, the period has to contain some order time t_i , and $T_j \in [t_i, d_j^*]$. Then, by a standard induction on j , after ρ joins the order at time T_j , all of ρ 's demands whose demand periods overlap $(0, T_j]$ are satisfied. Hence ρ 's order times are strictly increasing: $T_1 < T_2 < \dots < T_{J_{\rho}}$.

Consider any non-final iteration j of the loop. Define $\psi_j = (s_1, s_2, \dots, s_k)$, the state at the end of iteration j , to be the first k samples drawn from p , where $k = k_j$ is the number of samples needed to determine T_j in iteration j . Explicitly, k is determined by the condition

² That is, for any $i \in \mathbb{N}$, the event $I = i$ is determined by the first i samples.

Fig. 4 The increase in ρ 's take in each iteration corresponds to a play of the tally game



$$t_{k-1} \leq d_j^* < t_k, \tag{3}$$

which implies $T_j = t_{k-1}$. (The sample s_k is included in ψ_j because, for T_j to be the maximum order time less than or equal to d_j^* , the order time following T_j must exceed d_j^* .) When we look at the related warehouse shipments, $t_{k-1} \leq d_j^* < t_k$ implies $\sigma(t_{k-1}) \leq \sigma(d_j^*) \leq \sigma(t_k)$. The last inequality is in fact strict, because the algorithm chooses t_k minimally. Since \bar{x} ships each s_i over the interval $(t_{i-1}, t_i]$, the last relation implies that

$$\sum_{i=1}^{k-1} s_i \leq \sigma(d_j^*) < \sum_{i=1}^k s_i. \tag{4}$$

Define ρ 's take during iteration j , denoted X_j , to be ρ 's take over the interval $(T_{j-1}, T_j]$ (interpreting T_0 as 0). To finish the proof, consider the sum $\sum_{j=1}^{J_\rho-1} X_j$, that is, ρ 's take up to the start of the last iteration.

The sum's upper index $J_\rho - 1$ is a stopping time. (Indeed, ψ_j determines which of ρ 's demands remain unsatisfied at the start of iteration $j + 1$. Iteration $j + 1$ will be the final iteration J_ρ iff those unsatisfied demands can be satisfied by a single order. Thus, ψ_j determines whether $J_\rho - 1 = j$.) Clearly $J_\rho - 1$ has finite expectation. (Indeed, J_ρ is at most the number of demands.)

We claim that expectation of each term X_j in the sum, given the state at the start of iteration j , is at least $\mathcal{Z}(p)$:

$$\mathbf{E}[X_j \mid \psi_{j-1}] \geq \mathcal{Z}(p). \tag{5}$$

Before we prove Claim (5), observe that it implies the desired bound on J_ρ , as follows. The upper index $J_\rho - 1$ of the sum $\sum_{j=1}^{J_\rho-1} X_j$ is a stopping time with finite expectation, and the conditional expectation of each term is at least $\mathcal{Z}(p)$, so, by Wald's Lemma (see Appendix), the expectation of the sum is at least $\mathbf{E}[J_\rho - 1]\mathcal{Z}(p)$. On the other hand, the value of the sum never exceeds $\tau_\rho(U) - 1$. (Indeed, at the start of the last iteration $j = J_\rho$, some demand (ρ, r, d) remains unsatisfied, and that demand, which has total take at least 1, does not overlap $(0, T_{j-1}]$, so ρ 's take up to time T_{j-1} can be at most 1 less than the total take.) Thus, $\mathbf{E}[J_\rho - 1]\mathcal{Z}(p) \leq \tau_\rho(U) - 1$. Since $\mathcal{Z}(p) \leq 1$, this implies the desired bound $\mathbf{E}[J_\rho] \leq \tau_\rho(U)/\mathcal{Z}(p)$.

To finish, we prove Claim (5). Fix any state ψ_{j-1} and let $k = k_{j-1} = |\psi_{j-1}|$. Consider iteration j . Note that ψ_{j-1} determines both T_{j-1} and d_j^* . Call the samples in

ψ_j , but not in ψ_{j-1} newly exposed. Crucially, ψ_{j-1} does not condition the newly exposed samples. Let random variable $h = k_j - k_{j-1}$ be the number of newly exposed samples. By Condition (4), h is the index such that $\sum_{i=1}^{k+h-1} s_i \leq \sigma(d_j^*) < \sum_{i=1}^{k+h} s_i$.

Define $z = \sigma(d_j^*) - \sum_{i=1}^k s_i$. Then $z \geq 0$ with probability 1 and z is determined by ψ_{j-1} . Using s'_1, s'_2, \dots, s'_h to denote the newly exposed samples (i.e., $s'_\ell = s_{k+\ell}$), the condition on h above is equivalent to

$$s'_1 + s'_2 + \dots + s'_{h-1} \leq z < s'_1 + s'_2 + \dots + s'_{h-1} + s'_h.$$

That is, the iteration exposes new samples just until their sum exceeds z . Upon consideration, this process corresponds to a play of the tally game with threshold z , in the definition of the statistic $\mathcal{Z}(p)$. (See Fig. 4.) Recall from the definition of $\mathcal{Z}(p)$ that the waste is $w = z - (s'_1 + s'_2 + \dots + s'_{h-1})$.

By inspection, using that $\sum_{i=1}^{k+h-1} s_i = \sigma(T_j)$, the waste w in this setting equals $\sigma(d_j^*) - \sigma(T_j)$, so ρ 's take X_j during the iteration, that is, $\tau_\rho(T_j) - \tau_\rho(T_{j-1})$, equals

$$\begin{aligned} & [\tau_\rho(d_j^*) - \tau_\rho(T_{j-1})] - [\tau_\rho(d_j^*) - \tau_\rho(T_j)] \\ & \geq 1 - [\tau_\rho(d_j^*) - \tau_\rho(T_j)] \\ & \geq 1 - [\sigma(d_j^*) - \sigma(T_j)] = 1 - w. \end{aligned}$$

The first inequality holds because, as observed previously, ρ 's take over interval $(T_{j-1}, d_j^*]$ is at least 1. The next inequality holds because ρ 's take over $(T_j, d_j^*]$ is at most the amount shipped.

Recall that, by definition of $\mathcal{Z}(p)$, the expectation of $(1 - w)$ is at least $\mathcal{Z}(p)$. Thus, the inequality above implies Claim (5)—that the conditional expectation of each X_j is at least $\mathcal{Z}(p)$. \square

The next utility lemma says that, in analyzing the expected waste in the tally game, it is enough to consider thresholds z in $[0, 1)$.

Lemma 2 For any distribution p on $[0, 1]$, $\sup_{z \geq 0} \mathcal{W}(p, z) = \sup_{z \in [0, 1)} \mathcal{W}(p, z)$.

Proof Play the tally game with any threshold $z \geq 1$. Consider the first prefix sum $s_1 + s_2 + \dots + s_h$ of the samples, such that the “slack” $z - (s_1 + s_2 + \dots + s_h)$ is less than 1. Let random variable z' be this slack. Note that $z' \in [0, 1)$. For any value $u \in [0, 1)$, the expected waste conditioned on the event “ $z' = u$ ” is $\mathcal{W}(p, u)$, which is at most $\sup_{y \in [0, 1)} \mathcal{W}(p, y)$. Thus, for any threshold $z \geq 1$, $\mathcal{W}(p, z)$ is at most $\sup_{y \in [0, 1)} \mathcal{W}(p, y)$. \square

2.2 Upper bound of $e/(e - 1) \approx 1.582$

Consider the specific probability distribution p on $[0, 1]$ with probability density function $p(y) = 1/y$ for $y \in [1/e, 1]$ and $p(y) = 0$ elsewhere.

Lemma 3 For this distribution p , $\mathcal{Z}(p) \geq (e-1)/e = 1 - 1/e$.

Proof By Lemma 2, $\mathcal{Z}(p)$ is the minimum of $\mathbf{E}[p]$ and $1 - \sup_{z \in (0,1)} \mathcal{W}(p, z)$.

By direct calculation, $\mathbf{E}[p] = \int_{1/e}^1 y p(y) dy = \int_{1/e}^1 1 dy = 1 - 1/e$. Now consider playing the tally game with threshold z . If $z \in [0, 1/e]$, then (since the waste is at most z) trivially $\mathcal{W}(p, z) \leq z \leq 1/e$. So, consider any $z \in [1/e, 1]$. Let s_1 be just the first sample. The waste is z if $s_1 > z$ and otherwise is at most $z - s_1$. Thus, the expected waste is

$$\begin{aligned} \mathcal{W}(p, z) &\leq \Pr[s_1 > z] \cdot z + \Pr[s_1 \leq z] \\ &\quad \cdot \mathbf{E}[z - s_1 \mid s_1 \leq z] \\ &= z - \Pr[s_1 \leq z] \cdot \mathbf{E}[s_1 \mid s_1 \leq z] \\ &= z - \int_{1/e}^z y p(y) dy = z - \int_{1/e}^z dy \\ &= z - (z - 1/e) = 1/e. \end{aligned}$$

Since both $\mathbf{E}[p]$ and $1 - \sup_{z \in (0,1)} \mathcal{W}(p, z)$ are at least $1 - 1/e$, the lemma follows. \square

From Lemma 3 and Lemma 1, the approximation ratio of our algorithm Round_p , with distribution p defined above, is at most $1/\mathcal{Z}(p) = \frac{e}{e-1} \approx 1.582$.

2.3 Upper bound of 1.574

On close inspection of the proof of Lemma 3, it is not hard to see that the estimate for the waste in that proof is likely not tight. The reason is that the proof estimates the waste based on just the first sample, while, for the distribution being analyzed, there is non-zero probability that *two* samples are generated before reaching the threshold, further reducing the waste. To improve the upper bound, we adjust the probability distribution (and the analysis) accordingly.

Define a probability distribution p on $[0, 1]$, having a point mass at 1, as follows. Fix $\theta = 0.36455$ (slightly less than $1/e$). Over the half-open interval $[0, 1)$, the probability density function is

$$p(y) = \begin{cases} 0 & \text{for } y \in [0, \theta) \\ 1/y & \text{for } y \in [\theta, 2\theta) \\ \frac{1 - \ln((y-\theta)/\theta)}{y} & \text{for } y \in [2\theta, 1). \end{cases}$$

Define the probability of choosing 1 to be $1 - \int_0^1 p(y) dy \approx 0.0821824$. Note that $p(y) \geq 0$ for $y \in [2\theta, 1)$ since

$\ln((1 - \theta)/\theta) \approx 0.55567$, so p is indeed a probability distribution on $[0, 1]$.

Lemma 4 The statistic $\mathcal{Z}(p)$ for this p is at least $0.63533 > 1/1.574$.

Proof Recall that $\mathcal{Z}(p) = \min\{\mathbf{E}[p], 1 - \sup_{z \in (0,1)} \mathcal{W}(p, z)\}$. By calculation, the probability measure μ induced by p has $\mu[1] \approx 0.0821824$ and

$$\mu[0, v) = \begin{cases} 0 & \text{for } v \in [0, \theta) \\ \ln(v/\theta) & \text{for } v \in [\theta, 2\theta) \\ \ln(v/\theta) - \int_{2\theta}^v \frac{\ln((y-\theta)/\theta)}{y} dy & \text{for } v \in [2\theta, 1). \end{cases}$$

The following calculation shows $\mathbf{E}[p] > 0.63533$:

$$\begin{aligned} \mathbf{E}[p] &= \mu[1] + \int_{\theta}^1 y p(y) dy \\ &= \mu[1] + \int_{\theta}^1 dy - \int_{2\theta}^1 \ln((y - \theta)/\theta) dy \\ &= \mu[1] + (1 - \theta) - ((y - \theta) \ln((y - \theta)/\theta) - y) \Big|_{2\theta}^1 \\ &= \mu[1] + 2 - 3\theta - (1 - \theta) \ln((1 - \theta)/\theta) \\ &> 0.0821 + 2 - 3 \cdot 0.36455 - (1 - 0.36455) \cdot 0.5557 \\ &> 0.63533. \end{aligned}$$

To finish, we show $\sup_{z \geq 0} \mathcal{W}(p, z) = \theta \leq 1 - 0.63533$.

By Lemma 2, assume that $z \in [0, 1)$. In the tally game defining $\mathcal{W}(p, z)$, let s_1 be the first random sample drawn from p . If $s_1 > z$, then the waste equals z . Otherwise, the process continues recursively with z replaced by $z' = z - s_1$. This gives the recurrence

$$\begin{aligned} \mathcal{W}(p, z) &= z \mu[z, 1] + \int_0^z \mathcal{W}(p, z - y) p(y) dy \\ &= z \mu[z, 1] + \int_{\theta}^z \mathcal{W}(p, z - y) p(y) dy. \end{aligned}$$

Break the analysis into three cases, depending on the value of z .

Case 1: $z \in [0, \theta)$. In this case $\mu[z, 1] = 1$, so $\mathcal{W}(p, z) = z \leq \theta$.

Case 2: $z \in [\theta, 2\theta)$. For $y \in [\theta, z]$, we have $z - y \in [0, \theta]$, so, by Case 1, $\mathcal{W}(p, z - y) = z - y$. Using the recurrence,

$$\begin{aligned} \mathcal{W}(p, z) &= z \mu[z, 1] + \int_{\theta}^z (z - y) p(y) dy \\ &= z \left(1 - \int_{\theta}^z p(y) dy \right) + \int_{\theta}^z (z - y) p(y) dy \\ &= z - z + \theta = \theta. \end{aligned}$$

Case 3: $z \in [2\theta, 1)$. For $y \in [\theta, z]$, we have $z - y \in [0, 2\theta]$, so, by Cases 1 and 2 and the recurrence,

$$\begin{aligned} \mathcal{W}(p, z) &= z\mu[z, 1] + \int_{\theta}^{z-\theta} \theta p(y)dy + \int_{z-\theta}^z (z-y)p(y)dy \\ &= z - z \int_{\theta}^{z-\theta} p(y)dy + \int_{\theta}^{z-\theta} \theta p(y)dy \\ &\quad - \int_{z-\theta}^z yp(y)dy \\ &= z - (z - \theta) \ln \frac{z - \theta}{\theta} - \int_{z-\theta}^z yp(y)dy \\ &= z - (z - \theta) \ln \frac{z - \theta}{\theta} - \int_{z-\theta}^z dy \\ &\quad + \int_{2\theta}^z \ln((y - \theta)/\theta)dy \\ &= (z - \theta) \left(1 - \ln \frac{z - \theta}{\theta} \right) + \int_{2\theta}^z \ln((y - \theta)/\theta)dy \\ &= (z - \theta) \left(1 - \ln \frac{z - \theta}{\theta} \right) + (y - \theta) \\ &\quad \cdot (\ln((y - \theta)/\theta) - 1) \Big|_{2\theta}^z \\ &= (z - \theta) \left(1 - \ln \frac{z - \theta}{\theta} + \ln \frac{z - \theta}{\theta} - 1 \right) + \theta = \theta. \end{aligned}$$

Thus, in all cases, $\mathcal{W}(p, z) \leq \theta$, completing the proof. □

Theorem 1 *JRP-D has a randomized polynomial-time 1.574-approximation algorithm, and the integrality gap of the LP relaxation is at most 1.574.*

Proof By Lemma 4, for any fractional solution x , Algorithm Round_p (using the probability distribution p from that lemma) returns a feasible schedule of expected cost at most 1.574 times $\text{cost}(x)$.

To see that the schedule can be computed in polynomial time, note first that the LP relaxation can be solved in polynomial time. The optimal solution x is minimal, so each x_t is at most 1, which implies that $\sigma(U) = \sum_t x_t$ is at most the number of demands, n . In Algorithm Round_p, each sample from the distribution p from Lemma 4 can be drawn in polynomial time. Each sample is $\Omega(1)$, and the sum of the samples is at most $\sigma(U) \leq n$, so the number of samples is $O(n)$. In the inner loop of the algorithm (starting at line 3), for each retailer ρ , the subset of orders joined can be computed in time $O(n)$, by amortization, so the total time for this step is $O(mn)$, where m is the number of retailers. □

3 Upper bound of 1.5 for equal-length periods

In this section, we present a 1.5-approximation algorithm for the case where all the demand periods are of equal length. In

this section, release times and deadlines are arbitrary rational numbers, and all demand periods have length 1.

Denote the input instance by \mathcal{I} . Let the *width* of the instance be the difference between the maximum deadline and the minimum release time. The building block of our approach is an algorithm that creates an optimal solution to an instance of width at most 3. Later, we divide \mathcal{I} into overlapping sub-instances of width 3, solve each of them optimally, and show that aggregating their solutions gives a 1.5-approximation.

Lemma 5 *A solution to any instance \mathcal{J} of width at most 3 consisting of unit-length demand periods can be computed in polynomial time.*

Proof Shift all demands in time, so that \mathcal{J} is entirely contained in interval $[0, 3]$. Recall that C is the warehouse ordering cost and c_ρ is the ordering cost of retailer $\rho \in [m]$. Without loss of generality, assume that $m \geq 1$ and each retailer has at least one demand.

Let d_{\min} be the first deadline of a demand from \mathcal{J} and r_{\max} the last release time. If $r_{\max} \leq d_{\min}$, then placing one order at any time from $[r_{\max}, d_{\min}]$ is sufficient. Its cost is then equal to $C + \sum_\rho c_\rho$, which is clearly equal to the optimum value in this case.

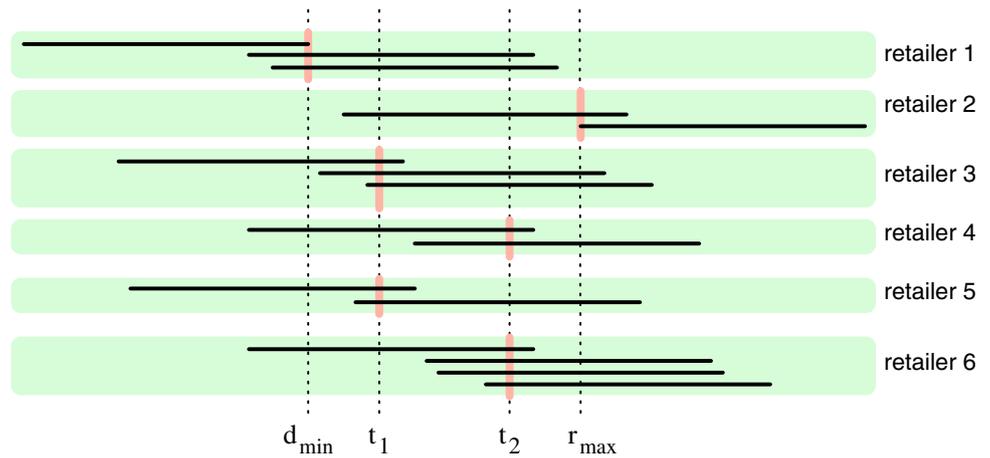
Now focus on the case $d_{\min} < r_{\max}$. Any feasible solution has to place an order at or before d_{\min} and at or after r_{\max} . Furthermore, by shifting these orders, assume that the first and last orders occur exactly at times d_{\min} and r_{\max} , respectively (Fig. 5).

The problem is thus to choose a set T of warehouse ordering times that contains d_{\min}, r_{\max} , and possibly other times from the interval (d_{\min}, r_{\max}) , and then to decide, for each retailer ρ , which warehouse orders it joins. Note that $r_{\max} - d_{\min} \leq 1$, and, therefore, each demand period contains d_{\min}, r_{\max} , or both. Hence, all demands of a retailer ρ can be satisfied by joining the warehouse orders at times d_{\min} and r_{\max} at additional cost of $2c_\rho$. It is possible to reduce the retailer ordering cost to c_ρ if (and only if) there is a warehouse order that occurs within \mathcal{D}_ρ , where \mathcal{D}_ρ is the intersection of all demand periods of retailer ρ . (To this end, \mathcal{D}_ρ has to be non-empty.)

Hence, the optimal cost for \mathcal{J} can be expressed as the sum of four parts:

- (i) the unavoidable ordering cost c_ρ for each retailer ρ ,
- (ii) the additional ordering cost c_ρ for each retailer ρ with empty \mathcal{D}_ρ ,
- (iii) the total warehouse ordering cost $C \cdot |T|$, and
- (iv) the additional ordering cost c_ρ for each retailer ρ whose \mathcal{D}_ρ is non-empty and does not contain any ordering time from T .

Fig. 5 An example of an instance and a schedule. Dashed vertical lines represent warehouse orders, with thick segments indicating which retailers join these orders. For example, retailer 1 joins the order at time d_{\min} , retailer 2 joins the order at time r_{\max} , and retailer 3 joins the order at time t_1



As the first two parts of the cost are independent of T , focus on minimizing the sum of parts (iii) and (iv), which we call the *adjusted cost*. Let $AC(t)$ be the minimum possible adjusted cost associated with the interval $[d_{\min}, t]$ under the assumption that there is an order at time t . Formally, $AC(t)$ is the minimum, over all choices of sets $T \subseteq [d_{\min}, t]$ that contain d_{\min} and t , of $C \cdot |T| + \sum_{\rho \in Q(T)} c_{\rho}$, where $Q(T)$ is the set of retailers ρ for which $\mathcal{D}_{\rho} \neq \emptyset$ and $\mathcal{D}_{\rho} \subseteq [0, t] - T$. (Note that the second term consists of expenditures that actually occur outside the interval $[d_{\min}, t]$.)

As there are no \mathcal{D}_{ρ} 's strictly to the left of d_{\min} , $AC(d_{\min}) = C$. Furthermore, $AC(t)$ for any $t \in (d_{\min}, r_{\max}]$ can be expressed recursively using the value of $AC(u)$, where $u \in [d_{\min}, t)$ is the warehouse-order time immediately preceding t in the set T that realizes $AC(t)$:

$$AC(t) = C + \min_{u \in [d_{\min}, t)} \left(AC(u) + \sum_{\rho: \emptyset \neq \mathcal{D}_{\rho} \subset (u, t)} c_{\rho} \right).$$

The second term inside the minimum represents the cost of retailers whose sets \mathcal{D}_{ρ} do not contain an order. The minimum in the definition of $AC(t)$ is determined by a u that is the deadline of some demand. Restricting attention to t 's and u 's that are deadlines of the demands, compute the relevant values of function $AC(\cdot)$ by dynamic programming in polynomial time. Finally, the total adjusted cost is $AC(r_{\max})$. The actual orders can be recovered by a standard extension of the dynamic program. \square

We now show how to construct an approximate solution for the original instance \mathcal{I} consisting of unit-length demand periods. For $i \in \mathbb{N}$, let \mathcal{I}_i be the sub-instance containing all demands entirely contained in $[i, i + 3)$. By Lemma 5, an optimal solution for \mathcal{I}_i , denoted $A(\mathcal{I}_i)$, can be computed in polynomial time. Let S_0 be the solution created by aggregating $A(\mathcal{I}_0), A(\mathcal{I}_2), A(\mathcal{I}_4), \dots$ and S_1 by aggregating $A(\mathcal{I}_1), A(\mathcal{I}_3), A(\mathcal{I}_5), \dots$. Among solutions S_0 and S_1 , output the one with the smaller cost.

Theorem 2 *The above algorithm produces a feasible schedule of cost at most 1.5 times optimum.*

Proof Each unit-length demand of instance \mathcal{I} is entirely contained in some \mathcal{I}_{2k} for some $k \in \mathbb{N}$. Hence, it is satisfied in $A(\mathcal{I}_{2k})$, and thus also in S_0 , which yields the feasibility of S_0 . An analogous argument shows the feasibility of S_1 .

To estimate the approximation ratio, fix an optimal solution OPT for instance \mathcal{I} and let opt_i be the cost of OPT's orders in the interval $[i, i + 1)$. Note that OPT's orders in $[i, i + 3)$ satisfy all demands contained entirely in $[i, i + 3)$. Since $A(\mathcal{I}_i)$ is an optimal solution for these demands, we have $\text{cost}(A(\mathcal{I}_i)) \leq opt_i + opt_{i+1} + opt_{i+2}$ and, by taking the sum, $\text{cost}(S_0) + \text{cost}(S_1) \leq \sum_i \text{cost}(A(\mathcal{I}_i)) \leq 3 \cdot \text{cost}(\text{OPT})$. Therefore, at least one of the two solutions (S_0 or S_1) has cost not larger than $1.5 \cdot \text{cost}(\text{OPT})$. \square

4 Lower bounds of 1.207 and 1.245

In this section, we prove the following lower bound on the integrality gap of the LP relaxation from Sect. 1:

Theorem 3 *The integrality gap of the LP relaxation is at least $\frac{1}{2}(1 + \sqrt{2}) \geq 1.207$.*

We then sketch a computer-assisted proof of a stronger lower bound: 1.245.

Fix an arbitrarily large integer U . Define universe $\mathcal{U} = \{i/U : i \in \mathbb{N}\} \cap [0, U]$ to contain the non-negative integer multiples of $1/U$ in the interval $[0, U]$. (The restriction to multiples of $1/U$ is a technicality; throughout, for intuition, one can consider instead $\mathcal{U} = [0, U]$.) Consider an instance with warehouse-order cost $C = 1$ and two retailers, where retailer 1 has order cost $c_1 = 0$ and retailer 2 has order cost $c_2 = \sqrt{2} + \epsilon$, where c_2 is a multiple of $1/U$ and $0 \leq \epsilon < 1/U$. Retailer 1 has a demand for every time interval of length 1; retailer 2 has a demand for every time interval of length c_2 :

$$\mathcal{D} = \{(1, t, t + 1) : t, t + 1 \in \mathcal{U}\} \\ \cup \{(2, t, t + c_2) : t, t + c_2 \in \mathcal{U}\}.$$

Intuitively, in any solution, retailer 1 must join at least one order in every subinterval of length 1, so the warehouse-order cost is at least 1 per time unit. Retailer 2 must join at least one order in any subinterval of length c_2 , so his order cost (not including the warehouse-order cost) is at least c_2 for every c_2 time units, i.e., 1 per time unit. Thus, even if the two retailers could coordinate orders perfectly, the total cost would be at least 2 per time unit.

We show next that, because the orders cannot be coordinated perfectly, the cost of any solution is at least about $1 + c_2 \approx 1 + \sqrt{2} > 2$ per time unit.

Throughout this section, $o(1)$ denotes a term that tends to zero as U tends to infinity.

Lemma 6 *For the above instance, the optimal cost is at least $(1 + \sqrt{2} - o(1))U$.*

Proof Fix any schedule for the instance. Partition the time interval $[0, U]$ into half-open subintervals, separated by the times of orders that retailer 2 joins. Consider any such subinterval $(t, t']$. That is, retailer 2 joins an order at time t (or $t = 0$), and, during the subinterval $(t, t']$ retailer 2 joins an order at time t' and no other time. We argue that the cost per time unit during $(t, t']$ is at least $1 + \sqrt{2} - o(1)$.

First consider the case that the schedule has an order during (t, t') . The order at time t' costs $1 + c_2$; the additional order during (t, t') costs at least 1. The interval length $t' - t$ is at most $c_2 + 1/U$ (otherwise it would contain a demand of retailer 2, which, by the choice of t and t' , would be unsatisfied). Thus, the cost per unit time is at least $(1 + 1 + c_2)/(c_2 + 1/U) = 1 + \sqrt{2} - o(1)$.

In the remaining case there is no order during (t, t') . The interval length $t' - t$ is at most $1 + 1/U$ (otherwise the interval would contain an unsatisfied demand of retailer 1). The order at time t' costs $1 + c_2$. Thus, the cost per time unit is at least $(1 + c_2)/(1 + 1/U) = 1 + \sqrt{2} - o(1)$.

The last subinterval $(t, t']$ has to end at time $U - c_2$ or later, so, in each subinterval of $[0, U - c_2] = [0, (1 - o(1))U]$ the algorithm pays at least $1 + \sqrt{2} - o(1)$ per time unit. \square

Next we observe that there is a fractional LP solution x that costs 2 per time unit: for each $t \in \mathcal{U}$, let $x_t = x_t^1 = 1/U$ and $x_t^2 = 1/(c_2U)$.

Recall that \mathcal{U} contains the integer multiples of $1/U$ in $[0, U]$. By calculation, the LP solution is feasible. (For each demand of retailer 1, the demand period intersects \mathcal{U} in $U + 1$ times, and so is satisfied. Likewise, for each demand of retailer 2, the demand period intersects \mathcal{U} in $c_2U + 1$ times, and so is satisfied.)

Since $|\mathcal{U}| = U^2 + 1$, the cost of fractional solution x is $2(U^2 + 1)/U = (2 + o(1))U$. By Lemma 6, any integer

solution has cost $(1 + \sqrt{2} - o(1))U$. Since the term $o(1)$ can be made arbitrarily small by choosing U large, the integrality gap of the LP is at least $(1 + \sqrt{2})/2$, proving Theorem 3.

Increasing the lower bound by a computer-based proof

We now sketch how to increase the lower bound slightly to 1.245. We reduce the problem to that of maximizing the minimum mean cycle in a finite configuration graph, which we solve with the help of linear programming.

Let the universe be $\mathcal{U} = [U]$, where U is an arbitrarily large integer (tending to infinity). Fix a vector $\delta \in \mathbb{N}_+^m$. (Later we choose $m = 5$ and $\delta = (6, 7, 8, 9, 11)$.) Focus on instances where, for each retailer $\rho \in [m]$, the retailer has uniform demands—one for every subinterval of length δ_ρ . That is, the demand set \mathcal{D} is $\mathcal{D} = \{(\rho, t, t + \delta_\rho - 1) : \rho \in [m]; \{t, t + \delta_\rho - 1\} \subseteq \mathcal{U}\}$.

Define the “uniform” fractional solution x by $x_t^\rho = 1/\delta_\rho$ for all $\rho \in [m]$ and $x_t = \max_\rho 1/\delta_\rho$ for all $t \in \mathcal{U}$. This solution is feasible for the LP and has cost $(C/\min_\rho \delta_\rho + \sum_\rho c_\rho/\delta_\rho) \cdot U$.

To bound the integer schedules we use a configuration graph. Given any feasible schedule, for each order in the schedule, define the *configuration* at the order time, t , to be a vector $\sigma \in \mathbb{N}_+^m$ where σ_ρ is the time elapsed since ρ last joined an order, up to and including time t . (If retailer ρ has not yet joined any order by time t , take $\sigma_\rho = t$.) Feasibility of the schedule ensures that each configuration σ satisfies $\sigma_\rho < \delta_\rho$ for all $\rho \in [m]$, because otherwise one of retailer ρ 's demands would not be met. Thus, there are at most $\prod_\rho \delta_\rho$ distinct configurations. These are the nodes of the configuration graph.

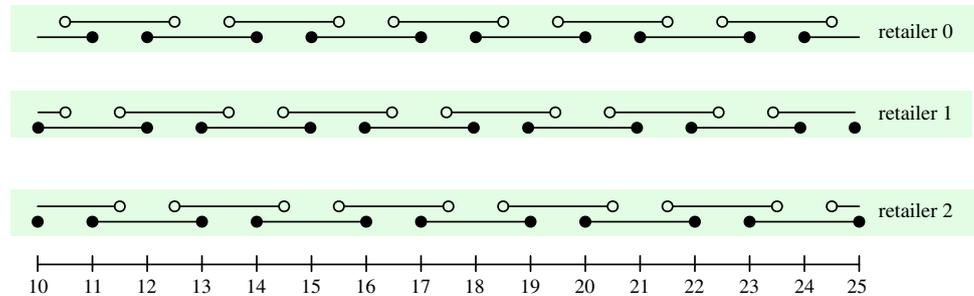
The edges of the graph model possible transitions from one order to the next. Let σ denote the configuration at some order time t . Let σ' denote the next configuration, at the next order time $t' > t$. Let R be the set of retailers in the order at time t' . Then $\sigma'_\rho = 0$ if $\rho \in R$ and $\sigma'_\rho = \sigma_\rho + t' - t$ otherwise. To ensure feasibility, for all $\rho \in [m]$, it must be that $\sigma_\rho + t' - t \leq \delta_\rho$ (even for $\rho \in R$). Without loss of generality, assume that t' is maximal subject to this constraint (otherwise, delay the second order without increasing the schedule cost). That is, $t' = t + \Delta(\sigma)$, where $\Delta(\sigma) = \min\{\delta_\rho - \sigma_\rho : \rho \in [m]\}$. For each σ and σ' that relate as described above, the configuration graph has a directed edge from σ to σ' . The *cost* of the edge is the cost of the corresponding order, $\text{cost}(\sigma, \sigma') = C + \sum_{\rho \in R} c_\rho$. Let $G = (V, E)$ be the subgraph induced by nodes reachable from the start configuration $(0, \dots, 0)$. Explicitly construct the graph G , labeling each edge (σ, σ') with its elapsed time $\Delta(\sigma)$, order set $R(\sigma, \sigma')$, and cost function $\text{cost}(\sigma, \sigma')$.

In the limit (as $U \rightarrow \infty$), every schedule will incur cost at least λ per time unit as long as, for every cycle C in this graph, the sum of the costs of the edges in C is at least λ times the sum of the times elapsed on the edges in C . The

Fig. 6 A linear program to choose the costs to maximize the integrality gap λ , given the configuration graph G and demand durations δ_ρ

$$\begin{aligned} &\text{Given } G = (V, E) \text{ and } \delta, \text{ choose } c, C, \text{cost}(), \text{ and } \Phi \text{ to maximize } \lambda \text{ subject to} \\ &C \geq 0 \\ &c_\rho \geq 0 \quad \forall \rho \in \{1, \dots, m\} \\ &C / \min_\rho \delta_\rho + \sum_\rho c_\rho / \delta_\rho \leq 1 \tag{6} \\ &\text{cost}(\sigma, \sigma') = C + \sum_{\rho \in R(\sigma, \sigma')} c_\rho \quad \forall (\sigma, \sigma') \in E \tag{7} \\ &\Phi_\sigma + \text{cost}(\sigma, \sigma') - \Phi_{\sigma'} \geq \Delta(\sigma) \lambda \quad \forall (\sigma, \sigma') \in E. \tag{8} \end{aligned}$$

Fig. 7 The demand periods in \mathcal{D}



integrality gap is then at least λ divided by the cost (per time unit) of the uniform fractional solution defined above. Note that λ is essentially the minimum mean cycle cost in G .

Given any fixed m and vector δ of period durations, the configuration graph is determined. We will choose the costs (the warehouse-order cost C and each retailer cost c_ρ) to maximize the resulting value of λ , subject to the constraint that the cost of the uniform fractional solution is at most 1. The linear program (LP) in Fig. 6 does this. The LP is based on the standard LP dual for minimum mean cycle, but the edge costs are not determined—they are chosen subject to appropriate side constraints. Constraint (6) of the LP is that the uniform fractional solution costs at most 1 per time unit.

We implemented this construction and, for various manually selected duration vectors δ with small m , we solved the linear program to find the maximum λ . For efficiency, we used the following observations to prune the configuration graph. We ordered δ so that $\delta_1 = \min_\rho \delta_\rho$. Without loss of generality we constrained c_1 to be 0 (otherwise replace C by $C + c_1$ and c_1 by 0; by inspection this gives an equivalent LP). Then, since $c_1 = 0$, without loss of generality, we assumed that retailer 1 is in every order $R(\sigma, \sigma')$. We pruned the graph further using similar elementary heuristics.

The best ratio we found was for $\delta = (6, 7, 8, 9, 11)$. The pruned graph G had about two thousand vertices. C was about 2.49, c_1 was 0, every other c_ρ was about 1.245, and λ was just above 1.245.

5 Lower bound of 1.2 for equal-length periods

In this section, we show an integrality gap for the linear program for JRP-D for instances with equal-length demand peri-

ods. The gap is for an instance with three retailers. Numbering them for convenience starting from 0, their order costs are $c_0 = c_1 = c_2 = \frac{1}{3}$. The warehouse-order cost is $C = 1$.

In the demand set \mathcal{D} , all intervals (demand periods) have length 2. Choose some large constant U that is a multiple of 3. As illustrated in Fig. 7, for $\rho = 0, 1, 2$, retailer ρ 's demand periods are

$$[3i + \rho, 3i + 2 + \rho] \quad \text{and} \quad (3i - \frac{3}{2} + \rho, 3i + \frac{1}{2} + \rho),$$

for $i = 0, \dots, U/3$.

To simplify the presentation, allow the demand periods to be either closed or open intervals. (This is only for convenience: to “close” any open interval, replace it by a closed interval slightly shifted to the right, with the shifts increasing over time. Specifically, replace each interval $(a, a + 2)$ by the interval $[a + a/U^2, a + a/U^2 + 2]$. This preserves the intersection pattern of all intervals; in particular any two intervals $(a, a + 2)$ and $(a + 2, a + 4)$ will remain disjoint after this change. Therefore, this change does not affect the values of the optimal fractional and integral solutions described below.)

This instance admits a fractional solution x whose cost is $\frac{5}{6}U + O(1)$: For each integer time t , place a $\frac{1}{2}$ -order that is joined by two retailers: the retailer $t \bmod 3$ whose closed interval starts at t , and the retailer $(t + 1) \bmod 3$ whose closed interval ends at t (let $s = (t - 1) \bmod 3$; then $x_t^s = 0$, while $x_t = x_t^\rho = \frac{1}{2}$ for $\rho \neq s$). The cost of the $\frac{1}{2}$ -order at each time t is $\frac{1}{2}(2 \cdot \frac{1}{3} + 1) = \frac{5}{6}$.

Now consider any integer solution \hat{x} . Without loss of generality, assume that \hat{x} places orders only at integer times. (Any order placed at a fractional time τ can be shifted either left or right to the first integer without changing the set of demands served.)

If a retailer ρ has an order at time t , then its next order must be at $t + 1, t + 2$ or $t + 3$, because for any t the interval $(t, t + 4)$ contains a demand period of retailer ρ . Thus, each retailer ρ has to join some order in each triple $\{t + 1, t + 2, t + 3\}$. So, the retailer-cost per time unit for ρ is at least $c_\rho/3 = \frac{1}{9}$, and the total retailer-cost per time unit is at least $\frac{3}{9} = \frac{1}{3}$.

Similarly, if there is a warehouse order at time t , then the next order must be at time $t + 1$ or $t + 2$, because the interval $(t, t + 3)$ contains a demand period of some retailer. So there must be some order in each pair $\{t + 1, t + 2\}$. So the warehouse-order cost per time unit is at least $C/2 = \frac{1}{2}$.

In total, the total cost per time unit is at least $\frac{1}{3} + \frac{1}{2} = \frac{5}{6}$ (matching the cost of the fractional solution), even if the retailer orders could be coordinated perfectly with the warehouse orders. In the rest of this section, we show that, because perfect coordination is not possible, the actual cost is higher.

Recall that (without loss of generality) in \hat{x} orders occur only at integer times. For each ρ , call the endpoints of ρ 's closed intervals ρ 's *endpoint times* (these are times t with $(t - \rho) \bmod 3 \in \{0, 2\}$). Call the midpoints of ρ 's closed intervals ρ 's *inner times* (these are times t with $(t - \rho) \bmod 3 = 1$). Assume (without loss of generality by the feasibility and optimality of \hat{x}) that \hat{x} satisfies the following conditions:

- (c1) For any ρ and any pair of consecutive endpoint times $\{t, t + 1\}$ of ρ , ρ joins an order at time t or $t + 1$ (because ρ has an open interval containing only integers $\{t, t + 1\}$).
- (c2) If t is an inner time of retailer ρ and ρ joins an order at t , then
 - (c2.1) there is no order at time $t - 1$ or $t + 1$, and
 - (c2.2) all retailers have orders at time t .

(For (c2.1): if there is an order at time $t - 1$ or $t + 1$, then retailer ρ can be moved to that order from the order at time t . For (c2.2): for each retailer $\rho' \neq \rho$ both time t and either $t - 1$ or $t + 1$ are endpoint times, but per (c2.1) there is no order at $t - 1$ or at $t + 1$, so, by (c1), ρ' must have an order at t .)

The idea of the analysis is similar to the argument in Sect. 4 for the lower bound of 1.245 for general instances: represent the possible schedules by walks in a finite configuration graph.

Fix any feasible schedule. At any integer time t , the *configuration* of the schedule at time t is the 4-digit string $s\sigma_0\sigma_1\sigma_2$, where $s = t \bmod 3$ and, for each retailer $\rho = 0, 1, 2$, the elapsed time since the retailer last joined an order is σ_ρ . Since the schedule is feasible, each σ_ρ is in $\{0, 1, 2\}$, so there are at most 3^4 possible configurations.

Suppose a schedule is in configuration $s\sigma_0\sigma_1\sigma_2$ at time t , then transitions to $s'\delta_0\delta_1\delta_2$ at time $t + 1$. Necessarily $s' = (s + 1) \bmod 3$. Let R be the set of retailers (possibly empty)

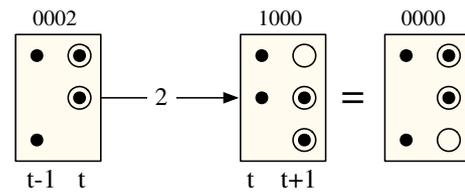


Fig. 8 Graphical representation of a transition. Configuration 0002 is on the left. At time $t + 1$ all retailers issue an order. After removing spurious orders (keeping track of only the last one), the new configuration is 1000, which is equivalent (by symmetry) to 0000. The transition costs $1 + \frac{1}{3} \cdot 3 = 2$

that join the order (if any) at time $t + 1$. For each retailer ρ , (i) if $\rho \notin R$ then $\delta_\rho = \sigma_\rho + 1$, while (ii) if $\rho \in R$ then $\delta_\rho = 0$. Say a pair $s\sigma_0\sigma_1\sigma_2 \rightarrow s'\delta_0\delta_1\delta_2$ is a *possible transition* if the pair relates this way for some R . The *cost* of the transition equals the cost of the order: 0 if $R = \emptyset$, or $1 + \frac{1}{3}|R|$ otherwise. (Here, unlike in Section 4, the elapsed time per transition is always 1, and R can be empty.)

Represent each possible configuration graphically by a rectangle with a row for each retailer $\rho = 0, 1, 2$. Each row has two cells, representing times $t - 1$ and t , respectively: a circle in the first cell means $\sigma_\rho = 1$, a circle in the second cell means $\sigma_\rho = 0$, no circle means $\sigma_\rho = 2$. A dot in the cell means that time is an endpoint time for the retailer; no dot means the time is an inner time. The dot pattern of any one row determines, and is determined by, s . Figure 8 shows an example of a single transition.

Any two configurations are *equivalent* if one can be obtained from the other by permuting the rows of the graphical representation (i.e., the retailers). Each graphical representation has one row with a dot in both columns, one row with a dot in the second column only, and one row with a dot in the first column only. Define the *canonical representative* of an equivalence class to be the configuration in which these three rows are, respectively, first, second, and third. In all such configurations, s is zero, so there are at most 3^3 equivalence classes.

Now restrict the configurations further to those that are realizable in \hat{x} , in that they do not violate conditions (c1)–(c2): by Condition (c1), if a row has two dots, then one of the dots must be circled; by Condition (c2), if a column has a dot-less circle, then all cells in the column have circles. Note that the equivalence relation respects these conditions: in a given equivalence class, either all configurations meet both conditions, or none do.

Finally, define graph G to have a node for each realizable equivalence class. For each possible transition $\sigma \rightarrow \sigma'$ between remaining configurations σ and σ' , add a directed edge in G from the equivalence class of σ to that of σ' . Give the edge cost equal to the cost of the transition. By a routine but tedious calculation, G is the 10-node graph shown in

Fig. 9 The complete transition diagram

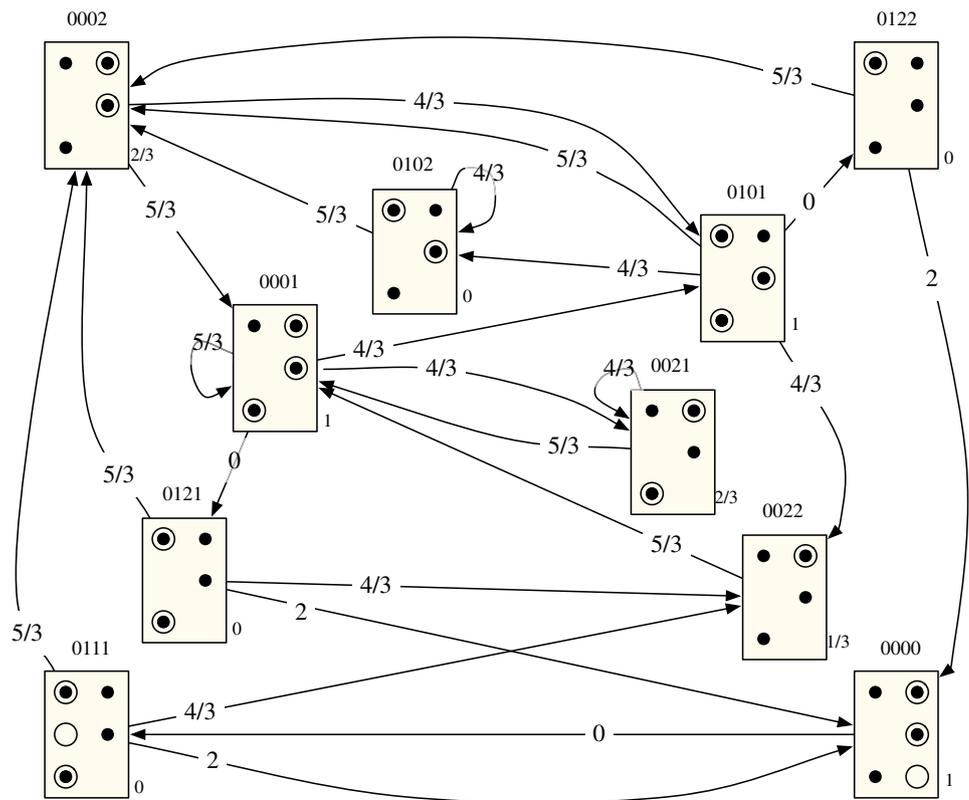


Fig. 9. Each node is represented by its canonical representative.

Next we argue that every cycle in this graph has average edge cost at least 1. Define the following potential function $\Phi(\sigma)$ on configurations:

σ	0000	0001	0002	0021	0022	0101	0102	0111	0121	0122
$\Phi(\sigma)$	1	1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	1	0	0	0	0

It is routine (if tedious) to verify that for each edge $\sigma \rightarrow \sigma'$, its cost $\text{cost}(\sigma, \sigma')$ satisfies

$$\text{cost}(\sigma, \sigma') \geq \Phi(\sigma') - \Phi(\sigma) + 1. \tag{9}$$

For any path of length U in G (summing inequality (9) along all edges on this path) the cost of the path is $U - O(1)$.

The equivalence classes of the configurations of the schedule \hat{x} (one for each time $t \in [U]$) form a path of length U in G . The cost of \hat{x} equals the cost of the path, which must be at least $U - O(1)$. Recalling that there is a fractional solution of cost $\frac{5}{6}U + O(1)$, this shows that the integrality gap is at least $\frac{6}{5} = 1.2$:

Theorem 4 *For instances with all demand periods equal, the integrality gap of the LP at least 1.2.*

(The bound in the above proof is tight: the following cycles have average edge cost 1: $0000 \rightarrow 0111 \rightarrow 0000$, $0101 \rightarrow 0122 \rightarrow 0002 \rightarrow 0101$, and $0121 \rightarrow 0022 \rightarrow 0001 \rightarrow 0121$.)

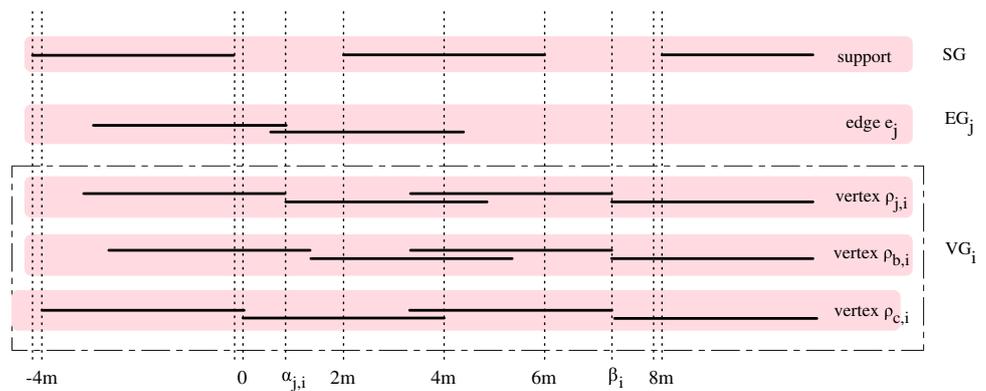
6 APX-hardness for equal-length demand periods

Let JRP-D_{E4} be the restriction of JRP-D where each retailer has at most four demands and all demand periods are of the same length. In this section, we show that JRP-D_{E4} is APX-hard by giving a PTAS-reduction from Vertex Cover in cubic graphs, that is graphs with every vertex of degree three. Vertex Cover is known to be APX-complete for such graphs (Alimonti and Kann 2000).

Roughly speaking, given any cubic graph $G = (V, E)$ with n vertices and $m (= 3n/2)$ edges, the reduction produces an instance \mathcal{J}_G of JRP-D_{E4} , such that G has a vertex cover of size K iff \mathcal{J}_G has a schedule of cost $10.5n + K + 6$. Since any vertex cover has size at least $m/3 = n/2$, this is a PTAS-reduction. Such reduction and a PTAS for \mathcal{J}_G would give a PTAS for Vertex Cover in degree-three graphs.

Construction of instance \mathcal{J}_G . Fix a given undirected cubic graph G with vertex set $V = \{0, \dots, n - 1\}$ and edges e_0, \dots, e_{m-1} . \mathcal{J}_G consists of $1 + m + n$ gadgets: one support gadget SG , an edge gadget EG_j for each edge e_j , and a vertex gadget VG_i for each vertex i . All retailer and order

Fig. 10 The construction of instance \mathcal{J}_G . The figure shows the support gadget SG, an edge gadget EG_j , and a vertex gadget VG_i for a vertex i with edges $e_j, e_b,$ and e_c . Shaded regions represent retailers. Horizontal line segments represent demand periods



costs equal 1 ($C = c_\rho = 1$) and all demand periods have length $4m$. All release times and deadlines are integers in the interval $[-4m, 12m + 1]$; without loss of generality, restrict attention to schedules with integer order times. The gadgets are as follows.

The support gadget SG SG has its own retailer, the *support-gadget retailer*, having three demands with periods $[-4m - 1, -1], [2m, 6m],$ and $[8m + 1, 12m + 1]$. These periods are separated by two *gaps* of length $2m$. Call the times $\{-1, 4m, 8m + 1\}$ *support gadget times*; orders at these times suffice to satisfy the three demands.

Edge gadgets EG_j Each edge e_j in G has its own *edge retailer*, having two demands with, respectively, periods $[2j + 1 - 4m, 2j + 1]$ and $[2j, 2j + 4m]$. These demands can be satisfied with a single order at time $2j$ or $2j + 1$. Think of these two times as being associated with this edge e_j , each associated with one endpoint of e_j (as explained below). All such times are in the first gap, $[0, 2m - 1]$.

Let $e_j = \{i, i'\}$, that is i, i' are the endpoints of edge e_j . Intuitively, to satisfy e_j 's retailer cheaply, there can be an order at time $2j$ or at $2j + 1$; this models that e_j can be covered by either of its two endpoints. We associate one of the two times $2j$ or $2j + 1$ (it does not matter which one) with i and call it $\alpha_{j,i}$, while the other one is associated with i' and, naturally, called $\alpha_{j,i'}$.

Vertex gadgets VG_i For each vertex i , define its “vertex” time $\beta_i = 8m - i$. (All such times are in the second gap, $[6m + 1, 8m]$.) Do the following for each of vertex i 's edges. Let e_j denote the edge. Add a new *vertex retailer* $\rho_{j,i}$, having four demands with respective periods

$$[\alpha_{j,i} - 4m, \alpha_{j,i}], [\alpha_{j,i}, \alpha_{j,i} + 4m], [\beta_i - 4m, \beta_i],$$

$$\text{and } [\beta_i, \beta_i + 4m].$$

Denote the periods of these four demands as $Q_{j,i}^0, Q_{j,i}^1, Q_{j,i}^2,$ and $Q_{j,i}^3$, in the above order. Note that $Q_{j,i}^0 \cap Q_{j,i}^1 = \{\alpha_{j,i}\}, Q_{j,i}^1 \cap Q_{j,i}^2 = [\beta_i - 4m, \alpha_{j,i} + 4m] \neq \emptyset, Q_{j,i}^2 \cap Q_{j,i}^3 = \{\beta_i\}$, but otherwise the four demand periods are pairwise-disjoint.

The important property is that retailer $\rho_{j,i}$'s four demands can be satisfied with two orders iff the two orders are at times $\alpha_{j,i}$ and β_i . Also, if orders do happen to be placed at these two times, then (because $\alpha_{j,i}$ is one of the two times belonging to e_j) the order at time $\alpha_{j,i}$ can satisfy both demands of edge e_j 's retailer with no additional warehouse cost for that retailer.

Figure 10 illustrates the reduction.

Lemma 7 *If G has a vertex cover U of size K , then \mathcal{J}_G has a schedule of cost at most $10.5n + K + 6$.*

Proof Let U be a vertex cover of size K .

To construct the schedule for \mathcal{J}_G , start with orders at the support-gadget times $\{-1, 4m, 8m + 1\}$, each of which is joined by the support-gadget retailer. This costs 6.

Next, consider each vertex i . If $i \notin U$, then have i 's vertex retailers (that is, retailers $\rho_{j,i}$ for $e_j \ni i$) join the support-gadget orders at times $\{-1, 4m, 8m + 1\}$. This option increases the schedule cost by $3 \cdot 3 = 9$.

Otherwise (for $i \in U$), create an order at time β_i . For each of three i 's retailers $\rho_{j,i}$ create an order at time $\alpha_{j,i}$, and have the retailer join that order and the one at time β_i . The order at time β_i is shared between i 's three retailers, so that order costs $1 + 3 = 4$. Each of the three orders created at times $\alpha_{j,i}$ costs 2. The total cost for the four orders is $3 \cdot 2 + 4 = 10$.

Next, consider each edge e_j . As U is a vertex cover, some vertex $i \in U$ covers e_j , that is $i \in U \cap e_j$. By the construction of the i 's gadget, since $i \in U$, there is already an order at e_j 's time $\alpha_{j,i}$. Have edge e_j 's retailer join this order. Both demands of this retailer will be satisfied, since they both contain $\alpha_{j,i}$. The cost increases by 1 per edge.

Adding up the above costs, the total cost is $6 + 9(n - K) + 10K + m = 10.5n + K + 6$. \square

Recovering a vertex cover from an order schedule We now show the converse: given any order schedule of cost $10.5n + K + 6$ for \mathcal{J}_G , we can compute a vertex cover of G of size K . Recall that $\alpha_{j,i}$ denotes the time (either $2j$ or $2j + 1$) that edge e_j shares with endpoint i .

Say that an order schedule S meeting the following desirable conditions is in *normal form*:

- (nf1) In S , the support-gadget retailer joins orders at the support-gadget times $\{-1, 4m, 8m + 1\}$.
- (nf2) In S , for each edge $e_j = \{i, i'\}$, the edge's retailer joins an order at time $\alpha_{j,i}$ or $\alpha_{j,i'}$.
- (nf3) For each vertex i , exactly one of the following two conditions holds:
 - (a) each of i 's retailers $\rho_{j,i}$ joins orders at times β_i and $\alpha_{j,i}$;
 - (b) each of i 's retailers $\rho_{j,i}$ joins the support-gadget orders at times $\{-1, 4m, 8m + 1\}$, and S has no order at time β_i nor at any time $\alpha_{j,i}$.
- (nf4) For each edge $e_j = \{i, i'\}$, at least one of its endpoints i, i' satisfies Condition (nf3) (a)
- (nf5) S has no orders other than the ones described above.

Given any feasible order schedule, we can put it in normal form without increasing the cost:

Lemma 8 *Given any order schedule S for \mathcal{J}_G , one can compute in polynomial time a normal-form schedule S' whose cost is at most the cost of S .*

Proof Modify S to satisfy Conditions (nf1) through (nf5) in turn, maintaining feasibility without increasing the cost, as follows.

(nf1) Combine all orders in times $(-\infty, -1]$ into a single order at time -1 . By inspection, the earliest deadline of any demand is the deadline of the first support-gadget demand, which is -1 . So this modification is *safe*—it maintains feasibility without increasing the cost—and the support-gadget retailer joins the order at time -1 .

Likewise, combine all orders in times $[8m + 1, \infty)$ into a single order at time $8m + 1$. The last release time of any demand is the release time of the last support-gadget demand, which is $8m + 1$. So this modification is also *safe* and the support-gadget retailer joins the order at time $8m + 1$.

Finally, combine all orders in times $[2m, 6m]$ into a single order at time $4m$. The support-gadget has demand period $[2m, 6m]$, so the support-gadget retailer must join at least one order at some time in $[2m, 6m]$. Thus this modification does not increase the cost. There are no deadlines in $[2m, 4m)$ and no release times in $(4m, 6m]$, so the modification maintains feasibility.

The resulting schedule satisfies Condition (nf1).

(nf2) Consider any edge $e_j = \{i, i'\}$. If the edge's retailer does not join an order at one of the times $\alpha_{j,i}$ or $\alpha_{j,i'}$ associated with e_j then, by inspection of his demands, the retailer must join at least two orders. Remove him from these two orders, reducing the cost by two, and have him join a (possibly) new order at, time, say $\alpha_{j,i}$, satisfying both his demands

and increasing the cost by two or less. The resulting schedule satisfies Conditions (nf1) and (nf2).

(nf3) Consider any vertex i . Assume first that S has an order at time β_i . If any of vertex i 's retailers, say $\rho_{j,i}$, does not join the order at time β_i , then move him from some order at any later time (there must be one in his last demand period $Q_{j,i}^3 = [\beta_i, \beta_i + 4m]$) to the existing order at time β_i . Then, if retailer $\rho_{j,i}$ does not join an order at time $\alpha_{j,i}$ (or there is no such order), he must participate in at least two orders at times other than β_i ; remove him from these two orders and have him join a (possibly new) order at time $\alpha_{j,i}$. Finally, remove the retailer from all orders other than those at times β_i and $\alpha_{j,i}$. These operations are *safe*, and now vertex i meets Condition (nf3).

In the other case S has no order at time β_i . Then each of vertex i 's retailers must join at least three orders. Remove each such retailer from all those orders and add him instead to the existing orders at the support-gadget times $\{-1, 4m, 8m + 1\}$. Note that support-gadget times are different than all times $\alpha_{j,i}$. This is *safe*, does not increase the cost, and now vertex i meets Condition (nf3).

As these operations affect only vertex retailers, Conditions (nf1) and (nf2) still hold too.

(nf4) Consider any edge $e_j = \{i, i'\}$. By Condition (nf2), there is an order at time $\alpha_{j,i}$ or $\alpha_{j,i'}$. By symmetry, we can assume that there is an order at time $\alpha_{j,i}$. If vertex i satisfies Condition (nf3) (a), we are done. Otherwise, i satisfies Condition (nf3) (b). Remove each of i 's three retailers from orders at times $-1, 4m, 8m + 1$, reducing i 's cost by 9. Then: create an order at time β_i and have them join this order (at cost 4), have retailer $\rho_{j,i}$ join the order at time $\alpha_{j,i}$ (at cost 1), and have the other two retailers $\rho_{j',i}$ and $\rho_{j'',i}$ join (possibly new) orders at times $\alpha_{j',i}$ and $\alpha_{j'',i}$ (at cost at most 2 each). The total cost of these new orders is at most 9, thus this modification does not increase the overall cost, and afterwards e_j satisfies Condition (nf4).

(nf5) Remove all retailers from orders not described above, then delete empty orders. As the orders described above satisfy all the demands, this is *safe*. Now Condition (nf5) holds as well. \square

Lemma 9 *Given an order schedule S for \mathcal{J}_G of cost $10.5n + K + 6$, one can compute in polynomial time a vertex cover of G of size K .*

Proof By Lemma 8, without loss of generality we can assume that that S is in normal form.

By Condition (nf1), the cost for the support-gadget orders (at times $\{-1, 4m, 8m + 1\}$, but not yet counting the retailer cost for any vertices) is $3 \cdot 2 = 6$.

By Condition (nf3), the cost associated with vertices is as follows. Fix any vertex i . If S has an order at time β_i then that order is joined by each of vertex i 's retailers, at cost $1 + 3 = 4$; also, each of i 's retailers joins its own order (at a time $\alpha_{j,i}$

where $e_j \ni i$), which costs 2. Thus the cost associated with vertex i is $4 + 2 \cdot 3 = 10$. Otherwise (that is, when S has no order at time β_i), each of i 's three retailers joins the three support-gadget orders, so the cost associated with i is $3 \cdot 3 = 9$. Putting it together, and letting ℓ be the number of vertices i that have an order at time β_i , the total cost associated with all vertices can be written as $10\ell + 9(n - \ell) = 9n + \ell$.

By Condition (nf2), the cost associated with edges is as follows. For each edge $e_j = \{i, i'\}$, Condition (nf4) guarantees that one of i or i' satisfies Condition (nf3)(a). By symmetry, assume that it is i . Since $\rho_{j,i}$ already has an order at $\alpha_{j,i}$, we can have retailer e_j join this order at cost 1. In total, the additional cost associated with the edge gadgets is $m = 1.5n$.

In sum, the schedule costs $6 + (9n + \ell) + 1.5n = 10.5n + \ell + 6$. Hence, $\ell = K$.

Now define U to contain the ℓ vertices i for which S makes an order at time β_i . For each edge $e_j = \{i, i'\}$, by Condition (nf4), there is an order at one of e_j 's associated times, say $\alpha_{j,i}$. By Condition (nf3), there is also an order at time β_i , so, by definition, U contains vertex i . Thus, U is a vertex cover. □

Here is the proof of APX-hardness. Recall that JRP-DE4 is JRP-D restricted to instances with equal-length demand periods and at most four demands.

Theorem 5 *JRP-DE4 is APX-hard.*

Proof Vertex Cover in cubic graphs is APX-hard (Alimonti and Kann, 2000, §3). We give a PTAS-reduction from that problem to JRP-DE4.

Given any cubic graph G with $n \geq 6$ vertices, and any $\epsilon > 0$, compute the instance \mathcal{J}_G from Lemma 7. By inspection of the proof, in \mathcal{J}_G all demand periods have equal length and (because G has degree three) each retailer has at most four demands, so \mathcal{J}_G is an instance of JRP-DE4.

Now suppose we are given any $(1 + \epsilon/24)$ -approximate solution S to \mathcal{J}_G . From S , compute a vertex cover U for G using the computation from Lemma 9. The computations of \mathcal{J}_G from G , and of U from S , can be done in time polynomial in n . To finish, we show that the vertex cover U has size at most $(1 + \epsilon)K^*$, where K^* is the size of the optimal vertex cover in G .

By Lemma 7, \mathcal{J}_G has an order schedule of cost at most $10.5n + K^* + 6$. Since G is cubic, $K^* \geq m/3 = n/2$, so $10.5n + 6 \leq 11.5n \leq 23K^*$. Thus S has cost at most

$$\begin{aligned} &(1 + \epsilon/24)(10.5n + K^* + 6) \\ &= 10.5n + K^* + 6 + (\epsilon/24)(10.5n + K^* + 6) \\ &\leq 10.5n + K^* + 6 + (\epsilon/24)(23K^* + K^*) \\ &= 10.5n + (1 + \epsilon)K^* + 6. \end{aligned}$$

Since all costs are integer, the cost of S is in fact at most $10.5n + K + 6$, where $K = \lfloor (1 + \epsilon)K^* \rfloor$. Using this bound

and Lemma 9, the vertex cover U has size at most $K \leq (1 + \epsilon)K^*$. □

Since JRP-DE4 \in APX (it has a constant-factor approximation algorithm), the theorem implies that JRP-DE4 is APX-complete.

Of course, APX-hardness implies that, unless $P = NP$, there is no PTAS for JRP-DE4: that is, for some $\delta > 0$, there is no polynomial-time $(1 + \delta)$ -approximation algorithm for JRP-DE4.

7 Final comments

The integrality gap for the standard JRP-D LP relaxation is between 1.245 and 1.574. We conjecture that neither bound is tight. Although we do not have a formal proof, we believe that our refined distribution for the tally game given here is optimal: it was optimized under the assumption that it never generates more than two samples, and allowing more than two samples, according to our calculations, can only increase the value of $\mathcal{Z}(p)$. Thus improving the upper bound will likely require a different approach.

There is a simple algorithm for JRP-D that provides a $(1, 2)$ -approximation, meaning that its warehouse-order cost is not larger than that in the optimum, while its retailer order cost is at most twice that in the optimum (Nonner and Souza 2009). One can combine that algorithm and the one here by choosing each algorithm with a certain probability. This simple approach does not improve the approximation ratio, but it may be possible to do so if, instead of using the algorithm presented here, one appropriately adjusts the probability distribution.

The computational complexity of general JRP-D, as a function of the maximum number p of demand periods of each retailer, is essentially resolved: for $p \geq 3$ the problem is APX-hard (Nonner and Souza 2009), while for $p \leq 2$, it can be solved in polynomial time (for $p = 1$ it can be solved with a greedy algorithm; for $p = 2$ one can apply a dynamic programming algorithm similar to that used in the proof of Lemma 5). For the case of equal-length demand periods, we showed that the problem remains APX-hard for $p \geq 4$. It would be nice to settle the case $p = 3$, which remains open. We conjecture that this case is also NP-complete.

Finally, we note that any LP-based algorithm for JRP-D can be used as a building block for general JRP (with arbitrary waiting costs) (Bienkowski et al. 2014). The construction combines One-Sided Retailer Push and Two-Sided Retailer Push algorithms (Levi et al. 2008) with an appropriately crafted and scaled instance of JRP-D. By plugging our 1.574-approximation to solve the JRP-D instance, the algorithm of

Bienkowski et al. (2014) yields a 1.791-approximation for JRP.

Acknowledgments We would like to thank Łukasz Jeż, Dorian Nogneng, Jiří Sgall, and Grzegorz Stachowiak for stimulating discussions and useful comments. We are also grateful to anonymous reviewers of earlier versions of this manuscript for pointing out several mistakes and suggestions for improving the presentation. A preliminary version of this work appeared in the Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP'13). Research supported by NSF Grants CCF-1217314, CCF-1117954, OISE-1157129; EPSRC grants EP/J021814/1 and EP/D063191/1; FP7 Marie Curie Career Integration Grant; Royal Society Wolfson Research Merit Award; and Polish National Science Centre Grant DEC-2013/09/B/ST6/01538.

Appendix: Wald's lemma

Here is the variant of Wald's Lemma (also known as Wald's identity, and a consequence of standard "optional stopping" theorems) that we use in Sect. 2. The proof is standard; we present it for completeness.

Lemma 10 (Wald's Lemma) *Consider a random experiment that, starting from a fixed start state S_0 , produces a random sequence of states S_1, S_2, S_3, \dots . Let random index $T \in \{0, 1, 2, \dots\}$ be a stopping time for the sequence (that is, for each positive integer t , the event " $T \leq t$ " is determined by state S_t). Let function $\phi : \{S_t\} \rightarrow \mathbb{R}$ map the states to \mathbb{R} . Suppose that, for some fixed constants ξ and F ,*

- (i) $(\forall t < T) \mathbf{E}[\phi(S_{t+1}) | S_t] \geq \phi(S_t) + \xi$,
- (ii) either $(\forall t < T) \phi(S_{t+1}) - \phi(S_t) \geq F$ in all outcomes, or $(\forall t < T) \phi(S_{t+1}) - \phi(S_t) \leq F$ in all outcomes, and
- (iii) T has finite expectation.

Then, $\mathbf{E}[\phi(S_T)] \geq \phi(S_0) + \xi \mathbf{E}[T]$.

Proof For each $t \geq 0$, define random variable $\delta_t = \phi(S_{t+1}) - \phi(S_t)$. By assumption (i), $\mathbf{E}[\delta_t | S_t] \geq \xi$ for $t < T$. Since the event " $T > t$ " is determined by S_t , this implies that $\mathbf{E}[\delta_t | T > t] \geq \xi$. Then the inequality in the lemma can be derived as follows:

$$\begin{aligned} \mathbf{E}[\phi(S_T) - \phi(S_0)] &= \mathbf{E}[\sum_{t < T} \delta_t] \\ &= \sum_{\tau \geq 0} \Pr[T = \tau] \cdot \mathbf{E}[\sum_{t < \tau} \delta_t | T = \tau] \\ &= \sum_{\tau \geq 0} \sum_{t < \tau} \Pr[T = \tau] \cdot \mathbf{E}[\delta_t | T = \tau] \\ &= \sum_{t \geq 0} \sum_{\tau > t} \Pr[T = \tau] \cdot \mathbf{E}[\delta_t | T = \tau] \\ &= \sum_{t \geq 0} \Pr[T > t] \cdot \mathbf{E}[\delta_t | T > t] \\ &\geq \sum_{t \geq 0} \Pr[T > t] \cdot \xi \\ &= \xi \mathbf{E}[T], \end{aligned}$$

Exchanging the order of summation in the third step above does not change the value of the sum, because (by assumptions (ii) and (iii)) either the sum of all negative terms is at least $\sum_{\tau \geq 0} \sum_{t < \tau} \Pr[T = \tau] F = F \sum_{\tau \geq 0} \tau \Pr[T = \tau] = F \mathbf{E}[T]$, which is finite, or (likewise) the sum of all positive terms is finite.

Each application in Sect. 2 has $\xi \geq \mathcal{Z}(p) > 0$ and $\phi(S_T) - \phi(S_0) \leq U$ for some fixed U . In this case Wald's Lemma implies $\mathbf{E}[T] \leq U/\xi \leq U/\mathcal{Z}(p)$.

References

- Alimonti, P., & Kann, V. (2000). Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1–2), 123–134.
- Arkin, E., Joneja, D., & Roundy, R. (1989). Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8(2), 61–66.
- Becchetti, L., Marchetti-Spaccamela, A., Vitaletti, A., Korteweg, P., Skutella, M., & Stougie, L. (2009). Latency-constrained aggregation in sensor networks. *ACM Transactions on Algorithms*, 6(1), 13:1–13:20.
- Bienkowski, M., Byrka, J., Chrobak, M., Jeż, Ł., & Sgall, J. (2014). Better approximation bounds for the joint replenishment problem. In: *Proceedings of the of the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 42–54.
- Brito, C., Koutsoupias, E., & Vaya, S. (2012). Competitive analysis of organization networks or multicast acknowledgement: How much to wait? *Algorithmica*, 64(4), 584–605.
- Buchbinder, N., Kimbrel, T., Levi, R., Makarychev, K., & Sviridenko, M. (2008). Online make-to-order joint replenishment model: Primal dual competitive algorithms. In: *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 952–961.
- Khanna, S., Naor, J., & Raz, D. (2002). Control message aggregation in group communication protocols. In: *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 135–146.
- Levi, R., Roundy, R., & Shmoys, D.B. (2005). A constant approximation algorithm for the one-warehouse multi-retailer problem. In: *Proceedings of the Sixteenth Annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pp. 365–374.
- Levi, R., Roundy, R., & Shmoys, D. B. (2006). Primal-dual algorithms for deterministic inventory problems. *Mathematics of Operations Research*, 31(2), 267–284.
- Levi, R., Roundy, R., Shmoys, D. B., & Sviridenko, M. (2008). A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Science*, 54(4), 763–776.
- Levi, R., & Sviridenko, M. (2006). Improved approximation algorithm for the one-warehouse multi-retailer problem. In: *Proceedings of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pp. 188–199.
- Nonner, T., & Souza, A. (2009). Approximating the joint replenishment problem with deadlines. *Discrete Mathematics, Algorithms and Applications*, 1(2), 153–174.