Better Approximation Bounds for the Joint Replenishment Problem*

Marcin Bienkowski[†] Jaroslaw Byrka[†] Marek Chrobak[‡] Łukasz Jeż[†,§]
Dorian Nogneng[¶] Jiří Sgall[‖]

Abstract

The Joint Replenishment Problem (JRP) deals with optimizing shipments of goods from a supplier to retailers through a shared warehouse. Each shipment involves transporting goods from the supplier to the warehouse, at a fixed cost C, followed by a redistribution of these goods from the warehouse to the retailers that ordered them, where transporting goods to a retailer ρ has a fixed cost c_ρ. In addition, we incur waiting costs for each order, possibly an arbitrary non-decreasing function of time, different for each order. The objective is to minimize the overall cost of satisfying all orders, namely the sum of all shipping and waiting costs.

JRP has been well studied in Operations Research and, more recently, in the area of approximation algorithms. For arbitrary waiting cost functions, the best known approximation ratio is 1.8. This ratio can be reduced to ≈ 1.574 for the JRP-D model, where there is no cost for waiting but orders have deadlines. As for hardness results, it is known that the problem is $\mathbb{APX}$-hard and that the natural linear program for JRP has integrality gap at least 1.245. Both results hold even for JRP-D. In the online scenario, the best lower and upper bounds on the competitive ratio are 2.64 and 3, respectively. The lower bound of 2.64 applies even to the restricted version of JRP, denoted JRP-L, where the waiting cost function is linear.

We provide several new approximation results for JRP. In the offline case, we give an algorithm with ratio ≈ 1.791, breaking the barrier of 1.8. We also show that the integrality gap of the linear program for JRP-L is at least $12/11 \approx 1.09$. In the online case, we show a lower bound of ≈ 2.754 on the competitive ratio for JRP-L (and thus JRP as well), improving the previous bound of 2.64. We also study the online version of JRP-D, for which we prove that the optimal competitive ratio is 2.

1 Introduction

The Joint Replenishment Problem (JRP) deals with optimizing shipments of goods from a supplier to a set $\mathcal{R}$ of retailers through a shared warehouse. Over time, retailers issue orders for items. All ordered items must be subsequently shipped to the retailers, although some shipments can be delayed, in order to aggregate orders into fewer shipments to reduce cost.

Specifically, for each $\rho \in \mathcal{R}$ we are given the cost c_ρ of transporting goods from the warehouse to ρ. We are also given the cost C of transporting goods from the supplier to the warehouse. A shipment of goods from the supplier to a subset $S \subseteq \mathcal{R}$ of retailers involves first shipping them to the warehouse and then redistributing them to all retailers in S, at cost equal $\mathsf{C} + \sum_{\rho \in S} \mathsf{c}_\rho$. Note that this cost is independent of the set of items shipped. The waiting cost of an item π ordered at time a and delivered at time $t \geq a$ is given by a function $h(t)$, possibly dependent on π, where we assume that the values of $h(t)$ are non-decreasing with t. The objective is to minimize the overall cost of satisfying all orders, namely the total cost of shipments plus the total waiting cost of all orders.

There are two natural restrictions on waiting costs that have been previously considered in the literature. One is to assume that the waiting costs are linear, that is for any order $\pi = (\rho, a, h)$ the cost function is $h(t) = t - a$ for $t \geq a$. We denote this version by JRP-L. In the other version, called JRP with deadlines (JRP-D), there is no waiting cost but ordered items must be shipped before pre-specified deadlines; this can be modeled by a threshold function $h(t)$.

1.1 Previous Work.

Several different, but mathematically equivalent, definitions of JRP can be found in the literature. In earlier papers, JRP is typically

*Research partially supported by NSF grants CCF-1217314 and OISE-1157129, MNiSW grant no. N N206 368839, 2010-2013, FNP Start scholarship , EU ERC project 259515 PAAl, and grant CE-ITI (project P202/12/G061 of GA ČR).

[†]Institute of Computer Science, University of Wrocław, Poland
[‡]Department of Computer Science, University of California at Riverside, USA
[§]Department of Computer, Control, and Management Engineering, Sapienza University of Rome, Italy
[¶]LIX, École Polytechnique, Palaiseau, France
[‖]Computer Science Institute, Faculty of Mathematics and Physics, Charles University, Czech Republic

phrased as an inventory management problem, where the inventory of some commodity needs to meet a set of demands that arrive over time. The objective is to balance the cost of orders that replenish the inventory and the cost of maintaining the inventory, referred to as the holding cost. (Note that in this formulation the meaning of the term "order" is different from our usage and it corresponds to what we call a shipment; while the holding cost corresponds to our waiting cost.) This formulation would not quite make sense in the online scenario, since the orders that need to be scheduled take place before demands. An online model of JRP, referred to as *Make-to-Order JRP*, was introduced by Buchbinder *et al.* [BKL+08]. In their description there is no inventory; instead, a collection of demands must be satisfied by subsequent orders. Except for minor terminology variations, our definition is essentially the same as that in [BKL+08]. Some of recent papers [BMSV+09, BKV12, KNR02, BBC+13b] on control message aggregation in networks, introduce a model where control packets (corresponding to orders, in our definition) need to be transmitted to a common destination (corresponding to the supplier), paying the transmission and delay costs. In particular, the flat-tree case studied in [BKV12] is equivalent to JRP-L.

JRP has been well studied in Operations Research and, more recently, also in the area of approximation algorithms. The problem is known to be strongly $\mathbb{NP}$-hard, even for the special cases of JRP-D and JRP-L [AJR89, BMSV+09, NS09]. $\mathbb{APX}$-hardness proofs, even for some restricted versions of JRP-D, were given by Nonner and Souza [NS09] and Bienkowski *et al.* [BBC+13a]. The first approximation algorithm, with ratio 2, was provided by Levi, Roundy and Shmoys [LRS06], and was subsequently improved by Levi et al. [LRSS08, LS06] to 1.8 (see also [LRS05]). For JRP-D, the ratio was reduced to 5/3 by Nonner and Souza [NS09] and then to ≈ 1.574 by Bienkowski *et al.* [BBC+13a]. All upper bounds are based on randomized rounding of the natural linear program for JRP. As shown in [BBC+13a], the integrality gap of this linear program is at least 1.245, even for JRP-D.

The online version of JRP was studied in the earlier discussed paper by Buchbinder *et al.* [BKL+08], who gave a 3-competitive algorithm, using a primal-dual scheme, and showed a lower bound of 2.64 on the competitive ratio, even for JRP-L. (See also Brito *et al.* [BKV12] for related work.)

1.2 Our Contributions. We provide several new approximation results for JRP. In the offline case, we give an algorithm with approximation ratio ≈ 1.791, breaking the barrier of 1.8 from [LRSS08, LS06]. The improvement is achieved by refining the analysis of the LP-rounding algorithm in [LRSS08, LS06] and combining it with a new algorithm that uses an approximation for JRP-D from [BBC+13a]. We also show that the integrality gap of the natural linear program for JRP-L is at least $12/11 \approx 1.09$. To our knowledge, this is the first explicit integrality gap construction for this most common version of JRP.

We also study online algorithms for JRP. We show that deterministic online algorithms, even for JRP-L, cannot be better than ≈ 2.754-competitive, improving the bound of 2.64 from [BKL+08]. For JRP-D, we prove that the optimal competitive ratio is 2.

2 Preliminaries

We now review our terminology and formalize the definition of JRP. Recall that $\mathcal{R}$ denotes the set of retailers. Each order can be specified by a triple $\pi = (\rho, a, h)$, where a is the time when π was issued, ρ is the retailer that issued π, and $h()$ is the waiting cost function of π, where $h(t) = \infty$ for $t < a$ and $h(t)$ is non-decreasing for $t \geq a$. Let Π be the set of all orders. In JRP-L we will assume that $h(t) = t - a$ for $t \geq a$, and in JRP-D we have $h(t) = 0$ for $a \leq t \leq d$ and $h(t) = \infty$ otherwise. Then d is called the *deadline* of order π. In JRP-D, we will in fact specify an order by a triple $\pi = (\rho, a, d)$.

A *shipment* is specified by a pair (S, t), where S is the set of retailers receiving the shipment and t is the time of the shipment. The cost of shipment (S, t) is $\mathsf{C} + \sum_{\rho \in S} \mathsf{c}_\rho$. A *schedule* is a set $\bar{S}$ of shipments. An order $\pi = (\rho, a, h)$ is said to be *pending* in $\bar{S}$ at time τ if $a \leq \tau$ and there is no shipment (S, t) in $\bar{S}$ with $\rho \in S$ and $a \leq t < \tau$. If $\pi = (\rho, a, h)$ is pending at time t and (S, t) is a shipment in $\bar{S}$ such that $\rho \in S$, then we say that (S, t) *satisfies* π. In such case, the waiting cost of π in $\bar{S}$ is $h(t)$. The cost of $\bar{S}$ is the sum of its shipment and waiting costs, that is

$$\textsc{Cost}(\bar{S}) = \textsc{Cost}_{\text{ship}}(\bar{S}) + \textsc{Cost}_{\text{wait}}(\bar{S}), \quad \text{where}$$
$$\textsc{Cost}_{\text{ship}}(\bar{S}) = \sum_{(S,t) \in \bar{S}} (\mathsf{C} + \sum_{\rho \in S} \mathsf{c}_\rho) \quad \text{and}$$
$$\textsc{Cost}_{\text{wait}}(\bar{S}) = \sum_{\pi = (\rho,a,h) \in \Pi} \min_{\substack{(S,t) \in \bar{S} \\ \rho \in S,\, t \geq a}} h(t) \ .$$

(In the last formula we assume that $\min \emptyset \equiv +\infty$.) The objective of JRP is to compute a schedule $\bar{S}$ with minimum $\textsc{Cost}(\bar{S})$.

We use the standard definition of approximation algorithms. We will say that a polynomial-time algorithm $\mathcal{A}$ is an *R-approximation algorithm* for JRP if for any instance it computes a schedule of shipments whose cost is at most R times the optimal cost for this instance.

In the online scenario, orders arrive over time, and at each time t an online algorithm must decide whether to ship at time t and, if so, to which retailers, based only on the existing orders. For online algorithms we use the term "*R-competitive*" as a synonym of "*R*-approximation".

For convenience, for online algorithms, we use a model where time is continuous, while some of previous works on this topic used the discrete-time model. Algorithms for the continuous model can be easily translated into the discrete model, preserving the same performance guarantee. In our lower bound proofs all waiting-cost functions are left-continuous, and for such functions lower bound arguments for competitive ratios carry over to the discrete case as well. This relationship will be formally spelled out in the full version of the paper (see a similar argument in [BKL+08]).

In the literature, some authors distinguish between absolute approximation ratios (as defined above) and asymptotic ratios, where an algorithm is allowed to pay some additional constant overhead cost, independent of the instance. While our upper bounds apply to the absolute ratio, our lower bound proofs can be extended to the asymptotic ratios by repeating the lower bound strategies a sufficient number of times.

3 An Upper Bound of 1.791 for Offline JRP

We now present our 1.791-approximation algorithm. The algorithm first computes an optimal solution (x^*, y^*) of the linear program for JRP. Then, it chooses randomly one of three different LP-rounding methods, with probabilities and other parameters suitably optimized, to obtain a ratio improving the bound of 1.8 from [LRSS08, LS06].

3.1 Linear Program. Let $T = \{a : (\rho, a, h) \in \Pi\}$ be the times when orders are placed. We can assume that all shipments occur at times in T. We use the following indicator variables: x_a represents a supplier-to-warehouse shipment at time a, $x_{\rho,a}$ represents a warehouse-to-retailer ρ shipment at time a, and $y_{\pi,a}$ represents an order π being satisfied by a shipment at time a.

The following linear program is the fractional relaxation of the natural integer program for JRP. Its goal is to minimize

$$\sum_{a \in T} \mathsf{C} \cdot x_a + \sum_{a \in T} \sum_{\rho \in \mathcal{R}} \mathsf{c}_\rho \cdot x_{\rho,a} + \sum_{\pi=(\rho,a,h)\in\Pi} \sum_{t \in T: t \geq a} h(t) \cdot y_{\pi,t}$$

subject to the following constraints:

$$(3.1) \quad x_a \geq x_{\rho,a} \quad \text{for all } \rho \in \mathcal{R}, a \in T$$

$$(3.2) \quad x_{\rho,a'} \geq y_{\pi,a'} \quad \text{for all } \pi = (\rho, a, h) \in \Pi, a' \in T$$

$$(3.3) \quad \sum_{t \geq a} y_{\pi,t} \geq 1 \quad \text{for all } \pi = (\rho, a, h) \in \Pi$$

$$(3.4) \quad x_a, x_{\rho,a} \geq 0 \quad \text{for all } \rho \in \mathcal{R}, a \in T$$

$$(3.5) \quad y_{\pi,a} \geq 0 \quad \text{for all } \pi \in \Pi, a \in T$$

The cost of any solution (x, y) to the LP above can be naturally split into three summands: the supplier-to-warehouse shipping cost, $\textsc{Cost}_{\textsc{wship}}(x, y)$; the warehouse-to-retailers shipping cost, $\textsc{Cost}_{\textsc{rship}}(x, y)$; and the waiting cost, $\textsc{Cost}_{\textsc{wait}}(x, y)$. When the solution (x, y) is a random variable, these denote appropriate *expected* costs.

Throughout the rest of the paper, we will fix an optimal (fractional) solution to the LP above and denote it by (x^*, y^*). Note that once x^* variables are fixed, y^* can be chosen greedily to minimize waiting cost: for any $\pi \in \Pi$, saturate constraints (3.2) starting from earliest possible $y_{\pi,a}$ till corresponding constraint (3.3) becomes feasible. In effect, without loss of generality, we may assume that constraints (3.3) are satisfied with equality in (x^*, y^*).

3.2 Known Algorithms: 2SRP and 1SRP. We say that a solution (x, y) is an (r_1, r_2, r_3)*-approximation* of (x^*, y^*) if the following three conditions hold:

- $\textsc{Cost}_{\textsc{wship}}(x, y) \leq r_1 \cdot \textsc{Cost}_{\textsc{wship}}(x^*, y^*)$,
- $\textsc{Cost}_{\textsc{rship}}(x, y) \leq r_2 \cdot \textsc{Cost}_{\textsc{rship}}(x^*, y^*)$, and
- $\textsc{Cost}_{\textsc{wait}}(x, y) \leq r_3 \cdot \textsc{Cost}_{\textsc{wait}}(x^*, y^*)$.

In our solution, we build on two LP-based, polynomial-time algorithms of Levi *et al.* [LRSS08]. Both are based on random shifting. The first one (denoted 2SRP) is called *Two-Sided Retailer Push Algorithm*, and the other (denoted 1SRP) is called *One-Sided Retailer Push Algorithm.*

Lemma 3.1. ([LRSS08]) *Algorithm 2SRP computes an integral solution (x, y) that is a $(1, 2, 2)$-approximation of the optimal fractional solution (x^*, y^*).*

Lemma 3.2. ([LRSS08]) *Algorithm 1SRP, parameterized by $c \in (0, 1/2]$, computes an integral solution (x, y) that is a $(\frac{1}{c}, \frac{1}{1-c}, \frac{1}{1-c})$-approximation of the optimal fractional solution (x^*, y^*).*

The currently best known 1.8-approximation algorithm [LRSS08] is obtained by simply running 2SRP with probability $\frac{3}{5}$ and 1SRP with probability $\frac{2}{5}$, setting $c = \frac{1}{3}$ in the latter.

3.3 High-level Idea. We start by showing that the $\textsc{Cost}_{\text{wait}}$ estimate of Algorithm 1SRP in Lemma 3.2 is not tight. To analyze it more accurately, we define a *shipping pace* of an algorithm and show a connection between the shipping pace and the waiting cost. Roughly speaking, shipping pace defines how fast the algorithm satisfies orders in comparison to how fast they are satisfied in the optimal fractional solution. Using our analysis it is possible to show that, for $c = \frac{1}{3}$, 1SRP computes in fact a $(3, \frac{3}{2}, \frac{9}{8})$-approximation. This improvement (over the $(3, \frac{3}{2}, \frac{3}{2})$-approximation guaranteed by Lemma 3.2) alone does not reduce the overall approximation ratio of the 1SRP-and-2SRP combination, as it is still dominated by the retailer shipment cost ratio.

However, we will add a third ingredient to this combination: Algorithm LPS. This new algorithm uses scaling of the fractional solution to obtain a new fractional solution obeying certain deadlines and then applies the recent result on JRP-D, the *deadline-constrained* variant of JRP [BBC+13a], to round it to an integral solution. By carefully choosing the scaling factor, probabilities of choosing Algorithms 2SRP, 1SRP and LPS, and fine-tuning the choice of c in Algorithm 1SRP, we eventually reduce the approximation ratio for JRP to about 1.791.

3.4 Shipping Pace. In (x^*, y^*), the orders can be thought of as being satisfied gradually with time. For any $\alpha \in [0,1]$ and any order $\pi = (\rho, a, h)$, we define $\mathsf{ft}_{\text{LP}}(\pi, \alpha)$ to be the first time when the already satisfied fraction of π in (x^*, y^*) is at least α. Formally:

$$\mathsf{ft}_{\text{LP}}(\pi, \alpha) = \min\left\{ t \in T : t \geq a \text{ and } \textstyle\sum_{t' \leq t} y^*_{\pi,t'} \geq \alpha \right\} .$$

To measure the waiting cost of an algorithm, we estimate how fast it satisfies each particular order in comparison to how fast these orders are satisfied in the fractional solution (x^*, y^*). Specifically, for any (randomized) algorithm $\mathcal{A}$ and an order $\pi = (\rho, a, h)$, we define $\mathsf{shp}_{\mathcal{A}}(\pi)$ to be time of the shipment satisfying π. Clearly, $\mathsf{shp}_{\mathcal{A}}(\pi) \geq a$ with probability 1. Note that, in the integral solution (x, y) generated by $\mathcal{A}$, it holds that $y_{\pi,t} = 1$ if and only if $t = \mathsf{shp}_{\mathcal{A}}(\pi)$.

Definition 3.1. *Let $G : [0,1] \to \mathbb{R}_{\geq 0}$ be an integrable and bounded function, such that $\int_0^1 G(z)\,\mathrm{d}z = 1$. We say that a (randomized) algorithm $\mathcal{A}$ has a* shipping pace G *if for any order $\pi = (\rho, a) \in \Pi$ and any $\alpha \in [0,1]$, it holds that*

$$Pr\Big[\mathsf{shp}_{\mathcal{A}}(\pi) \leq \mathsf{ft}_{\text{LP}}(\pi, \alpha) \Big] \geq \int_0^\alpha G(z)\,\mathrm{d}z .$$

A shipping pace is not unique; it is simply *a lower bound* on the shipping probability. By taking $\alpha = 1$, we obtain the property that if an algorithm has any shipping pace then it satisfies π by the time $\mathsf{ft}_{\text{LP}}(\pi, 1)$ with probability 1.

Lemma 3.3. *Let $\mathcal{A}$ be a (randomized) algorithm with shipping pace G that produces a solution (x, y). Then $\textsc{Cost}_{\text{wait}}(x, y) \leq \gamma(G) \cdot \textsc{Cost}_{\text{wait}}(x^*, y^*)$, where*

$$\gamma(G) = \sup_{w \in (0,1]} \left\{ \frac{\int_{1-w}^1 G(z)\,\mathrm{d}z}{w} \right\} .$$

Proof. We show that the relation above holds for the waiting cost of any individual order $\pi = (\rho, a, h)$. For the sake of this proof, we number all the consecutive times from the set $\{t \in T : t \geq a\}$ as $t_0 = a, t_1, t_2, \ldots$.

We first fix any $k \geq 0$ and show that

$$\sum_{j \geq k} \mathbf{E}[y_{\pi,t_j}] \leq \gamma(G) \cdot \sum_{j \geq k} y^*_{\pi,t_j} . \tag{3.6}$$

The relation (3.6) holds trivially for $k = 0$ as in this case both $\sum_{j \geq k} \mathbf{E}[y_{\pi,t_j}]$ and $\sum_{j \geq k} y^*_{\pi,t_j}$ are equal to 1 and $\gamma(G) \geq 1$ for any shipping pace G.

For $k \geq 1$, we define $w = \sum_{j \geq k} y^*_{\pi,t_j}$. By the definition of ft_{LP}, it holds that $\mathsf{ft}_{\text{LP}}(\pi, 1 - w) \leq t_{k-1}$. (Actually, this relation holds with equality if $y^*_{\pi,t_{k-1}} > 0$.) Then,

$$\begin{aligned} \sum_{j \geq k} \mathbf{E}[y_{\pi,t_j}] &= \Pr[\mathsf{shp}_{\mathcal{A}}(\pi) \geq t_k] \\ &= 1 - \Pr[\mathsf{shp}_{\mathcal{A}}(\pi) \leq t_{k-1}] \\ &\leq 1 - \Pr[\mathsf{shp}_{\mathcal{A}}(\pi) \leq \mathsf{ft}_{\text{LP}}(\pi, 1 - w)] \\ &\leq 1 - \int_0^{1-w} G(z)\,\mathrm{d}z = \int_{1-w}^1 G(z)\,\mathrm{d}z , \end{aligned}$$

where the first equality holds because y_{π,t_j} are binary random variables and exactly one of them is non-zero, and the second inequality follows from the definition of the shipping pace. Thus, relation (3.6) holds also for $k \geq 1$.

The waiting cost associated with π is

$$\begin{aligned} \textsc{Cost}^{\pi}_{\text{wait}}(x, y) &= \sum_{i \geq 0} h(t_i) \cdot \mathbf{E}[y_{\pi,t_i}] \\ &= h(t_0) \cdot \sum_{j \geq 0} \mathbf{E}[y_{\pi,t_j}] \\ &\quad + \sum_{i \geq 0} \big(h(t_{i+1}) - h(t_i) \big) \cdot \sum_{j \geq i+1} \mathbf{E}[y_{\pi,t_j}]. \end{aligned}$$

The waiting cost of π in (x^*, y^*) can be expressed analogously, but without taking expected values. Thus, relation (3.6) immediately implies that $\textsc{Cost}^{\pi}_{\text{wait}}(x, y) \leq \gamma(G) \cdot \textsc{Cost}^{\pi}_{\text{wait}}(x^*, y^*)$. The lemma follows by summing the waiting costs for all orders $\pi \in \Pi$. ◻

3.5 Waiting Cost of 1SRP. We start with a brief description of Algorithm 1SRP (see Algorithm 2 in [LRSS08]). The algorithm is parametrized by $c \in (0, \frac{1}{2}]$. It first computes the optimal fractional solution (x^*, y^*) and then it schedules the shipments, in two phases. In the first phase, it schedules the supplier-to-warehouse shipments. Intuitively, one can visualize this schedule in terms of the "virtual warehouse time", equal to the accumulated fractional shipping value for the warehouse, $X_t = \sum_{t' \le t} x_{t'}$. The algorithm chooses uniformly a random $\psi \in [0, c]$ and schedules the shipments at virtual warehouse times $\psi, \psi + c, \psi + 2c, \ldots$, which then can be translated into real times. More formally, these shipments are scheduled at (real) times t for which there is i such that $X_{t-1} < \psi + ic \le X_t$. In the second phase, we define tentative shipments from the warehouse to each retailer ρ. This is done similarly, by choosing a random $\psi_\rho \in [0, 1 - c]$ and tentatively scheduling these shipments at retailer ρ's virtual times $\psi_\rho, \psi_\rho + 1 - c, \psi_\rho + 2(1 - c), \ldots$. For each tentative shipment of ρ, say at a (real) time t, the actual shipment to ρ will take place at the first time $t' \ge t$ for which there is a supplier-to-warehouse shipment.

Lemma 3.4. *Algorithm 1SRP, with parameter $c \in (0, \frac{1}{2}]$, has a shipping pace*

$$G_{1\mathrm{SRP}}(z) = \frac{1}{1-c} \cdot \begin{cases} z/c & \text{for } z \in [0, c), \\ 1 & \text{for } z \in [c, 1-c), \\ (1-z)/c & \text{for } z \in [1-c, 1]. \end{cases}$$

Proof. (Sketch.) Every order is analyzed as if it was waiting first for a shipment (in the computed integral solution) at its retailer and then at the warehouse, with the analysis carried out with respect to the retailer's virtual time (the amount by which the fractional solution satisfies the order). Then the waiting at the retailer has uniform distribution $\mathcal{U}[0, 1 - c]$ and the waiting at the warehouse is upper-bounded with a uniform distribution $\mathcal{U}[0, c]$. Hence, the distribution of the total waiting time is bounded by the convolution of these two uniform distributions, which results in the trapezoidal shape of the shipping pace $G_{1\mathrm{SRP}}(x)$, see Figure 1. □

Side note. We can use Lemma 3.4 along with Lemma 3.3 to improve the waiting cost ratio of Algorithm 1SRP. Namely, the supremum of $(\int_{1-w}^{1} G_{1\mathrm{SRP}}(z)\,\mathrm{d}z)/w$ is achieved for $w = 1 - c$ and is then equal to $(2 - 3c)/(2(1 - c)^2)$. Setting $c = \frac{1}{3}$, we obtain that Algorithm 1SRP returns a $(3, \frac{3}{2}, \frac{9}{8})$-approximation. As we noted earlier, this result alone cannot improve the combination of 2SRP and 1SRP, because in any combination of these algorithms the retailer-cost ratio dominates the waiting-cost ratio.

3.6 Algorithm LPS. As indicated by the comment in the last paragraph, to improve the overall approximation guarantee, we therefore need to improve the two first coefficients in the approximation ratio. To this end, we design a new algorithm that performs well in terms of warehouse and retailer shipping costs and has a (reasonably) bounded waiting cost ratio. This algorithm (see below) randomly scales up the optimum solution (x^*, y^*), then, on this basis, it creates an instance of JRP-D, the variant of JRP with deadlines, to which it applies an approximation algorithm from [BBC+13a].

Lemma 3.5. ([BBC+13a]) *Fix any (not necessarily optimal) fractional solution $(\hat{x}, \hat{y})$ of an instance of the JRP-D problem. It is possible to compute, in polynomial time, an integral solution (x, y) which is (λ, λ)-approximation of $(\hat{x}, \hat{y})$, where $\lambda \approx 1.574$, i.e.,*

- $\textsc{Cost}_{\text{WSHIP}}(x, y) \le \lambda \cdot \textsc{Cost}_{\text{WSHIP}}(\hat{x}, \hat{y})$ *and*
- $\textsc{Cost}_{\text{RSHIP}}(x, y) \le \lambda \cdot \textsc{Cost}_{\text{RSHIP}}(\hat{x}, \hat{y})$.

Algorithm LPS uses a probability distribution D of the scaling parameter that we will define later.

Algorithm LPS

1. Compute an optimal fractional solution (x^*, y^*).
2. Choose $\zeta \in (0, 1]$ from a distribution with the density function $D : (0, 1] \to \mathbb{R}_{\ge 0}$.
3. Create a new fractional solution $(\widehat{x}, \widehat{y})$ by setting $\widehat{x} = \min\{1, x^*/\zeta\}$, and (greedily) choosing $\widehat{y}$ to minimize the waiting cost, subject to fixed fractional shipments $\widehat{x}$.
4. Create an instance $\mathcal{L}$ of JRP-D, by inserting a deadline for each order π at the first time t for which $\sum_{t' \le t} \hat{y}_{\pi,t'} \ge 1$. (Thus, in $(\widehat{x}, \widehat{y})$ each order is served "just in time".)
5. Randomly round the fractional solution $(\hat{x}, \hat{y})$ for $\mathcal{L}$ using the (λ, λ)-approximation algorithm from [BBC+13a], where $\lambda \approx 1.574$.
6. Return the obtained solution (x, y).

Lemma 3.6. *Let $\xi = \int_0^1 \frac{1}{z} \cdot D(z)\,\mathrm{d}z$. Algorithm LPS produces a feasible integral solution (x, y) with*

$$\textsc{Cost}_{\text{WSHIP}}(x, y) \le \lambda \cdot \xi \cdot \textsc{Cost}_{\text{WSHIP}}(x^*, y^*) \quad \textit{and}$$
$$\textsc{Cost}_{\text{RSHIP}}(x, y) \le \lambda \cdot \xi \cdot \textsc{Cost}_{\text{RSHIP}}(x^*, y^*).$$

Proof. We analyze the output (x, y) of Algorithm LPS for a fixed $\zeta \in (0, 1]$. By Step 3, $\textsc{Cost}_{\text{WSHIP}}(\widehat{x}, \widehat{y}) \le \textsc{Cost}_{\text{WSHIP}}(x^*, y^*)/\zeta$ and by Step 5, $\textsc{Cost}_{\text{WSHIP}}(x, y) \le \lambda \cdot \textsc{Cost}_{\text{WSHIP}}(\widehat{x}, \widehat{y})$ (see also Lemma 3.5). Thus, $\textsc{Cost}_{\text{WSHIP}}(x, y) \le (\lambda/\zeta) \cdot \textsc{Cost}_{\text{WSHIP}}(x^*, y^*)$. By integrating this estimate above over the probability distribution of ζ, we immediately obtain the first property of the lemma. The proof for the second property is analogous. □

Lemma 3.7. *Algorithm LPS has a shipping pace $G_{\mathrm{LPS}} \equiv D$.*

Proof. Fix any order $\pi = (\rho, a, h) \in \Pi$. We first show that if Algorithm LPS randomly choses scaling factor ζ, then it holds that

$$\mathsf{shp}_{\mathrm{LPS}}(\pi) \le \mathsf{ft}_{\mathrm{LP}}(\pi, \zeta) \ . \tag{3.7}$$

To this end, define $T(\pi) = [a, \mathsf{ft}_{\mathrm{LP}}(\pi,\zeta)] \cap T$. Then, using the definition of $\widehat{x}$ from Step 3, relation (3.2) from the original LP, and the definition of $\mathsf{ft}_{\mathrm{LP}}(\pi,\zeta)$, we obtain

$$\begin{aligned}\textstyle\sum_{t\in T(\pi)} \widehat{x}_{\rho,t} &= \textstyle\sum_{t\in T(\pi)} \min\{1, x^*_{\rho,t}/\zeta\} \\ &\ge \min\left\{1, \textstyle\sum_{t\in T(\pi)} x^*_{\rho,t}/\zeta\right\} \\ &\ge \min\left\{1, \textstyle\sum_{t\in T(\pi)} y^*_{\pi,t}/\zeta\right\} \\ &\ge 1.\end{aligned}$$

As the variables $\widehat{y}$ are greedily chosen to minimize the waiting costs (cf. Step 3), $\sum_{t\le \mathsf{ft}_{\mathrm{LP}}(\pi,\zeta)} \widehat{y}_{\rho,t} \ge 1$, and therefore the deadline for order π in $\mathcal{L}$ is set no later than at time $\mathsf{ft}_{\mathrm{LP}}(\pi,\zeta)$. As the solution (x, y) is feasible for $\mathcal{L}$, it obeys its deadlines, shipping π to ρ before $\mathsf{ft}_{\mathrm{LP}}(\pi,\zeta)$, i.e., (3.7) holds.

Relation (3.7) means that for any fixed α, the random event $\zeta \le \alpha$ implies that $\mathsf{shp}_{\mathrm{LPS}}(\pi) \le \mathsf{ft}_{\mathrm{LP}}(\pi,\alpha)$, and thus

$$\int_0^\alpha D(z)\,\mathrm{d}z = \Pr[\zeta \le \alpha] \le \Pr\Big[\mathsf{shp}_{\mathrm{LPS}}(\pi) \le \mathsf{ft}_{\mathrm{LP}}(\pi,\alpha)\Big],$$

which is exactly the definition of shipping pace D. □

3.7 Combining Algorithms 1SRP and LPS.

Algorithm 1SRP+LPS simply runs Algorithm 1SRP with probability p and Algorithm LPS with probability $1-p$. (Recall that we still have to choose parameter c in Algorithm 1SRP and the probability density D in Algorithm LPS.) We observe that such an algorithm has pace $G_{\mathrm{1SRP+LPS}} \equiv p \cdot G_{\mathrm{1SRP}} + (1-p)\cdot G_{\mathrm{LPS}}$. The following result is an immediate consequence of (i) using Lemma 3.2 and Lemma 3.6 to estimate the total warehouse and retailer shipping costs, (ii) applying Lemma 3.3 and Lemma 3.7 to estimate the waiting costs, and (iii) using the inequality $\gamma(G) = \sup_{w\in(0,1]}(\int_{1-w}^1 G(z)\,\mathrm{d}z)/w \le \sup_{z\in[0,1]} G(z)$.

Lemma 3.8. *Let $\xi = \int_0^1 \frac{1}{z}\cdot D(z)\,\mathrm{d}z$. Algorithm 1SRP+LPS computes an integral solution (x, y) that is a (r_1, r_2, r_3)-approximation of the optimal fractional solution (x^*, y^*), where*

$$\begin{aligned} r_1 &= p/c + (1-p)\lambda\xi \ , \\ r_2 &= p/(1-c) + (1-p)\lambda\xi \ , \text{ and} \\ r_3 &= \sup_{z\in[0,1]} G_{\mathrm{1SRP+LPS}}(z) \ . \end{aligned}$$

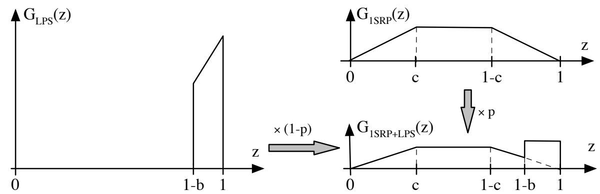

Figure 1: Shipping paces of Algorithms 1SRP, LPS and 1SRP+LPS.

Our next step is to choose a probability density function D in Algorithm LPS. This choice affects the approximation ratio in two ways. On the one hand, we want the value of ξ in Lemma 3.8 to be as small as possible. To this end, the probability mass should be accumulated close to the point 1. On the other hand, we need to take into account that Algorithm 1SRP+LPS approximates the waiting cost within the factor $\sup_{z\in[0,1]} G_{\mathrm{1SRP+LPS}}(z)$, where $G_{\mathrm{1SRP+LPS}} \equiv p\cdot G_{\mathrm{1SRP}} + (1-p)\cdot G_{\mathrm{LPS}} = p\cdot G_{\mathrm{1SRP}} + (1-p)\cdot D$.

Therefore, we choose D to be supported on the interval $[1-b, 1]$, where $b \le c$ is a parameter that we will fix later. Furthermore, we choose D to be such an increasing linear function that the resulting function $G_{\mathrm{1SRP+LPS}}$ is constant on $[1-b, 1]$, cf. Figure 1. To this end, we require that — within the interval $[1-b, 1]$ — the slope of the function $(1-p)\cdot G_{\mathrm{LPS}} \equiv (1-p)\cdot D$ matches the negated slope of the function $p\cdot G_{\mathrm{1SRP}}$. These considerations imply that once we fix parameters p, c and b, the probability density D should be

$$D(z) = \beta\cdot z + \frac{1}{b} + \frac{\beta\cdot b}{2} - \beta \ , \tag{3.8}$$

where

$$\beta = \frac{p}{(1-p)\cdot c\cdot(1-c)} \ ,$$

for $z \in [1-b, 1]$ and zero outside of this interval. Straightforward calculations verify that D is indeed a probability density, i.e., $\int_0^1 D(x)\,\mathrm{d}x = 1$. Furthermore, $G_{\mathrm{1SRP+LPS}} \equiv p\cdot G_{\mathrm{1SRP}} + (1-p)\cdot G_{\mathrm{LPS}}$ is constant on the interval $[1-b, 1]$ and its value there is equal to

$G_{1\mathrm{SRP+LPS}}(1) = (1-p)\cdot D(1) = (1-p)/b + pb/(2c(1-c))$. Thus

$$\begin{aligned} \xi &= \int_0^1 \frac{1}{z}\cdot D(z)\,\mathrm{d}z \\ &= \int_{1-b}^1 \beta\,\mathrm{d}z + \left(\frac{1}{b} + \frac{\beta\cdot b}{2} - \beta\right)\int_{1-b}^1 \frac{1}{z}\,\mathrm{d}z \\ &= \beta\cdot b - \left(\frac{1}{b} + \frac{\beta\cdot b}{2} - \beta\right)\cdot \ln(1-b)\ . \end{aligned}$$

We numerically optimize the parameters p, c and b. Specifically, we choose $p = 0.822599$, $c = 0.342538$ and $b = 0.136366$. For those values the maximum of function $G_{1\mathrm{SRP+LPS}}$ is achieved in the interval $[1-b, 1]$, and is at most 1.549968. Using the bounds of Lemma 3.8, we conclude that Algorithm 1SRP+LPS is an (R_1, R_2, R_2)-approximation of (x^*, y^*), where $R_1 \le 2.700277$ and $R_2 \le 1.549968$.

3.8 Combining All Algorithms. Finally, we combine algorithm 1SRP+LPS with algorithm 2SRP. The resulting algorithm 2SRP+1SRP+LPS uses 2SRP with probability $(R_1 - R_2)/(R_1 - R_2 + 1)$ and 1SRP+LPS with probability $1/(R_1 - R_2 + 1)$. Such algorithm is a (R, R, R)-approximation (of the fractional optimal solution), where

$$R = \frac{2\cdot R_1 - R_2}{R_1 - R_2 + 1} \le 1.790713\ .$$

We therefore obtained the following result.

Theorem 3.1. *There is a polynomial-time* 1.791*-approximation algorithm for JRP. The approximation ratio holds even against an optimal fractional solution of natural LP relaxation of the problem.*

4 Integrality Gap for JRP-L

In this section we show that the integrality gap for the natural linear program for JRP-L (namely the linear program given in Section 3.1, but with the linear cost function) is at least $12/11 \approx 1.09$. Our construction is inspired by some ideas behind the integrality gap proof for JRP-D in [BBC+13a].

We will use an instance with three retailers that we identify by numbers 0, 1 and 2: $\mathcal{R} = \{0, 1, 2\}$. The supplier-to-warehouse shipment cost is $\mathsf{C} = 3$ and all warehouse-to-retailer shipment costs are $\mathsf{c}_\rho = \mathsf{c} = 2$, for $\rho = 0, 1, 2$.

Orders will be issued only at integral times, between 0 and K, for some large integer K, that is $T = \{0, 1, ..., K\}$. We assume that K is a multiple of 3. For any retailer ρ, we will issue two orders at each time $3i+\rho$ and a single order at each time $3i+\rho+1$ (see Figure 2).

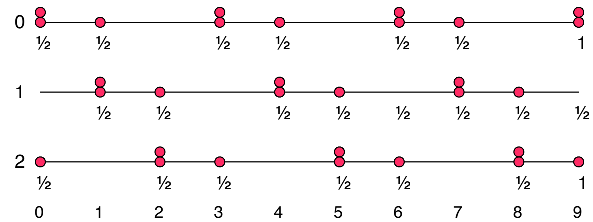

Figure 2: The instance of JRP-L used in the integrality gap construction, with $K = 9$. The horizontal lines correspond to the three retailers, numbered 0, 1 and 2; while the dots represent their orders. The figure also shows the fractional (half-integral) solution.

More precisely, to deal correctly with boundary cases, we define the order set Π as follows: for each time $t \in T$, Π contains the following three orders, all with waiting cost $h(\tau) = \tau - t$ for $\tau \ge t$:

$$(t \bmod 3, t, h),\ (t \bmod 3, t, h),\ \text{and}\ ((t+2) \bmod 3, t, h).$$

For each time t, we will refer to the three retailers as the *two-retailer*, *one-retailer*, and *zero-retailer*, according to the number of their orders issued at time t. Thus, the two-retailer is $t \bmod 3$, the one-retailer is $(t+2) \bmod 3$ and the zero-retailer is $(t+1) \bmod 3$.

4.1 Fractional Cost. We consider a fractional solution (x^*, y^*) in which each retailer has a fractional shipment of $\frac{1}{2}$ at each time $t < K$ when it issues an order (see Figure 2). Formally, for each $t = 0, 1, ..., K-1$ let

$$x^*_t = \tfrac{1}{2} \text{ and } x^*_{\rho,t} = \begin{cases} \frac{1}{2} & \text{if } \rho \in \{t \bmod 3, (t+2) \bmod 3\} \\ 0 & \text{if } \rho = (t+1) \bmod 3 \end{cases}$$

for each $\rho = 0, 1, 2$. For $t = K$, to guarantee feasibility, we let $x^*_K = x^*_{0,K} = x^*_{2,K} = 1$ and $x^*_{1,K} = \frac{1}{2}$. (Recall that K is a multiple of 3.) The values of y are assigned greedily, that is, if $\pi = (\rho, a, h)$ for $\rho = a \bmod 3$ then for $a \le K-1$ we let $y^*_{\pi,a} = y^*_{\pi,a+1} = \frac{1}{2}$, and for $a = K$ we let $y^*_{\pi,K} = 1$. Similarly, if $\pi = (\rho, a, h)$ for $\rho = (a+2) \bmod 3$ then we let $y^*_{\pi,a} = \frac{1}{2}$ for $a \le K-1$, $y^*_{\pi,a+2} = \frac{1}{2}$ for $a \le K-2$, $y^*_{\pi,K} = \frac{1}{2}$ for $a = K-1$ and $y^*_{\pi,K} = 1$ for $a = K$.

We now estimate the cost of this solution. Consider some time $t = 0, \ldots, K-1$. With this time t we associate the cost of the shipment at t and the waiting cost of all retailers in the interval $[t, t+1)$. The shipment at t costs $\frac{1}{2}(\mathsf{C} + 2\mathsf{c})$. In the interval $[t, t+1)$, the waiting cost of the two-retailer is $\frac{1}{2}\cdot 2 = 1$, and the waiting costs of the one-retailer and the zero-retailer are both $\frac{1}{2}$. So the total cost associated with time t is $\frac{1}{2}(\mathsf{C} + 2\mathsf{c}) + 1 + \frac{1}{2} + \frac{1}{2} = \frac{11}{2}$. We thus conclude that the cost of (x^*, y^*) per time unit is $\frac{11}{2}$, not counting the shipment cost at time K, so the overall cost of (x^*, y^*) is $\frac{11}{2}K + O(1)$. (Although it is not needed for our proof, one can show that (x^*, y^*) is at most a constant away from the actual optimum solution.)

4.2 Integral Cost. To estimate the integral cost, we will represent integral solutions by paths through a state diagram. In this diagram, the states will store the number of yet unsatisfied orders of each retailer at a given time. Moving from time t to $t+1$ will be represented by a transition between the corresponding states, labelled by the cost of this step. This cost will include the shipment cost at time t, if any, and the waiting cost incurred in the time slot $[t, t+1)$. This state diagram may have infinitely many states, but we can reduce it to only six states, by eliminating states and transitions that cannot occur in an optimal integral solution and by using symmetry.

The state at time t will be represented by a triple of integers (A, B, C), where A is the number of yet unsatisfied orders of the two-retailer, B is the number of yet unsatisfied orders of the one-retailer, and C is the number of yet unsatisfied orders of the zero-retailer, right after the orders were issued at time t but before the shipment is completed at time t, if any. For each state we will have $A \geq 2$ and $B \geq 1$, due to the orders placed at time t. For example, the initial state, at time 0, will be $(2, 1, 0)$.

Let (x, y) be some optimal integral solution for the above instance. We observe that, without loss of generality, we can assume that (x, y) satisfies the properties below:

(a) If in (x, y) the total number of yet unsatisfied orders at some time t is at least 9 then (x, y) has a shipment at time t. This holds because without a shipment at time t, (x, y) would pay cost 9 for waiting in the time interval $[t, t+1)$, which is at least as large as the cost of any shipment.

(b) If (x, y) has a shipment at time t then this shipment includes the two-retailer. This is true, since and thus the waiting cost for the α-retailer in the time interval $[t, t+1)$ would be at least $A \geq 2$, which is the same as the cost of joining the shipment at time t.

(c) If (x, y) has a shipment at time t then this shipment includes the one-retailer. To justify this, note that if (x, y) has a shipment involving this retailer at time $t+1$ then we could reduce the cost by removing this retailer from the shipment at time $t+1$ and have him join the shipment at time t. On the other hand, if there is no such shipment at time $t+1$, then this retailer would pay at least $2B \geq 2$ for waiting until the next shipment, so he can instead join the shipment at time t, without increasing the cost.

(d) If (x, y) has a shipment at time t and the $C \geq 2$ then this shipment includes the zero-retailer. This follows by the same argument as in part (b).

From these observations, we can assume that each state (A, B, C) in our state diagram has the following properties:

- (A, B, C) has a transition that corresponds to the shipment which involves all retailers with yet unsatisfied orders, going to state $(2, 1, 0)$. If $C = 0$ then the cost of this transition is 7, otherwise its cost is 9. If $A + B + C \geq 9$ and $C \geq 2$ then this is the only transition from this state.
- If $A + B + C \leq 8$ then (A, B, C) has a transition that does not involve a shipment, going to state $(C + 2, A + 1, B)$. The cost of this transition is $A + B + C$.
- If $C = 1$ then $(A, B, 1)$ has a transition corresponding to the shipment that involves the two- and one-retailer, going to state $(3, 1, 0)$. The cost of this transition is 8.

If we start from the initial state $(2, 1, 0)$ and expand the state diagram according to the rules above, we obtain the diagram shown in Figure 3. Each state is labeled by three integers (A, B, C), as explained above, and its potential value, to be discussed shortly. Transitions are represented by arrows with two labels. The first label indicates which retailers (two-, one- or zero-) are involved in a shipment, with "`s`" indicating that the corresponding retailer is in the shipment and "`-`" indicating that it's not. The second label is the cost of the transition.

To estimate the cost of the integral solutions, we use the potential function Φ on the states such that $\Phi(2, 1, 0) = 3$, $\Phi(3, 1, 0) = 2$ and $\Phi(s) = 0$ for all other states (see Figure 3). By routine verification, for any transition $s \to s'$ this potential satisfies

$$\ell(s, s') \geq \Phi(s') - \Phi(s) + 6,$$

where $\ell(s, s')$ is the cost associated with this transition. By adding up this inequality over all transitions corresponding to an optimal integral solution (x, y), we obtain that the cost of (x, y) is at least $6K - O(1)$. (This analysis is tight, since there are some cycles in the diagram where the per-transition cost is 6.)

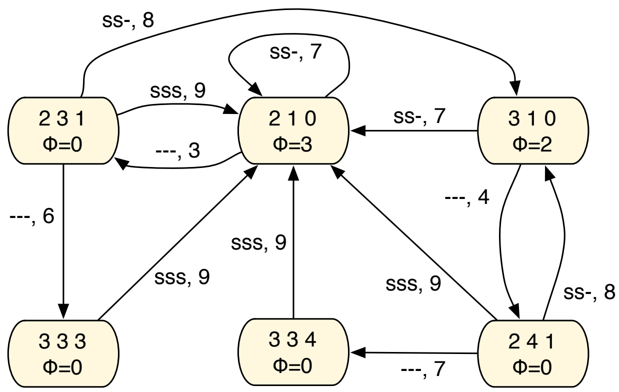

Figure 3: The state diagram showing possible behaviors of an optimal integral solution.

Finally, putting together our estimates of the fractional and integral solutions, we obtain that the ratio between the cost of (x, y) and the cost of (x^*, y^*) is

$$\frac{6K - O(1)}{\frac{11}{2}K + O(1)} \xrightarrow[K]{} \frac{12}{11}.$$

The argument is now complete, giving us the following theorem.

THEOREM 4.1. *The integrality gap of the linear program for JRP-L is at least* $\frac{12}{11} \approx 1.09$.

5 A Lower Bound of 2.754 for Online JRP-L

We now show our lower bound of 2.754 for the competitive ratios for JRP, which improves the previous lower bound of 2.64 by Buchbinder *et al.* [BKL+08]. Since we use only linear waiting-cost functions in our construction, as in [BKL+08], our result applies to JRP-L as well.

5.1 Single-Phase Game. In our lower-bound proof it will be convenient to consider a simple version of JRP-L that we refer to as the *Single-Phase JRP-L*. In the Single-Phase JRP-L all orders are issued at the beginning at time 0. Recall that the waiting cost is linear. In addition to the set of retailers and orders, the instance specifies also an *expiration time* θ. At time θ all orders expire: they need not be shipped anymore, but each incurs the waiting cost $h(\theta) = \theta$. Note that all information about the instance is known to the online algorithm, except for θ, which represents the adversary strategy. Thus, the Single-Phase JRP-L is in fact a generalization of the well studied rent-or-buy problem.

We claim that a lower bound of R for Single-Phase JRP-L implies a lower bound of R for JRP-L (and thus for JRP as well). Since a similar argument appeared before in [BKL+08, BBC+13b], we only briefly sketch the proof of this claim. Suppose that we have an adversary strategy that forces ratio R for Single-Phase JRP-L. We modify it into an adversary strategy that forces the same ratio for JRP-L. This strategy creates a large number of single-phase instances, concatenated together, with the i-th instance scaled by a factor of M^i, for some very large M, in the following sense: each order is replaced by M^i identical orders and the time is accelerated by a factor of M^i as well. By accelerating the time we mean that all time values used to make decisions in the strategy are multiplied by M^{-i}. These two changes together ensure that the waiting costs are not affected. The adversary applies the same strategy in each phase, forcing ratio R for each phase. Each phase may produce some number of non-satisfied orders, but these can be satisfied by one shipment for all retailers at the end of the game. This will add only a constant to the adversary shipment cost and, since the phase lengths are decreasing so fast, the increase of the adversary's waiting cost will be also negligible.

5.2 Single-Phase Construction. We will use an instance of Single-Phase JRP-L with $N+1$ retailers in $\mathcal{R}$, denoted $\rho_0, \rho_1, \ldots, \rho_N$. The costs of shipping from the warehouse to each of them are as follows: $\mathsf{c}_{\rho_0} = \mathsf{c}_0 = 0$ and $\mathsf{c}_{\rho_i} = \mathsf{c}_i = \mathsf{c}$ for all $i > 0$ and c that we fix later. These are normalized so that $\mathsf{C} = 1$, i.e., the cost of shipping from the supplier to the warehouse is 1.

Each retailer ρ_i places w_i identical orders $(\rho_i, 0, h_i)$ at time 0, where $h_i(t) = t$ for all i. Equivalently, we can view this as issuing a single order $\pi_i = (\rho_i, 0, h'_i)$ with weight w_i, that is with waiting cost function $h'_i(t) = w_i \cdot t$. We will adopt this latter terminology in this section. We choose the weights to be quickly decreasing, that is $w_i \gg w_{i+1}$ for all $i < N$, so that the slopes of the functions h'_i are decreasing rapidly with i. As a result, in the proof below, when the algorithm satisfies an order π_i, the waiting costs (of the algorithm and the adversary) of all orders $\pi_{i+1}, \pi_{i+2}, \ldots$ will be negligible. For clarity, in the calculations below we will assume these costs to be 0. (By adjusting the weights appropriately, we can make these costs at most an arbitrarily small ϵ, and then our lower bound will approach R.)

Let $\mathcal{A}$ be an online algorithm for Single-Phase JRP-L. To describe the adversary strategy, we first normalize the way $\mathcal{A}$ proceeds. Using a simple exchange argument, it is easy to show that, without loss of generality, $\mathcal{A}$ satisfies all the demands in increasing order of their indices, i.e., if $i < j$ then π_i is satisfied earlier than or together with π_j. Then, the adversary stops the game the moment that $\mathcal{A}$ satisfies more than one order with a single shipment. ("Stopping" means that the expiration time θ is set to the current time.) To complete the strategy's description we can thus focus on $\mathcal{A}$ satisfying orders $\pi_0, \pi_1, \ldots$ in this order, each with a dedicated shipment. If the waiting cost associated with π_i at the moment of its satisfaction is smaller than a certain threshold value σ_i, the game ends, otherwise it continues. This means that as long as the game did not end at of before $\mathcal{A}$'s shipment satisfying π_i, $\mathcal{A}$'s cost for these shipments is at least $\sum_{j=0}^{i}(1 + \mathsf{c}_j + \sigma_j)$. In particular, if the game does not end due to any aforementioned reason at or before the time that $\mathcal{A}$ satisfies π_N, then the game ends *naturally* with this shipment; otherwise we say that the game ends *prematurely*.

As was the case with c_i's, all the thresholds σ_i coincide and are denoted σ, with the exception of σ_0. We now give the values of all the parameters. We let c be the only real root of

$$\mathsf{c}^2(\mathsf{c}+1) = 1 \ , \tag{5.9}$$

and

$$\sigma_0 \equiv \frac{1}{\mathsf{c}+1} = \mathsf{c}^2 \ , \quad \sigma \equiv \sigma_0^2 = \mathsf{c}^4 = \mathsf{c}^2 + \mathsf{c} - 1 \ , \tag{5.10}$$

where the identities follow from (5.9). We have $\mathsf{c} \approx 0.7548$, $\sigma_0 \approx 0.5698$ and $\sigma \approx 0.3247$.

We claim that unless the game ends naturally, the competitive ratio of $\mathcal{A}$ is at least $R = 2 + \mathsf{c}$. To see this, let us consider all the ways in which the game can end prematurely.

Let ω be $\mathcal{A}$'s waiting cost of π_0 when it satisfies π_0. If $\omega < \sigma_0$ then $\mathcal{A}$'s cost is at least $1 + \omega$, whereas Opt can pay the waiting cost ω alone, resulting in ratio no smaller than

$$1 + \frac{1}{\sigma_0} = 2 + \mathsf{c} = R \ .$$

If $\mathcal{A}$ satisfies π_0 together with another order by a single shipment, then $\mathcal{A}$'s cost is at least $1 + \mathsf{c} + \omega$, whereas Opt will either pay the waiting cost ω for π_0 or 1 for satisfying π_0 at time 0. Thus, the competitive ratio is at least

$$\frac{1 + \mathsf{c} + \omega}{\min\{1, \omega\}} \geq \frac{2 + \mathsf{c}}{1} = 2 + \mathsf{c} = R \ .$$

Now we consider analogous two cases regarding the shipment for π_i, where $i \geq 1$, assuming that the game did not end before. This means that $\mathcal{A}$ already suffered a cost of at least $\sigma_0 + 1 + (i-1)(\sigma + 1 + \mathsf{c})$ associated with satisfying orders $\pi_0, ..., \pi_{i-1}$, plus some additional cost associated with satisfying π_i. Let now ω denote the waiting cost of π_i when $\mathcal{A}$ satisfies π_i.

If $\omega < \sigma$ then Opt satisfies the orders π_j for all $j < i$ with a single shipment at time 0, and pays the waiting cost ω for π_i. The competitive ratio is at least

$$\begin{aligned}
&\frac{\sigma_0 + 1 + (i-1)(\sigma + 1 + \mathsf{c}) + \omega + 1 + \mathsf{c}}{1 + (i-1)\mathsf{c} + \omega} \\
&= 1 + \frac{i + \mathsf{c} + \sigma_0 + (i-1)\sigma}{1 + (i-1)\mathsf{c} + \omega} \\
&\geq 1 + \frac{i + \mathsf{c} + \sigma_0 + (i-1)\sigma}{1 + (i-1)\mathsf{c} + \sigma} \ ,
\end{aligned}$$

which after substituting formulas (5.10) for σ_0 and σ, as well as using (5.9), becomes

$$\begin{aligned}
1 + \frac{1 + i\mathsf{c} + i\mathsf{c}^2}{i\mathsf{c} + \mathsf{c}^2} &= 2 + \frac{1 + i\mathsf{c}^2 - \mathsf{c}^2}{\mathsf{c}(i + \mathsf{c})} \\
&= 2 + \frac{i\mathsf{c}^2 + \mathsf{c}^3}{\mathsf{c}(i + \mathsf{c})} \\
&= 2 + \mathsf{c} = R \ .
\end{aligned}$$

Let us consider the remaining case in which $\mathcal{A}$ satisfies another order together with π_i. In this case Opt satisfies all the previous orders with a single shipment at time 0. As for π_i, Opt can satisfy it also with the shipment at time 0 or it can pay the waiting cost ω for π_i, whichever is cheaper. Thus the ratio is at least

$$\begin{aligned}
&\frac{\sigma_0 + 1 + (i-1)(\sigma + 1 + \mathsf{c}) + \omega + 1 + 2\mathsf{c}}{1 + (i-1)\mathsf{c} + \min\{\mathsf{c}, \omega\}} \\
&\geq 1 + \frac{i + 2\mathsf{c} + \sigma_0 + (i-1)\sigma}{1 + i\mathsf{c}} \ ,
\end{aligned}$$

which after substituting formulas (5.10) for σ_0 and σ, becomes

$$1 + \frac{1 + (i+1)\mathsf{c} + i\mathsf{c}^2}{1 + i\mathsf{c}} = 1 + \frac{(1+\mathsf{c})(1 + i\mathsf{c})}{1 + i\mathsf{c}} = 2 + \mathsf{c} = R \ .$$

Thus the ratio is at least R if the game ends prematurely. But if it does not, then $\mathcal{A}$'s cost for each shipment, except the one for π_0, is at least $1 + \mathsf{c} + \sigma = \mathsf{c}^2 + 2\mathsf{c}$, by (5.10). On the other hand, Opt satisfies all orders with a single shipment at time 0, which costs $1 + N\mathsf{c}$. With $N \to \infty$, Opt's cost of 1 for shipment from the supplier to the warehouse becomes negligible and Opt's cost per order tends to c. Therefore, the competitive ratio tends to $R = 2 + \mathsf{c}$. Summarizing the above argument, we obtain:

Theorem 5.1. *Each online deterministic algorithm for JRP-L has competitive ratio at least* 2.754.

6 An Upper Bound of 2 for Online JRP-D

We now present an online algorithm for JRP-D with (optimal) competitive ratio 2. A matching lower bound is given in Section 7. We will denote the shipments of the algorithm by (B_1, t_1), (B_2, t_2),..., where $t_1 \le t_2 \le ....$ The set B_j of retailers participating in the j-th shipment is called the jth *batch*. For convenience, we introduce a "dummy" 0'th shipment at time $t_0 = 0$, which we think of as if it shipped to all the retailers in the instance at no cost. (All that matters is that at time 0 no retailer has any pending orders.) For a retailer ρ and time t, we define the *deadline* of ρ to be the earliest deadline of an order in ρ pending at time t. If a retailer does not have any pending orders, its deadline is $+\infty$. Without loss of generality, we can assume that all shipments (of an online algorithm and the adversary) take place only at deadlines of some retailers. If t_j is the deadline of a retailer ρ then we say that ρ, or the order in ρ with deadline t_j, *triggers* shipment (B_j, t_j).

6.1 Algorithm $\mathcal{G}$. Suppose that we just completed shipment (B_{j-1}, t_{j-1}). We wait until we reach a deadline of a retailer, which will become the trigger retailer for the jth shipment. We denote this retailer by χ_j and its deadline by t_j. At time t_j our batch is $B_j = \{\chi_j\} \cup X_j$, where X_j contains the maximum number of retailers, in order of increasing deadlines, such that $\mathsf{c}(X_j) \le \mathsf{C}$.

If $j = 1$ then, according to our convention, $j-1 = 0$ refers to the dummy shipment at time $t_0 = 0$. Thus the first shipment will occur at the first deadline of the instance.

6.2 Analysis. We now analyze this algorithm. To simplify the analysis we will assume that all order arrival times and deadlines are different. The instance can be converted to have this property by an infinitesimal perturbation of arrival times and deadlines.

We divide the sequence of shipments into *phases*. A phase is a maximal interval $[g, h]$ of integers (indices of shipments), where $1 \le g \le h$, such that the adversary does not make any shipments in the time interval $(t_g, t_h]$. In other words: (i) there are no adversary shipments in $(t_g, t_h]$, (ii) the adversary shipped in $(t_{g-1}, t_g]$, and (iii) either t_h is the last deadline or the adversary shipped in $(t_h, t_{h+1}]$. Note that the first phase starts with the first shipment (that is $g = 1$). Indeed, the adversary must ship in the interval $(t_0, t_1]$, because t_1 is the first deadline. The lemmas below elucidate properties of phases that will be critical to our analysis.

LEMMA 6.1. *Let $[g, h]$ be a phase and $g < j \le h$. Let π be the order in χ_j that triggers shipment B_j. Then π was pending at time t_{j-1}, and among all orders pending at time t_{j-1} it was the earliest-deadline order not in B_{j-1}.*

Proof. Suppose that $\pi = (\chi_j, a, d)$, that is $d = t_j$. If we had $a > t_{j-1}$ then the adversary would have to make a shipment in the interval $[a, d] \subseteq (t_{j-1}, t_j]$, which would contradict the definition of a phase. So π was pending at time t_{j-1}. It must also be in fact the earliest-deadline order outside B_{j-1}, because B_j is the first shipment after B_{j-1}. □

LEMMA 6.2. *Let $[g, h]$ be a phase and $g \le j \le h$. Suppose that $\rho \in B_j$, where for $j = h$ we assume that $\rho = \chi_h$. Let $j' < j$ be maximum such that $\rho \in B_{j'}$ (if there is no shipment to ρ before t_j, let $j' = 0$). Then $j' < g$ and the adversary must ship to ρ in the interval $(t_{j'}, t_g]$.*

Proof. Right after the shipment $B_{j'}$, there were no orders in ρ, so all orders in ρ satisfied by B_j arrived after $t_{j'}$. Denote the order with the earliest deadline out of those by π and its deadline by d. Then $d \le t_h$: if $j = h$, this follows from our assumption that $\rho = \chi_h$, and otherwise from Lemma 6.1 by considering the order that triggers B_j.

Hence, the adversary has to satisfy π in $(t_{j'}, t_h]$, and as he makes no shipments in $(t_g, t_h]$, he must do so in $(t_{j'}, t_g]$; so in particular we get $j' < g$. □

THEOREM 6.1. *Algorithm $\mathcal{G}$ is 2-competitive for JRP-D.*

Proof. Consider a phase $[g, h]$. Using the above lemma, if $\rho \in B_j$, where either $g \le j < h$ or $j = h$ and $\rho = \chi_h$, then with the $\mathcal{G}$'s warehouse-to-ρ shipment at time t_j we can associate a unique warehouse-to-ρ shipment of the adversary that occurred not later than at time t_h.

With this in mind, we can now analyze the algorithm using a charging argument, as follows:

- We charge $\mathsf{c}(\chi_g)$, namely the cost of the warehouse-to-χ_g shipment at time t_g to the associated warehouse-to-χ_g shipment of the adversary (as described above). The charging ratio here is 1.
- We charge $\mathsf{C} + \mathsf{c}(X_h)$, representing the cost of the first supplier-to-warehouse shipment at time t_g and the cost of the warehouse-to-X_h shipment at time t_h, to the adversary cost of C of the supplier-to-warehouse shipment in $(t_{g-1}, t_g]$. Since $\mathsf{c}(X_h) \le \mathsf{C}$, the charging ratio is at most 2.
- For $j = g, ..., h-1$, we charge the cost $\mathsf{C} + \mathsf{c}(X_j) + \mathsf{c}(\chi_{j+1})$, that represents the supplier-to-warehouse shipment cost at time t_{j+1} and the cost of shipments warehouse-to-X_j and warehouse-to-χ_{j+1}, to

$\mathsf{c}(X_j)+\mathsf{c}(\chi_{j+1})$, namely the adversary's warehouse-to-retailer shipment cost associated with the retailers in $X_j \cup \{\chi_{j+1}\}$. By the choice of X_j and Lemma 6.1, we have $\mathsf{c}(X_j) + \mathsf{c}(\chi_{j+1}) > \mathsf{C}$, so the charging ratio is at most 2.

In all cases the charging ratio is at most 2, and different charges are assigned to different portions of the adversary cost, which implies the competitive ratio of Algorithm $\mathcal{G}$. □

7 A Lower Bound of 2 for Online JRP-D

In this section we show that no online algorithm for JRP-D can have competitive ratio smaller than 2, thus proving the optimality of the algorithm from the previous section.

THEOREM 7.1. *Every deterministic online algorithm for JRP-D has competitive ratio at least* 2.

Proof. Similarly to the proof in Section 5, we actually provide this lower bound for the restricted variant of JRP-D called *Single-Phase JRP-D*. In Single-Phase JRP-D, all orders arrive at time 0. The adversary can stop the game at any time θ (the expiration time), unknown to the online algorithm. All orders not satisfied by time θ incur no shipping cost. By an argument similar to the one given in Section 5, any lower bound for Single-Phase JRP-D implies the same lower bound for JRP-D. (For JRP-D the argument in Section 5 has to be slightly refined; the details will be given in the full version of this paper.)

In our instance, the supplier-to-warehouse shipping cost is $\mathsf{C} = 1$. We have $N+1$ retailers ρ_i, $i = 0, 1, ..., N$, for some sufficiently large N. Retailer ρ_0 has shipping cost $\mathsf{c}_{\rho_0} = 0$, and each retailer ρ_i, for $i > 0$, has shipping cost $\mathsf{c}_{\rho_i} = 1$. For each i, retailer ρ_i issues one order π_i at time 0 with deadline equal to i.

Let $\mathcal{A}$ be an online algorithm for JRP-D. $\mathcal{A}$ must ship to each ρ_i no later than at time i. Without loss of generality, $\mathcal{A}$ ships only at integer times, so as long as $\mathcal{A}$ ships to each retailer separately then each ρ_i will be shipped at time i. The adversary will stop the game as soon as $\mathcal{A}$ ships to more than one retailer. If this does not happen, the game stops after the shipments to all retailers, that is right after time N.

We argue now that this forces the competitive ratio of $\mathcal{A}$ to be arbitrarily close to 2. If $\mathcal{A}$ ships to each retailer separately, it pays 1 for retailer 0 and 2 for each retailer ρ_i, $i \geq 1$, for the total cost of $2N + 1$. The adversary can ship to all retailers at the beginning, paying $N + 1$. So the ratio approaches 2 with $N \to \infty$.

Suppose that at some time k, $\mathcal{A}$ ships to ρ_k and some other retailer, and let k be the first such k. Then $\mathcal{A}$'s cost is $1 + 2(k-1) + 3 = 2k + 2$. The adversary can ship to all retailers $\rho_0, ..., \rho_k$ at the beginning, paying $k + 1$. So the ratio is 2 also in this case. □

8 Final Comments

There are still significant gaps between the lower and upper bounds for the approximability of different variants of JRP. For JRP-D, we have $\mathbb{APX}$-hardness [NS09, BBC+13a] and an integrality gap of 1.245 [BBC+13a], while the best approximation ratio is 1.574 [BBC+13a]. Interestingly, JRP-L, which is the most common variant of JRP in the literature (in fact, JRP is frequently defined using linear waiting costs), is even less understood than JRP-D. The best upper bound is the same as for the general case, namely 1.791, shown in this paper, even though at this time not even an approximation scheme has been ruled out. Some progress on this problem was recently reported in [NS13]. The approximability of the online version of JRP-L also remains open.

It may be possible to use our approach to reduce the ratio for the general version of JRP to below 1.791 by modifying the 1.574-approximation algorithm for JRP-D in [BBC+13a], to obtain a parametrized bicriteria approximation that can be then optimized after combining it with Algorithms 2SRP and 1SRP. We leave this as a project for future work.

JRP can be naturally generalized to trees of arbitrary depth. This multi-level JRP problem with deadlines was studied by Bechetti *et al.* [BMSV+09], who provided a 2-approximation algorithm. (The objective function in [BMSV+09] is different than ours, but their proof works for our model.) Khanna *et al.* [KNR02] considered the case of linear waiting costs. Very recently, Lehilton Chaves (private communication) has shown that the general case can be reduced to the so-called multi-stage assembly problem, for which a 2-approximation algorithm was given by Levy *et al.* [LRS06].

References

[AJR89] Esther Arkin, Dev Joneja, and Robin Roundy. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8(2):61–66, 1989.

[BBC+13a] Marcin Bienkowski, Jaroslaw Byrka, Marek Chrobak, Neil Dobbs, Tomasz Nowicki, Maxim Sviridenko, Grzegorz Świrszcz, and Neal E. Young. Approximation algorithms for the joint replenishment problem with deadlines. In *Proc. of the 40th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 135–147, 2013.

[BBC+13b] Marcin Bienkowski, Jaroslaw Byrka, Marek Chrobak, Łukasz Jeż, Jiří Sgall, and Grzegorz Stachowiak. Online control message aggregation in chain networks. In *Proc. of the 13th Algorithms and Data Structures Symposium (WADS)*, pages 133–145, 2013.

[BKL+08] Niv Buchbinder, Tracy Kimbrel, Retsef Levi, Konstantin Makarychev, and Maxim Sviridenko. Online make-to-order joint replenishment model: primal dual competitive algorithms. In *Proc. of the 19th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 952–961, 2008.

[BKV12] Carlos Brito, Elias Koutsoupias, and Shailesh Vaya. Competitive analysis of organization networks or multicast acknowledgement: How much to wait? *Algorithmica*, 64(4):584–605, 2012.

[BMSV+09] Luca Becchetti, Alberto Marchetti-Spaccamela, Andrea Vitaletti, Peter Korteweg, Martin Skutella, and Leen Stougie. Latency-constrained aggregation in sensor networks. *ACM Transactions on Algorithms*, 6(1):13:1–13:20, 2009.

[KNR02] Sanjeev Khanna, Joseph Naor, and Danny Raz. Control message aggregation in group communication protocols. In *Proc. of the 29th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 135–146, 2002.

[LRS05] Retsef Levi, Robin Roundy, and David B. Shmoys. A constant approximation algorithm for the one-warehouse multi-retailer problem. In *Proc. of the 16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 365–374, 2005.

[LRS06] Retsef Levi, Robin Roundy, and David B. Shmoys. Primal-dual algorithms for deterministic inventory problems. *Mathematics of Operations Research*, 31(2):267–284, 2006.

[LRSS08] Retsef Levi, Robin Roundy, David B. Shmoys, and Maxim Sviridenko. A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Science*, 54(4):763–776, 2008.

[LS06] Retsef Levi and Maxim Sviridenko. Improved approximation algorithm for the one-warehouse multi-retailer problem. In *Proc. of the 9th Int. Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 188–199, 2006.

[NS09] Tim Nonner and Alexander Souza. Approximating the joint replenishment problem with deadlines. *Discrete Mathematics, Algorithms and Applications*, 1(2):153–174, 2009.

[NS13] Tim Nonner and Maxim Sviridenko. An efficient polynomial-time approximation scheme for the joint replenishment problem. In *Proc. of the 16th Integer Programming and Combinatorial Optimization (IPCO)*, pages 314–323, 2013.