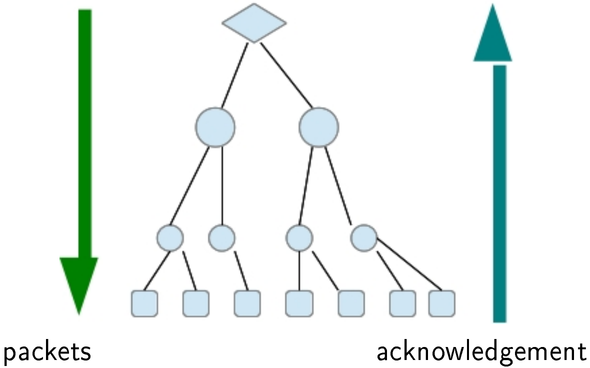
Approximation Algorithms for the Joint Replenishment Problem with Deadlines

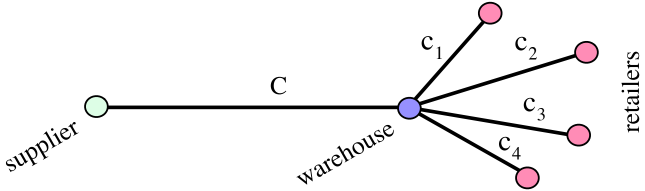
Marcin Bienkowski, Jaroslaw Byrka, Marek Chrobak,
Neil Dobbs, Tomasz Nowicki, Maxim Sviridenko,
Grzegorz Świrszcz, Neal E. Young

The problem, v1, acknowledgement aggregation



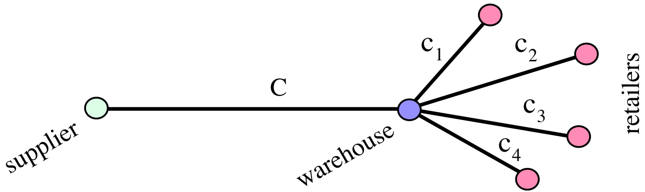
We will mostly consider hight 2 trees

The problem, v2, joint replenishment



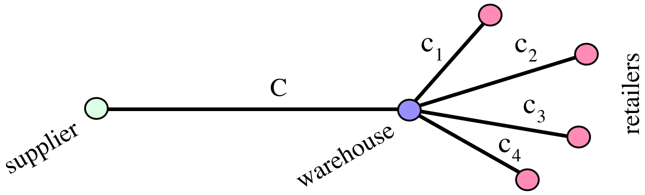
- given set of demands (retailer, time interval)
- compute a valid delivery schedule to quickly replenish used stock at retailers
- we may assume no storage at the warehouse
- minimize shipment costs (retailer orders + warehouse orders)
- pay per order, independent of the amount of items shipped

The problem, v3, make to order production planning



- just like before, but time is reversed
- given set of demands (retailer, time interval)
- compute a valid delivery schedule to provide items in time
- we may assume no storage at the warehouse
- minimize shipment costs (retailer orders + warehouse orders)
- pay per order, independent of the amount of items shipped

The problem, we decide for notation v2:JRPD



- cost model: linear waiting cost vs. deadlines
- some algorithms work in both models
- today we concentrate on deadlines
- also consider the uniform deadline case

LP-relaxation

$$
\begin{array}{llll}
\text{minimize} & cost(\boldsymbol{x}) & = & \sum_{t=1}^{U} \left(C\, x_t + \sum_{\rho=1}^{m} c_\rho\, x_t^\rho \right) \\
\text{subject to} & x_t & \geq x_t^\rho & \text{for all } t \in \mathcal{U}, \rho \in \{1, \ldots, m\} \quad (1) \\
& \sum_{t=r}^{d} x_t^\rho & \geq 1 & \text{for all } (\rho, r, d) \in \mathcal{D} \quad (2) \\
& x_t, x_t^\rho & \geq 0 & \text{for all } t \in \mathcal{U}, \rho \in \{1, \ldots, m\}.
\end{array}
$$

Previous results

- NP-complete : Becchetti et al. '09
- APX-hard (even for 3 demands per retailer) : Nonner and Souza '09
- 2-apx. primal-dual algorithm: Levi, Roundy and Shmoys '06
- 1.8-apx. : Levi et. al '08
- $5/3 \approx 1.67$ -apx. : Nonner and Souza '09

Our contribution

For general demands:

- e/(e-1) $\approx$ 1.58 apx. (easier version of the analysis)
- 1.574-apx. (more refined analysis)
- 1.245-lower bound on the inegrality gap

For equal length intervals

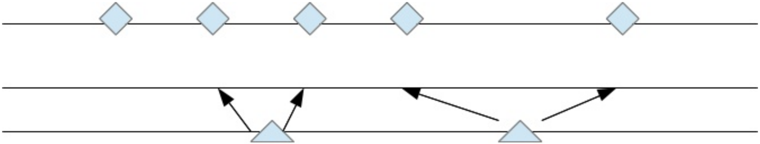
- 1.5-apx.
- APX-hardness (even with up to 4 demands per retailer)
- 1.2-lower bound on integrality gap

Easy algorithms: (bifactor approximation)

- Cost naturally splits into warehouse and retailer orders
- We consider LP-rounding alg. which directly relate the cost of the algorith to the corresponding part of LP-cost
- We bound $ALG = ALG_w + ALG_r \leq \lambda_w LP_w + \lambda_r LP_r$
- We say ALG is a (λ_w, λ_r)-apx algorithm.
- We will next show a (1,2) and a (3, 1.5)-apx algorithm

Easy algorithms: (1,2)-apx

- One level problem is easy
- Ignore retailer level cost to compute warehouse orders
- Compute optimal retailer orders given the fixed warehouse orders
- Show that retailer orders are now only twice more expensive than in OPT (or in LP)



Single retailer orders in OPT

Easy algorithms (3,1.5)-apx

- LP encodes density of shipments over time
- define "LP-time" between two events and the total LP-shipment between these events
- observe that "LP-time geos faster" on warehouse edge than on any retailers edge

Consider the following algorithm:

- plan warehouse shipment every 1/3 of "warehouse LP-time"
- plan retailer ρ shipment every 2/3 of "retailer ρ LP-time"

Note that we may combine (1,2) and (3, 1.5) to get (1.8, 1.8), which is essentially the work of Levi et al.

Our algorithm: Distribution

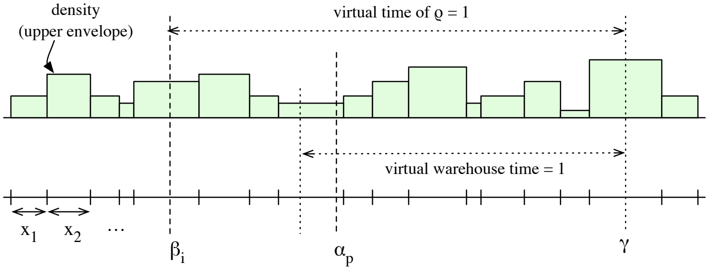
Instead of scheduling warehouse orders every 1/3 of LP-time, we do it iteratively and the next order is selected to be a certain random distance from the previous one.

Fix $\theta = 0.36455$ (slightly less than $1/e$). Over the half-open interval $[0, 1)$, the probability density function p is

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

The probability of choosing 1 is $1 - \int_0^1 p(y)\, dy \approx 0.0821824$.

LP virtual time



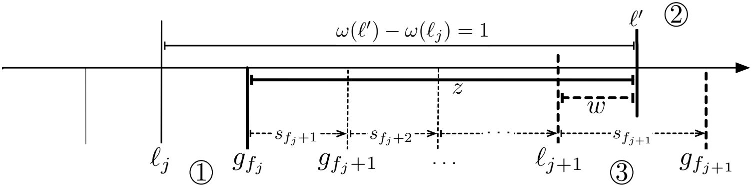
Algorithm $\mathrm{Round}_p(C, c_\rho, \mathcal{D}, \boldsymbol{x})$

1: Draw independent random samples $s_1, s_2, \ldots$ from p. Let $g_i = \sum_{h \le i} s_h$.
Set *global cutoff sequence* $\boldsymbol{g} = (g_1, g_2, \ldots, g_I)$, where $I = \min\{i \mid g_i \ge \hat{U} - 1\}$.

2: For each retailer ρ independently, choose ρ's *local cutoff sequence* $\boldsymbol{\ell}^\rho \subseteq \boldsymbol{g}$ greedily to touch all intervals $[a, b]$ with $\omega_\rho(b) - \omega_\rho(a) \ge 1$.
That is, $\boldsymbol{\ell}^\rho = (\ell_1^\rho, \ell_2^\rho, \ldots, \ell_{J^\rho}^\rho)$ where ℓ_j^ρ is $\max\{g \in \boldsymbol{g} \mid \omega_\rho(g) - \omega_\rho(\ell_{j-1}^\rho) \le 1\}$ (interpret ℓ_0^ρ as 0), and J^ρ is $\min\{j \mid \omega_\rho(\hat{U}) - \omega_\rho(\ell_j^\rho) \le 1\}$.

3: For each $g_i \in \boldsymbol{g}$, define time $t_i \in [U]$ to be minimum such that $\sum_{t=1}^{t_i} x_t \ge g_i$. Return the schedule $\big\{\big(t_i, \{\rho \mid g_i \in \boldsymbol{\ell}^\rho\}\big) \mid g_i \in \boldsymbol{g}\big\}$.

Algorithm: intuition



1. go forward one unit of retailer time (deadline)
2. go backward one unit of warehouse time
3. go forward to the next warehouse order (there must be at least one)
4. see how much time left before deadline (z on the picture)

Algorithm: fragment of easier variant of analysis

- take probability density function $p(y) = 1/y$ for $y \in [1/e, 1]$ and $p(y) = 0$ elsewhere.
- we bound the move on the warehouse time:
  $\mathbf{E}[p] = \int_{1/e}^{1} y\, p(y)\, dy \;=\; \int_{1/e}^{1} 1\, dy = 1 - 1/e.$
- we bound the waist on the retailer time:
  $\Pr[s_1 > z]z \;+\; \Pr[s_1 \leq z]\, \mathbf{E}[z - s_1 \mid s_1 \leq z].$
  This simplifies to $z \;-\; \Pr[s_1 \leq z]\, \mathbf{E}[s_1 \mid s_1 \leq z]$, which by calculation is

$$z - \int_{1/e}^{z} y\, p(y)\, dy \;=\; z - \int_{1/e}^{z} dy \;=\; z - (z - 1/e) \;=\; 1/e.$$

Walds theorem

Let random index $T \in \{0, 1, 2, \ldots\}$ be a stopping time for the sequence, that is, for any positive integer t, the event "$T < t$" is determined by state S_t.

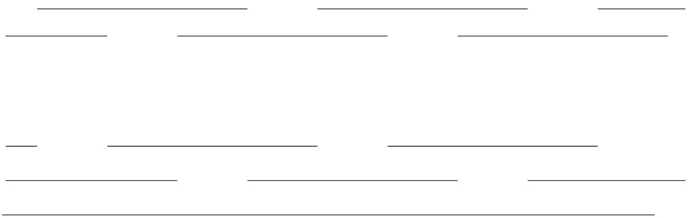
Lemma (Wald's equation)

Suppose that (i) $(\forall t < T)\ \mathbf{E}[\phi(S_{t+1}) \mid S_t] \geq \phi(S_t) + \xi$ *for fixed* ξ, *and (ii) either* $(\forall t < T)\ \phi(S_{t+1}) - \phi(S_t) \geq F$ *or* $(\forall t < T)\ \phi(S_{t+1}) - \phi(S_t) \leq F$, *for some fixed finite* F, *and* T *has finite expectation.*
Then $\xi\,\mathbf{E}[T] \leq \mathbf{E}[\phi(S_T) - \phi(S_0)]$.

In the applications here, we always have $\xi = \mathcal{Z}(p) > 0$ and $\phi(S_T) - \phi(S_0) \leq U$ for some fixed U. In this case Wald's equation implies $\mathbf{E}[T] \leq U/\mathcal{Z}(p)$.

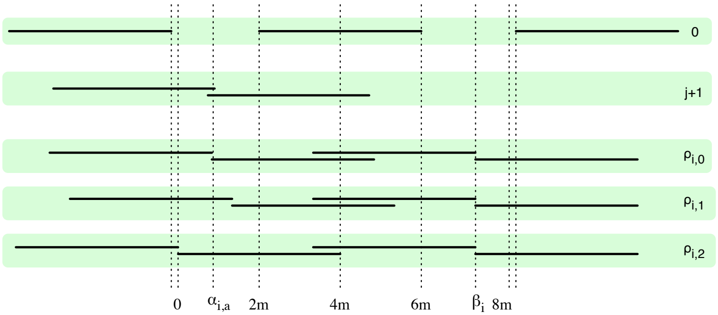
1.5-apx for uniform length demands

- instances of length 3 are plynomial time solvable
- create a set of small instances that cover all requests
- show that there exists solution to the set of small instances with cost 1.5 OPT

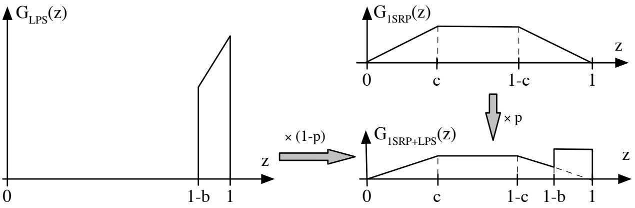


APX-hardness for uniform length demands, sketch

- reduce from degree 3 vertex cover
- synchronize 3 instances of a vertex
- for each edge: put two fresch copies of its endpoints nearby
- show that VC using K vertices corresponds to a solution of cost $10.5n + K + 6$



- best known apx. so far: 1.8
- we can now improve it to 1.791 by a combination of 3 algorithms [*]



[*] Joint work with Bienkowski, B., Chrobak, Jez, and Sgall

More recent results: online 2-level trees

- 3-competitive alg. for JRP [Buchbinder et al '08]
- 2.753-lower bound on comp. ratio for linear waiting costs [*]
- 2-competitive algorithm for online JRPD [*]
- matching lower bound of 2 on competitiveness fro JRPD [*]

[*] Joint work with Bienkowski, B., Chrobak, Jez, and Sgall

Even more related work: linear waiting penalty

Offline:

- polynomial time solvable on line networks [*]
- constant factor apx. for general trees

Online:

- already on a single edge it encodes a rent-or-buy problem
- 5-competitive alg. for a line [*]
- 2+ϕ lower bound on a line [*]
- open for general trees

[*] Joint work with Bienkowski, Chrobak, Jez, Sgall, and Stachowiak

Than you for your attention!