Order Acceptance and Sequencing Policies in a Make-to-Order Environments with Family-Dependent Due-Dates

N.D. van Foreest
Faculty of Economics and Business
University of Groningen

Joint work with Jacob Wijngaard and Remco Germs

June 25, 2010

Overview

- Illustrative example
- Related work and other examples
- Evaluation of some intuitive policies with simulation
- Policy optimization with Markov decision theory
- Comparing optimal policy and threshold policy
- Conclusions and furher work

Illustrative problem

- Consider a card controlled shop floor.
- The bottleneck resource (the packaging machine) is subject to significant family dependent setups.
- Meeting due-dates, and reducing leadtime variance, is essential for the MTO stream, and important to reduce inventories in the supply chain.
- If (customer) order rejection (or rerouting/subcontracting) is allowed, what would be good
  - customer order acceptance policies
  - work order release policies (from the order book)?

More Formal Problem Description

- Production Situation
  - Orders arrive randomly at rate λ, and are to be produced by a single machine.
  - Order i has four attributes: family, due-date, size, reward. Due dates are firm.
  - The family determines the leadtime (due-date – arrival date), reward and size.
  - Change of family requires a setup-time (not necessarily setup-cost)
  - Orders can be rejected (subcontract/produce elsewhere or on other, less efficient machine) at arrival.

More Formal Problem Description (2)

- Reward function:
  - $N(T) =$ total arrived reward in $[0, T]$.
  - $R(T) =$ total accepted reward in $[0, T]$.
  - $J(T) = R(T)/N(T)$ is accepted reward ratio,
  - $\lambda J(T) =$ the utilization is job reward equals the job size.
- Objective: identify policy that maximizes $\lim_{T\to\infty} J(T)$

Other Applications

- Scheduling the packaging process in a paint factory.
- E. Schmidt et al., Using Cyclic Planning to Manage Capacity at Alcoa, Interfaces, 2001 (Producing base ball bats)
- J.W.M. Bertrand, J.C. Wortmann, J. Wijngaard, Production Control, 1992 (Planning of a tube mill)
- N. Vandaele, I.V. Nieuwenhuyse, S. Cupers, Optimal grouping for a Nuclear Magnetic Resonance Scanner, EJOR 2003.
- Packaging of Eggs (Student master thesis assignment)
- Vehicle routing with selective pickups and deliveries.

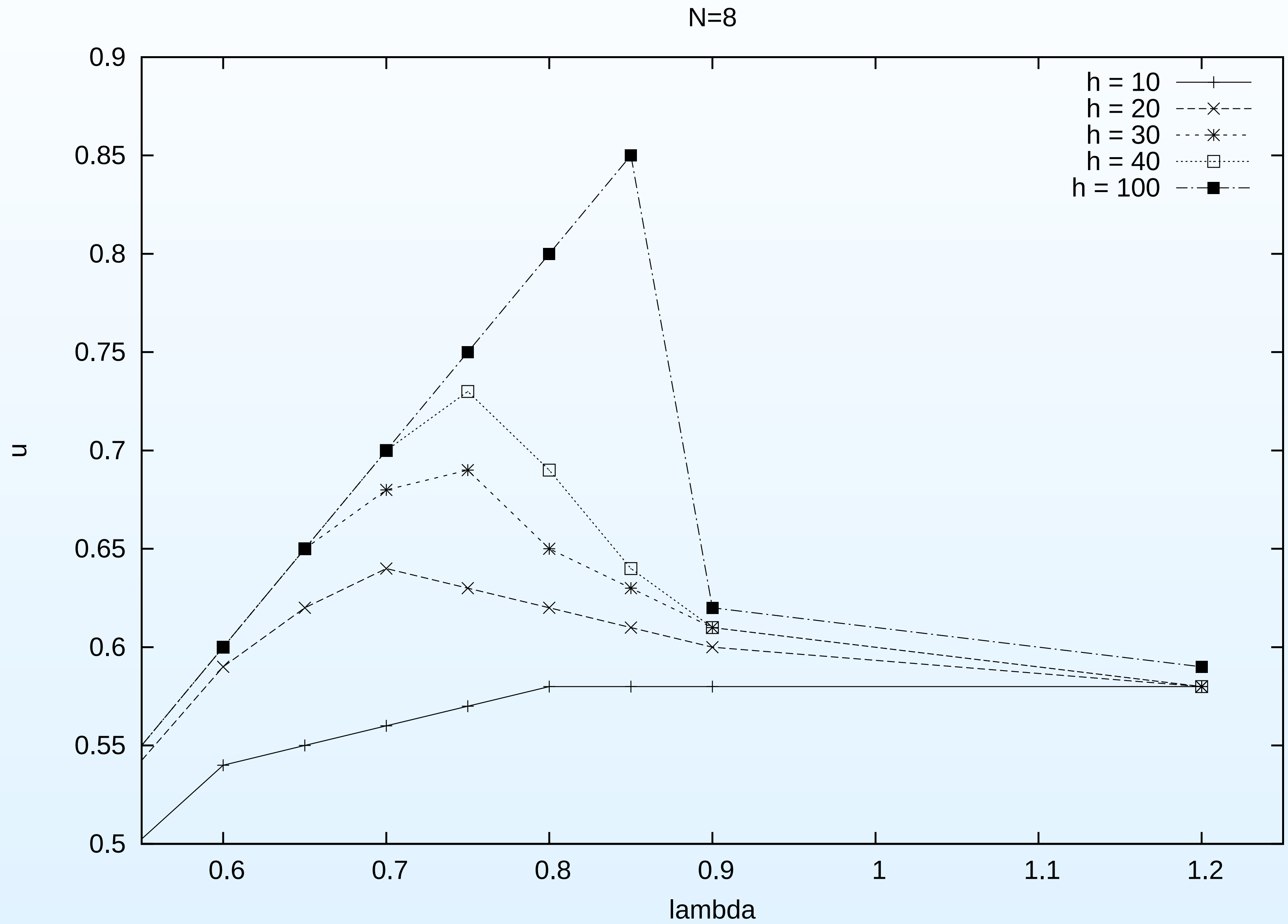
Theoretical Work

- $c\mu$ rule cannot be optimal, due to the firm due dates and the setups.
- H. Takagi, Analysis of Finite-Capacity Polling systems, AAP 1991
- E. Kim and M.P. Van Oyen, Finite-capacity Multi-class Production Scheduling with Setup Times, IIE Transactions. 1999
- F.A.W. Wester, J. Wijngaard, W.H.M. Zijm, Order acceptance strategies in a production-to-order environment with set-up times and duedates, IJPR 1992,
- H.A. ten Kate, Order acceptance and production control, PhD. Thesis (J. Wijngaard, W.H.M. Zijm), 1995
- W.H.M. Zijm, J.M.J. Schutten, S. L. van de Velde, Scheduling a single machine with release and due dates and family-dependent set-up times, Management Science 1996.
- E.M.M. Winands, I.J.B.F. Adan, G.J. van Houtum, Stochastic Economic Lot Scheduling Problem: A Survey, Beta Report 2005.
- N.D. van Foreest, J.T. van der Vaart, J. Wijngaard, Scheduling and Order Acceptance for the Customized Stochastic Lot Scheduling Problem, IJPR 2010

Simplest Characteristic Case

- Poisson (λ) arrival process
- N families; $P($ arrival is of family $i) = 1/N$; independent.
- all jobs and setups have the same unit size, unit reward per job
- all families have equal lead time horizon
- Non-preemptive service of jobs and setups.
- No machine breakdowns.
- Actions: Reject, Combine, and Spawn
- Initial problem: Find a stationary policy that performs well w.r.t. acceptance ratio, and is robust (relatively insensitive to load).
- Consider the greedy policy: accept a job whenever possible.

Greedy Policy (simulation results) $N = 8$



Why this Bad Behavior?

- Suppose last order in schedule is green, and the schedule is tight.
-

$$
\begin{aligned}
q &= P(\text{The next order belongs to another family}) \\
&= P(\text{no green arrival during}[0,1))P(\text{next order is not green}) \\
&= e^{-\lambda/N}\frac{N-1}{N}.
\end{aligned}
$$

- Expected runlength $R = 1/q \approx 8/7$ when $N = 8$ and $\lambda < 1.3$.
- Utilization u in this case:

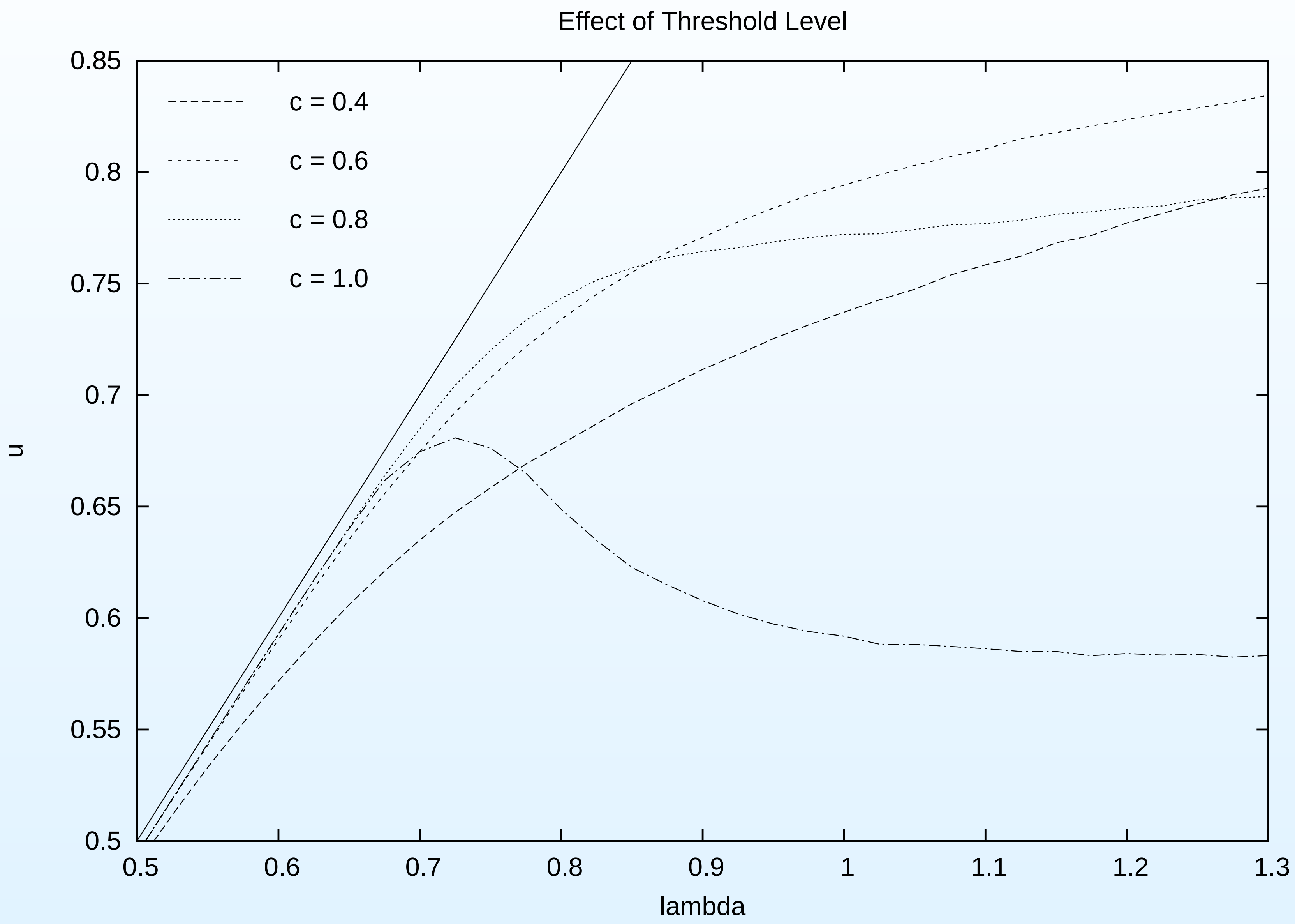
$u \approx \frac{R}{R+s} \approx \frac{8/7}{8/7+1} = \frac{8}{15} \approx 0.53$

- When the schedule becomes tight, it stays tight for long times, since the load suddenly changes to 2λ.

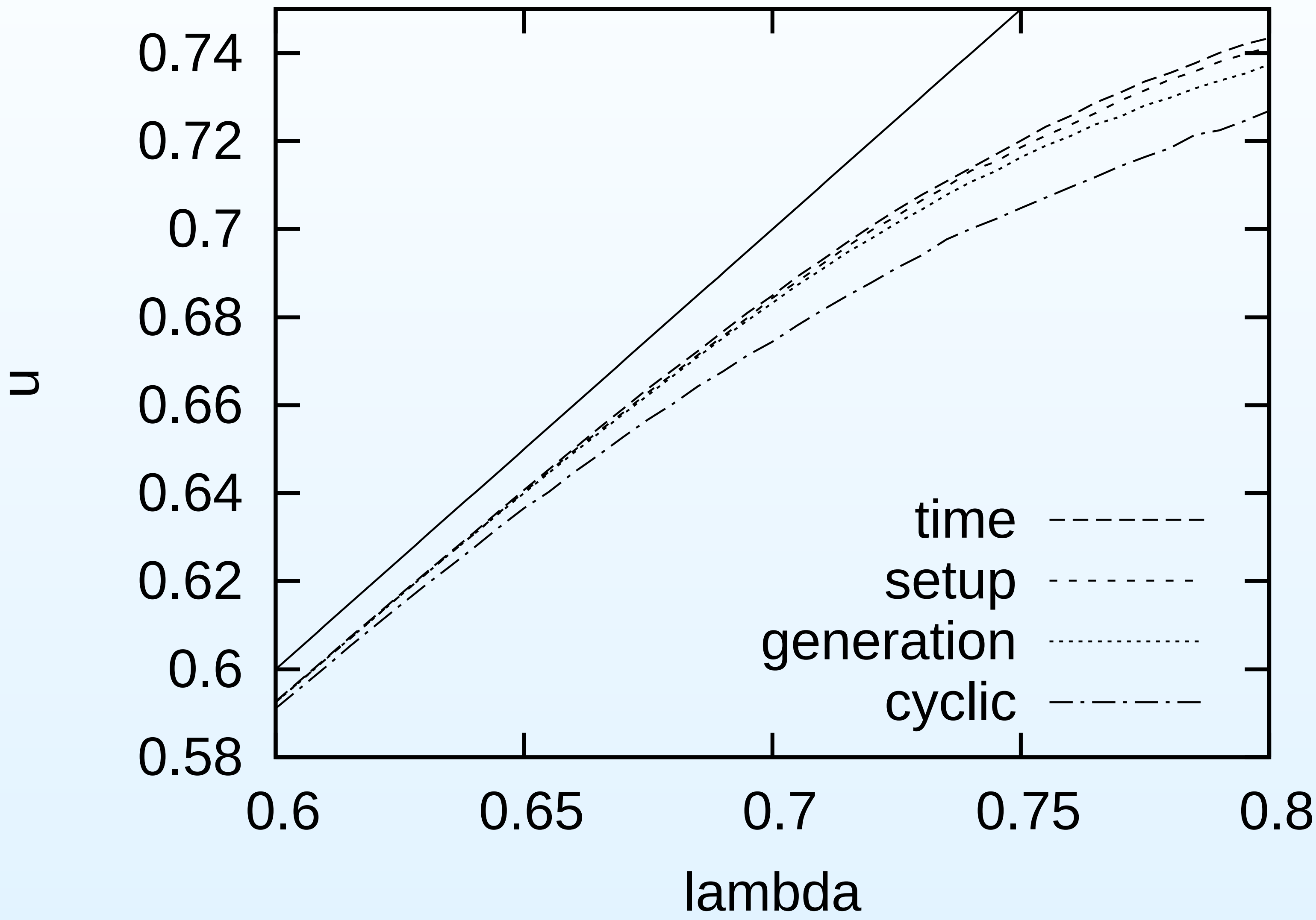
A Better Policy

- Problem: the greedy policy spawns new runs at the end of the schedule.
- Thus, new job is *tight* and prevents *combination potential*.
- Promising resolution: restrict the Spawn action.
- Threshold policy: Suppose an order of family f arrives, then
  - always accept the order if it can be combined with a run of its family;
  - if it is not possible to combine, then only accept when the slack $\geq ch$, where $c \in [0, 1]$, and h is the lead time.

Threshold Policies (simulation results)



Cyclic policy, Restrict # setups (simulation results)



Identifying the Optimal Policy

- Simulation results suggest that a threshold type policy is optimal.
- Is the optimal policy of threshold type?
- Proof by interchange arguments fail, due to the rejection of orders.
- Exact numerical analysis by Markov decision theory.

Building the MDP

Building the MDP is rather involved

1. Find a memory efficient state description
2. Identify smart state aggregation methods
3. Specify the transition matrix per action
4. Use the transition matrices to generate reachable space (enumerating the entire state space is impossible)
5. Use policy iteration to compute optimal policy

Visualizing Structure of Optimal Policy (2 families)

	Family 1		Family 2		
Slack	`Spawn`	`Reject`	`Spawn`	`Reject`	Total
8	0.00	0.00	2.35	0.00	2.35
7	0.00	0.00	2.82	0.00	2.82
6	3.49	0.00	5.26	0.00	8.75
5	4.19	0.00	3.00	0.00	7.19
4	6.43	0.00	2.28	0.00	8.71
3	8.49	0.00	2.75	0.00	11.25
2	0.00	12.41	4.15	0.00	16.65
1	0.00	14.65	0.00	6.51	21.16
0	0.00	15.75	0.00	5.47	21.22
Total	22.60	42.81	22.60	11.98	100.00

- $N = 2, h = (8, 10), \lambda = (2/5, 4/5)$.
- Visualization method is based on an approach of Haijema, Van der Wal and Van Dijk (C&OR, 2007).

Visualizing Structure of Optimal Policy (3 families)

	Family 1		Family 2		Family 3		
Slack	`Spawn`	`Reject`	`Spawn`	`Reject`	`Spawn`	`Reject`	Total
8	1.01	0.00	1.11	0.00	1.01	0.00	3.12
7	1.21	0.00	1.33	0.00	1.21	0.00	3.74
6	2.14	0.00	2.59	0.00	2.14	0.00	6.86
5	2.96	0.00	2.11	0.00	2.96	0.00	8.04
4	4.19	0.46	2.35	0.00	4.19	0.46	11.65
3	0.00	6.95	2.34	0.00	0.00	6.95	16.24
2	0.00	7.63	2.12	0.26	0.00	7.63	17.63
1	0.00	7.08	2.09	0.50	0.00	7.08	16.76
0	0.00	6.41	0.00	3.15	0.00	6.41	15.97
Total	11.51	28.52	16.03	3.91	11.51	28.52	100.00

- $N = 3, h = (10, 10, 10), \lambda = (2/5, 2/5, 2/5)$.
- Threshold policy is *not* optimal.

Visualizing Structure of Optimal Policy (3 families)

	Family 1		Family 2		Family 3		
Slack	Spawn	Reject	Spawn	Reject	Spawn	Reject	Total
8	1.01	0.00	1.11	0.00	1.01	0.00	3.12
7	1.21	0.00	1.33	0.00	1.21	0.00	3.74
6	2.14	0.00	2.59	0.00	2.14	0.00	6.86
5	2.96	0.00	2.11	0.00	2.96	0.00	8.04
4	4.19	0.46	2.35	0.00	4.19	0.46	11.65
3	0.00	6.95	2.34	0.00	0.00	6.95	16.24
2	0.00	7.63	2.12	0.26	0.00	7.63	17.63
1	0.00	7.08	2.09	0.50	0.00	7.08	16.76
0	0.00	6.41	0.00	3.15	0.00	6.41	15.97
Total	11.51	28.52	16.03	3.91	11.51	28.52	100.00

Analysis of the threshold policy

- We compared the performance of the threshold policy and the optimal policy for a wide range of parameter settings:
  - Number of families;
  - Arrival rate;
  - Processing and setup time;
  - Lead time horizon;
  - Reward per arriving order.
- Conclusion: the performance of the threshold policy is near optimal and robust to asymmetries in processing time, arrival rate, profit and lead time.
- Latest results: using family-dependent due-dates results in significantly higher profits than the standard practice of using uniform due-dates.

Conclusions

- Introducing slack into the schedule ensures combination potential. In other words, occasionally rejecting orders reduces loss probability.
- Cyclic policies are suboptimal
- Threshold type policy step is simple to understand and implement, robust, and performs excellently.

Possible Extensions

- Find simple models to estimate the utilization for each family.
- Find a good model to estimate the threshold level.
- What is the structure of the optimal policy?
- Would stochastic job sizes or setup times have strong effect on the structural results we obtained up to now?
- Include occasional tardiness?