This is an Article Template for My Documents with LaTeX

Author1[a,*] and Author2[b]

[a]School of Economics and Management, Southeast University, Nanjing 211189, China
[b]School of Management, University of Science and Technology of China, Hefei 230096, China

January 4, 2023

Abstract

LaTeX, which is pronounced «Lah-tech» or «Lay-tech» (to rhyme with «blech» or «Bertolt Brecht»), is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing. LaTeX is not a word processor! Instead, LaTeX encourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content.

Keywords: LaTeX, word processor, scientific document, high-quality typesetting.

Contents

1 Section 1

The COVID-19 pandemic is rampant around the world, putting heavy pressure on the medical supply chain (Inderfurth and Kleber, 2013). In particular, new dangerous variants (such as Delta and Omicron) of coronavirus spread faster in these two years.

Li and Shang (2021) first propose a remanufacturing pattern named hybrid combinatorial remanufacturing (HCR) for the PCB-based product that its modules are connected to the printed circuit board (PCB), and find it suitable for the product with high material costs and low operating costs.

2 Section 2

This is contents for Section 2. For example, Literature Review.

3 Section 3

Table 1. Table 1

In-transit products before deadline					Net demands					Inventory of modules				
$S_1 - s_1$	$\cdots$	$S_i - s_i$	$\cdots$	$S_I - s_I$	$N_1 - n_1$	$\cdots$	$N_i - n_i$	$\cdots$	$N_I - n_I$	k_1	$\cdots$	k_{m_j}	$\cdots$	k_{M_j}

A normal equation, see (1).

$$\min_{d_{s_i,a_t}} C = \sum_{i=1}^{I} \sum_{s_i=1}^{S_i} \sum_{a_t \in \Omega} d_{s_i,a_t} C_{s_i,a_t}, t \leftrightarrow s_i, 1 \leq t \leq T. \tag{1}$$

A matrix equation is shown in (2).

$$X_t(a_t) = \begin{bmatrix} e_{u,s_1} & \cdots & e_{u,s_i} & \cdots & e_{u,s_I} & e_{u,n_1} & \cdots & e_{u,n_i} & \cdots & e_{u,n_I} & e_{u,1} & \cdots & e_{u,m_j} & \cdots & e_{u,M_j} \\ e_{r,s_1} & \ddots & e_{r,s_i} & & e_{r,s_I} & e_{r,n_1} & \ddots & e_{r,n_i} & & e_{r,n_I} & e_{r,1} & \ddots & e_{r,m_j} & & e_{r,M_j} \\ e_{o,s_1} & & e_{o,s_i} & & e_{o,s_I} & 0 & & 0 & & 0 & e_{o,1} & & e_{o,m_j} & & e_{o,M_j} \\ e_{p,s_1} & & e_{p,s_i} & \ddots & e_{o,s_I} & 0 & & 0 & \ddots & 0 & e_{p,1} & & e_{p,m_j} & \ddots & e_{p,M_j} \\ e_{w,s_1} & \cdots & e_{w,s_i} & \cdots & e_{w,s_I} & 0 & \cdots & 0 & \cdots & 0 & e_{w,1} & \cdots & e_{w,m_j} & \cdots & e_{w,M_j} \\ 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & & \vdots & \vdots & \ddots & & & \vdots & \vdots & \ddots & & & \vdots \\ 0 & & 0 & & 0 & 0 & & 0 & & 0 & 0 & & 1 & & 0 \\ \vdots & & & \ddots & \vdots & \vdots & & & \ddots & \vdots & \vdots & & & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}. \tag{2}$$

3.1 Subsection 3.1

My dear and beautiful sister is shown in F

3.2 Subsection 3.2

We have the following four strategies for actions.

Strategy I.

Strategy II.

Strategy III.

Strategy IV.

Algorithm 1 Get_Current_Action(Z_t, Q_t, *epsilon*, strategy)

---

initialize a_t = null, u_planned = False, r_planned = False, routine = False
$\Omega_t \leftarrow$ Get_Action_Space(Z_t)
if *epsilon* $>$ eps **then**
    reweight the elements in Ω_t according to the risk preference
    randomly select an element from Ω_t as a_t
else
    repeat
        $a_t \leftarrow \arg\max_{a_t} \mathrm{Q}_t$ //find a_t with the highest Q value in Q_t
        if $a_t \in \Omega_t$ **then**
            break
        else
            exclude $\mathrm{Q}(Z_t, a_t)$ from Q_t
        end if
    until Q_t has no element left
end if
if $\mathrm{Q}_t = \emptyset$ **then**
    return a_t //return null as no action available
end if
if strategy is IV **then**
    if a_t makes $k_{m_j} < 0$ for $\forall m_j$ **then** //module inventory is not enough
        if $a_t = u$ **then**
            u_planned $\leftarrow$ True //action u is planned
        else
            r_planned $\leftarrow$ True //action r is planned
        end if
        routine $\leftarrow$ True //execute routine job to fill in the inventory gap
        $\Omega_t \leftarrow$ Get_Action_Space(Z_t) //re-obtain Ω_t with action u or r planned

```
        randomly select an element from Ω_t as a_t
        return a_t
    end if
    if a_t ∈ {u, r} then
        routine ← False                              //routine job has done
    end if
end if
return a_t
```

Theorem 1. *Following the DeSoRVA rule, $N_i - n_i > 0$.*

Proof. Assume that there is no action available for the first time, but the demand has not been met for product s_i, i.e., $N_i - n_i > 0$. □

From Theorem 1, we have the following 7 schemes.

- Scheme PS.
- Scheme RS.
- Scheme RQ.
- Scheme EQ.
- Scheme ED.
- Scheme OQ.
- Scheme OD.

4 Conclusion

Conclude this paper.

Appendix

References

Inderfurth, K., Kleber, R., 2013. An advanced heuristic for multiple-option spare parts procurement after end-of-production. Production and Operations Management 22, 54–70. doi:10.1111/j.1937-5956.2012.01358.x.

Li, S., Shang, Y., 2021. A quality status encoding scheme for pcb-based products in iot-enabled remanufacturing. Frontiers of Computer Science 15, 155615. doi:10.1007/s11704-020-9175-0.