1. Multi-period, Multi-product p-Batch Processing Time Maximization Problem

The MPpBPTM problem arises when a set of different products are processed simultaneously in a same production batch. In this problem, it is considered that the quantity produced of each product is directly proportional to the processing time, however, with a different constant of proportionality (production rate) for each product. In addition, there is a maximum quantity allowed for the production of batch products, defined both individually and for the set. The production quantity of each product is mainly defined according to the demand for the product. However, it is still possible to stock the products and / or send them to the outlets. In both cases, there is a stocking / shipping limit for each product and a stocking / shipping limit for the set of products in the batch. Also, there is a time limit available for processing the batch. About the production, the industry manager has as priority to meet demand and he prefer to send the remaining production to outlets rather then stock it in the factory.

The Multi-period, Multi-product p-Batch Processing Time Maximization (MPMPpBPTM) problem é uma versão mais complexa do MPpBPTM na qual considera-se um horizonte de planejamento contendo mais do que um único período (dias, semanas, etc). Neste caso as decisões devem ser tomadas para cada período.

O MPMPpBPTM é mais complexo do que o MPpBPTM porque, no caso do primeiro, a decisão sobre a produção de determinado produto em determinado período pode influir fortemente nos limites permitidos para produção do mesmo produto e de outros produtos em outros períodos.

Para melhor compreensão do problema MPMPpBPTM e de sua complexidade, considere o problema MPpBPTM com dois produtos e um único período de planejamento (1 dia) apresentado na Tab. 1, e o problema MPMPpBPTM com dois produtos e um horizonte de planejamento de dois dias, conforme apresentado na Tab. 2, onde: p_i is the production rate of product i; UD_i is the demand for the product i; UD_{id} is the demand for the product i on day d; UO_i is the maximum amount of product i that can be shipped to outlets; UI_i é o limite de estoque para o produto i in the factory on the first day de planejamento; O is the maximum quantity allowed for shipment of all products to outlets; I é o limite de estoque in the factory para all products in the batch, on the first day de planejamento; Z is the timeout for batch processing; Z_d is the timeout for batch processing on day d.

Table 1: **Benchmark MPpBPTM 02.**

i	1	2		total
p_i	60	40		
UD_i	1000	500		
UO_i	600	600	O	1000
UI_i	3000	2000	I	3000
			Z	100

Table 2: **Benchmark MPMPpBPTM 02.**

i	1	2		total
p_i	60	40		
UD_{i1}	1000	500		
UD_{i2}	500	500		
UO_i	600	600	O	1000
UI_i	3000	2000	I	3000
			Z_1	100
			Z_1	50

No caso do problema MPpBPTM 02, desejamos encontrar o tempo máximo de planejamento em um único período para o lote contendo os produtos 1 e 2. Neste caso, podemos encontrar T, aplicando as equações matemáticas (1) e (2) propostas por Fraga (2024) para solução deste problema:

$$T^* = \lfloor \min\{\min_i\{(\mathrm{UO}_i + \mathrm{UI}_i + \mathrm{UD}_i)/\mathrm{p}_i\}, (\mathrm{O} + \mathrm{I} + \sum_i \mathrm{UD}_i)/\sum_i \mathrm{p}_i\}\rfloor \tag{1}$$

$$T = \min\{T^*, \mathrm{Z}\} \tag{2}$$

Aplicando estas fórmulas, teremos a solução T = 55.

1.1 Mathematical Model

Given that:
UD_i is the demand for the product i;
I is the maximum quantity allowed for additional factory storage of all products in the batch;
UI_i is the maximum quantity allowed for stocking the product i in the factory;
O is the maximum quantity allowed for shipment of all products to outlets;
UO_i is the maximum amount of product i that can be shipped to outlets;
p_i is the production rate of product i;
Z is the timeout for batch processing;
P_i is the amount of product i produced;
D_i is the amount of product i delivered for the demand;
O_i amount of product i shipped to factory outlets;
I_i is the amount of product i that will be stored at the factory;
T is the batch processing time;

We have the problem:

$$max \quad T \tag{3}$$

$s.t.$

$$P_i - \text{p}_i * T = 0 \quad \forall i \tag{4}$$

$$P_i - D_i - O_i - I_i = 0 \quad \forall i \tag{5}$$

$$D_i \leq \text{UD}_i \quad \forall i \tag{6}$$

$$O_i \leq \text{UO}_i \quad \forall i \tag{7}$$

$$\sum_i O_i \leq \text{O} \tag{8}$$

$$I_i \leq \text{UI}_i \quad \forall i \tag{9}$$

$$\sum_i I_i \leq \text{I} \tag{10}$$

$$T \leq \text{Z} \tag{11}$$

$$\min\{\text{UD}_i - D_i, O_i + I_i\} = 0 \quad \forall i \tag{12}$$

$$\min\{\text{O} - \sum_i \{O_i\}, \text{UO}_i - O_i, I_i\} = 0 \quad \forall i \tag{13}$$

$$T, D_i, O_i, I_i \in \mathbb{Z}^+ \quad \forall i \tag{14}$$

where constraints in Eq. (2) relate the quantity produced, P_i, to batch processing time T. Constraints in Eq. (3) calculate the quantity produced, P_i, as a function of the primary variables, D_i, O_i and I_i. Constraints in Eq. (4), (5), and (7) state that the quantity delivered to demand, the quantity shipped to the outlets, and the factory-stocked quantity of each product must be less than their respective known limits. Constraints deined by Eq. (6) and (8) state that both the sum of product quantities sent to the outlets and the sum of product quantities stocked in the factory must be less than their respective maximum allowed values. The restriction in Eq. (9) establishes that there is a batch processing time limit, Z, that must be respected. Equations (10) and (11) define the production distribution priority. And finally, the constraints in Eq. (12) inform the nature of the decision variables.

1.2 Analytical Solution

The method proposed in this paper for solution of the MPpBPTM problem is composed of two distinct parts, which are: a) calculation of the maximum processing time of the predefined multi-product p-batch; and b) calculation of the part of the production used to meet demand, the part of the production sent to outlets and the part of the production stocked in factory. Next, we explain each of these parts in detail.

Calculation of maximum p-batch processing time:

It is possible to split the batch processing time into two time slots:

$$T = T' + T'' \tag{15}$$

where T' is the maximum processing time whose production will be used exclusively to meet the demand.

Then, being $D_i^{'}$ the quantity produced of the product i after T' units of processing time, we have:

$$D_i^{'} - \mathrm{p}_i * T' = 0 \quad \forall i \tag{16}$$

$$D_i^{'} \in Z \quad \forall i \tag{17}$$

and, as $D_i^{'} \leq D_i, \forall i$, we have:

$$D_i^{'} \leq \mathrm{UD}_i \quad \forall i \tag{18}$$

From the Eq. (16) and the restrictions in Eq. (17) and (18), we have:

$$T' \leq \lfloor \mathrm{UD}_i / \mathrm{p}_i \rfloor \quad \forall i \tag{19}$$

And, as we wish maximum value to T', from Eq. (19) we have:

$$T' = \min_i \{ \lfloor \mathrm{UD}_i / \mathrm{p}_i \rfloor \} \tag{20}$$

Observe that there may be a leftover corresponding to the demand not met by the production in the first part of the processing time, T'. This leftover, S_i, will be defined according to the equations in Eq. (21):

$$S_i = \mathrm{UD}_i - D_i^{'} \quad \forall i \tag{21}$$

It is also possible to consider that outlets are extensions of the factory stock. Therefore, it is possible to consider the factory stock of product i and the outlets stock i as single stock for the product i, E_i, so that:

$$E_i = O_i + I_i \tag{22}$$

$$E_i \leq \mathrm{UO}_i + \mathrm{UI}_i \tag{23}$$

$$\sum_i E_i \leq \mathrm{O} + \mathrm{I} \tag{24}$$

$$E_i \in Z \quad \forall i \tag{25}$$

As the time T' was used only for production that will meet the demand, T'' will be the remaining processing time, *i.e.* the maximum processing time used for production that will meet the leftover of demand, sent to the outlets and stored in factory. So we have:

$$E_i + S_i - \mathrm{p}_i * T'' = 0 \quad \forall i \tag{26}$$

Thus, after replacing Eq. (26) in Eq. (23) and applying some algebra, we have:

$$T'' \leq (\mathrm{UO}_i + \mathrm{UI}_i + S_i)/\mathrm{p}_i \quad \forall i \tag{27}$$

And, after replacing Eq. (26) in Eq. (24) and applying some algebra, we have:

$$T'' \leq (\mathrm{O} + \mathrm{I} + \sum_i S_i) / \sum_i \mathrm{p}_i \tag{28}$$

Finally, as T'' is an integer value and we want to maximize this value, then:

$$T'' = \lfloor \min\{ \min_i \{ (\mathrm{UO}_i + \mathrm{UI}_i + S_i)/\mathrm{p}_i \}, (\mathrm{O} + \mathrm{I} + \sum_i S_i) / \sum_i \mathrm{p}_i \} \rfloor \tag{29}$$

So, the analytical method for finding maximum T can be decomposed into 4 steps:
step 1: find T', where:

$$T' = \lfloor \min_i \{ \mathrm{UD}_i / \mathrm{p}_i \} \rfloor \tag{30}$$

step 2: calculate S_i for all products.

$$S_i = \mathrm{UD}_i - \mathrm{p}_i * T' \quad \forall i \tag{31}$$

step 3: find T'', where:

$$T'' = \lfloor \min\{\min_i\{(\mathrm{UO}_i + \mathrm{UI}_i + S_i)/\mathrm{p}_i\}, (\mathrm{O} + \mathrm{I} + \sum_i S_i)/\sum_i \mathrm{p}_i\}\rfloor \tag{32}$$

step 4: calculate T, where:

$$T^* = T' + T'' \tag{33}$$

$$T = \min\{T^*, \mathrm{Z}\} \tag{34}$$

Note that, as we reduce the value of T', the consequent reduction of $D_i^{'}$ will be directly attributed to the leftover of demand S_i and, therefore, there will be no change in the optimal solution if it is considered to T', a value between 0 and the value found in step 1. Thus it is possible to use a simplification of the method presented, making $T' = 0$. In this case $S_i = \mathrm{UD}_i, \forall i$. The simplified version of the method is presented below:

step 1: find T^*, where:

$$T^* = \lfloor \min\{\min_i\{(\mathrm{UO}_i + \mathrm{UI}_i + \mathrm{UD}_i)/\mathrm{p}_i\}, (\mathrm{O} + \mathrm{I} + \sum_i \mathrm{UD}_i)/\sum_i \mathrm{p}_i\}\rfloor \tag{35}$$

step 2: calculate T, where:

$$T = \min\{T^*, \mathrm{Z}\} \tag{36}$$

This simplification of the method demonstrates that the value calculated to T does not depend on the way the production of the product i, P_i, is distributed between service to demand, D_i, outlets, O_i, and factory's stock, I_i.

Full solution:

We can determine now the values of D_i, O_i and I_i, $\forall i$, simply attending the priorities for distribution. The following three algorithms can be used to find a complete solution: The Algorithm (1) is used for finding an optimimal value for T. Then an initial production distribution can be done by the Algorithm (2). This distribution is made without considering the restrictions defined for the set of products. SO and SI represent, respectively, the total amount that can be sent to the outlets and the total amount that can be stocked in factory after the initial distribution. If this values are negative, this indicates that the proposed initial distribution is not a feasible distribution. So, after applying Algorithm (2), if $SO \geq 0$ and $SI \geq 0$, solution found is an optimal solution. However, if $SO < 0$ or $SI < 0$, solution is not yet feasible. But as the solution found to T is feasible to the problem defined in Eq. (3) to (14), we have the following statements:

$$if \quad SO < 0 \quad then \quad SI > 0 \quad and \quad |SO| < SI$$

$$if \quad SI < 0 \quad then \quad SO > 0 \quad and \quad |SI| < SO$$

Algorithm 1 Solving MPpBPTM problem | Part 01 - find an optimal T to the problem defined in (3) to (14).

Require: $\mathrm{UD}_i, \mathrm{UO}_i, \mathrm{UI}_i, \mathrm{p}_i, \forall i, \mathrm{O}, \mathrm{I}, \mathrm{Z}$
$T^* \leftarrow \lfloor \min\{\min_i\{(\mathrm{UO}_i + \mathrm{UI}_i + \mathrm{UD}_i)/\mathrm{p}_i\}, (\mathrm{O} + \mathrm{I} + \sum_i \mathrm{UD}_i)/\sum_i \mathrm{p}_i\}\rfloor$
$T \leftarrow \min\{T^*, \mathrm{Z}\}$
Return: T.

Algorithm 2 Solving MPpBPTM problem | Part 02 - calculate D_i, O_i and I_i, $\forall i$, ignoring the restrictions for the set of products of the batch.

Require: $\mathrm{UD}_i, \mathrm{UO}_i, \mathrm{UI}_i, \mathrm{p}_i, \forall i, \mathrm{O}, \mathrm{I}, \mathrm{Z}, T$.
for all i **do**
$E_i \leftarrow T * \mathrm{p}_i$
$D_i \leftarrow \min\{\mathrm{UD}_i, E_i\}$
$E_i \leftarrow E_i - D_i$
$O_i \leftarrow \min\{\mathrm{UO}_i, E_i\}$
$E_i = E_i - O_i$
$\mathrm{UO}_i = \mathrm{UO}_i - O_i$
$I_i = E_i$
$\mathrm{UI}_i = \mathrm{UI}_i - I_i$
end for
$SO = \mathrm{O} - \sum_i O_i$
$SI = \mathrm{I} - \sum_i I_i$
Return: $D_i, O_i, I_i, E_i, \mathrm{UO}_i, \mathrm{UI}_i, \forall i, SO, SI$.

Therefore it is possible to meet the restrictions by applying the Agorithm (3). This algorithm redistributes the production of factory's stock to the outlets or vice versa, respectively, due to having a SI or SO negative.

Algorithm 3 Solving MPpBPTM problem | Part 03: redistribute production to comply with limitation restrictions for the batch set.

Require: $O_i, I_i, \text{UO}_i, \text{UI}_i, \forall i, SO, SI.$
  if $SO < 0$ **then**
    for all i **do**
      $O_i \leftarrow O_i - \min\{O_i, UI_i, |SO|\}$
      $I_i \leftarrow I_i + \min\{O_i, UI_i, |SO|\}$
      $SO \leftarrow SO + \min\{O_i, UI_i, |SO|\}$
      if $SO = 0$ **then** break looping for;
      end if
    end for
  end if
  if $SI < 0$ **then**
    for all i **do**
      $I_i \leftarrow I_i - \min\{I_i, UO_i, |SI|\}$
      $O_i \leftarrow O_i + \min\{I_i, UO_i, |SI|\}$
      $SI \leftarrow SI + \min\{I_i, UO_i, |SI|\}$
      if $SI = 0$ **then** break looping for;
      end if
    end for
  end if
  Return: $O_i, I_i, \forall i.$

2. RESULTS AND DISCUSSION

To test the developed model and analytical solution method, we created a solver in LINGO named MPpBPTM.lng and also a solver in C++ named COPSolver: library for solving the multi-product p-batch processing time maximization problem, both available at codeocean and GitHub. A brief explanation of the software and links for downloads can be found in the paper of Fraga (2023). The tests were performed on a notebook with an Intel i7 processor. We tested the solvers developed for the benchmarks presented on Tab. 3, 4 and 5, and for randon benchmarks generated by the following functions:

$$\text{p}_i = \text{rand}()\%30 + 10 \tag{37}$$

$$\text{UD}_i = \text{rand}()\%3000 + 800; \tag{38}$$

$$\text{seed1} = \text{rand}()\%3000 + 500; \tag{39}$$

$$\text{seed2} = \text{rand}()\%5000 + 1000; \tag{40}$$

$$\text{O} = \text{N}/2 * \text{seed1}; \tag{41}$$

$$\text{UO}_i = \text{rand}()\%(\text{seed1} - 500) + 500; \tag{42}$$

$$\text{I} = \text{N}/2 * \text{seed2}; \tag{43}$$

$$\text{UI}_i = \text{rand}()\%(\text{seed2} - 1000) + 1000; \tag{44}$$

$$\text{Z} = 100; \tag{45}$$

Randomly generated benchmarks were named RMPpBPTM N, being N the number of products. Seeking to enable the reproduction of the results, in the computational construction of the benchmarks we used the function srand((unsigned) source), where source is a defined value. To build the results presented in this work, we used source=0. Files containing benchmarks used for the tests performed and all results can be found at tbfraga.github.io/COPSolver/benchmarks.

Table 6 presents the results obtained by applying the analytical method (COPSolver) and the LINGO solver for the solution of the previously presented benchmarks. Tables (7) and (8) shows the production distribution for MPpBPTM 3 and 10 found by both COPSolver and LINGO solver. It is important to point out that the LINGO solver was developed with the purpose of validating the results found by the proposed analytical method. It is possible to verify that both solvers are capable of finding optimal solutions for the proposed benchmarks very quickly, even for very large-scale problems. However, as the analytical method is a polynomial time complexity method, it is much more efficient. In addition, the modular structure of the library developed for COPSolver allows the easy application of the proposed solution method when developing new heuristics for solving MPpBS problems, making this tool a significant technological contribution to the production management of some industries as well as for researchers in the area. Since we have considered here a static problem, in a future work we will study the complexities of solving the problem dynamically and try to propose new solution methods if needed.

Table 3: **Benchmark MPpBPTM 2.**

i	1	2	total
p_i	60	40	
UD_i	1000	500	
UO_i	600	600	1000
UI_i	3000	2000	3000
		Z	100

Table 4: **Benchmark MPpBPTM 3.**

i	1	2	3	total
p_i	60	40	50	
UD_i	1000	500	800	
UO_i	600	600	600	1500
UI_i	3000	2000	1000	3500
			Z	100

Table 5: **Benchmark MPpBPTM 10.**

i	1	2	3	4	5	6	7	8	9	10	total
p_i	60	40	50	40	30	50	60	10	20	40	
UD_i	1000	500	800	500	400	500	2000	300	500	1000	
UO_i	600	600	600	1500	300	200	500	800	0	200	3000
UI_i	3000	2000	1000	800	3000	1000	400	300	200	0	5000
										Z	100

Table 6: **Results obtained with the LINGO solver and the analytical method.**

problem	LINGO solver	time (s)	analytical method	time (s)
MPpBPTM 2	55	0.03	55	< 0.01
MPpBPTM 3	48	0.03	48	< 0.01
MPpBPTM 10	30	0.05	30	< 0.01
RMPpBPTM 20	100	0.06	100	< 0.01
RMPpBPTM 50	98	0.08	98	< 0.01
RMPpBPTM 100	98	0.11	98	< 0.01
RMPpBPTM 1,000	78	1.20	78	< 0.01
RMPpBPTM 2,000	70	3.25	70	< 0.01
RMPpBPTM 5,000	70	15.12	70	< 0.02
RMPpBPTM 10,000	70	55.19	70	< 0.04

Table 7: **Results obtained with the LINGO solver and COPSolver for the MPpBPTM 3.**

product	production	delivered	send to outlets	stocked in factory
P1	2,880	1,000	300	1,580
P2	1,920	500	600	820
P3	2,400	800	600	1,000

3. CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORKS

In this paper we presented the multi-product p-batch processing time maximization problem, as well as a mathematical model and an exact analytical solution method for this problem. The mathematical model and analytical method were tested, respectively, by a solver developed with the LINGO software, from LINDO Systems, and with a solver developed

Table 8: **Results obtained with the LINGO solver and COPSolver for the MPpBPTM 10.**

product	production	delivered	send to outlets	stocked in factory
P1	1,800	1,000	400	400
P2	1,200	500	600	100
P3	1,500	800	600	100
P4	1,200	500	700	0
P5	900	400	300	200
P6	1,500	500	200	800
P7	1,800	1800	0	0
P8	300	300	0	0
P9	600	500	0	100
P10	1,200	1000	200	0

in C++ language. Optimum results were found for all proposed benchmarks, very quickly, even in the case of very large-scale problems, which demonstrates the efficiency of the proposed method. As in this paper we considered a static problem, in a future work we will study the complexities that arise when considering the same problem in a dinamic scenario. We will also verify if there is the possibility of extending the proposed analytical method to solve a class of integer programming problems with similar characteristics to the studied problem.

4. CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

T.B. Fraga: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing – original draft, Writing – review & editing, Supervision, Project administration. Í.R.B. Aquino: Investigation. R.C.S. Menêzes: Investigation.

5. ACKNOWLEDGMENTS

We are enormously grateful to Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and to Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for the financial support provided to our projects. We also thank LINDO systems team for the LINGO software license, without which this work would not have been possible and the to the owner of the company in the plastics sector, who allowed us to learn about his company's production process. Finally, we would like to thank Pró-reitoria de Extensão e Cultura da UFPE (PROExC) and the Research Director of Propesqi (Pró-reitoria de Pesquisa e Inovação da UFPE) for their support and recognition of our work, and dear Professors Antônio José da Silva Neto and João Flávio Vieira de Vasconcellos from IPRJ/UERJ, who contributed significantly to the formation of essential skills for the development of our projects. We also thank my co-worker Marcos Luiz Henrique, for having helped by evaluating the mathematical model and solution method proposed in this paper.

6. REFERENCES

Altay Guvenir, H. and Erel, E., 1998. "Multicriteria inventory classification using a genetic algorithm". *European Journal of Operational Research*, Vol. 105, No. 1, pp. 29–37.

Balaji, K. and Kumar, V.S., 2014. "Multicriteria inventory abc classification in an automobile rubber components manufacturing industry". *Procedia CIRP*, Vol. 17, pp. 463–468. Variety Management in Manufacturing.

Barbosa de Paula, N.O., de Araújo Costa, I.P., Drumond, P., Ângelo Lellis Moreira, M., Simões Gomes, C.F., dos Santos, M. and do Nascimento Maêda, S.M., 2022. "Strategic support for the distribution of vaccines against covid-19 to brazilian remote areas: A multicriteria approach in the light of the electre-mor method". *Procedia Computer Science*, Vol. 199, pp. 40–47. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19.

de Moura Pereira, D.A., Diniz, B.P., Araújo, G.N., Araújo, A.C., de Siqueira Silva, M.J., Neto, J.C., Araújo, J.M.B., Tomaz, P.P.M. and dos Santos, M., 2023. "Development of strategic planning of a financial education company in brazil: an approach based on the new multicriteria decision analysis method s.w.o.t-d.m.s". *Procedia Computer Science*, Vol. 221, pp. 681–688. Tenth International Conference on Information Technology and Quantitative Management (ITQM 2023).

Ebrahimi, A.H., Johansson, P.E., Bengtsson, K. and Åkesson, K., 2014. "Managing product and production variety – a language workbench approach". *Procedia CIRP*, Vol. 17, pp. 338–344. Variety Management in Manufacturing.

Flores, B.E., Olson, D.L. and Dorai, V., 1992. "Management of multicriteria inventory classification". *Mathematical and Computer Modelling*, Vol. 16, No. 12, pp. 71–82.

Flores, B.E. and Whybark, D.C., 1986. "Multiple criteria abc analysis". *International Journal of Operations & Production Management*, Vol. 6, No. 3, pp. 38–46.

Fraga, T.B., 2023. "COPSolver: Open source software for solving combinatorial optimization and other decision problems

— library for solving the multi-product p-batch processing time maximization problem". *Software Impacts*, Vol. 18, p. 100592.

Fraga, T.B., 2024. "COPSolver: Open source software for solving combinatorial optimization and other decision problems — library for solving the multicriteria classification problem". *In press*.

Kiran, D.R., 2019. *Production Planning and Control: A Comprehensive Approach*. Butterworth-Heinemann, Elsevier.

Lolli, F., Ishizaka, A. and Gamberini, R., 2014. "New ahp-based approaches for multi-criteria inventory classification". *International Journal of Production Economics*, Vol. 156, pp. 62–74.

Mariano Ribeiro, J.V., Sussel Gonçalves Mendes, T., Jorge Coelho Simões, S., Gili Massi, K., Ivo Mioni Camarinha, P. and Cassiano Ferreira, C., 2023. "Strategic landscape analysis relating multicriteria analysis and socioeconomic and environmental context to define potential areas for active restoration in são paulo, brazil". *Journal of South American Earth Sciences*, Vol. 130, p. 104561.

Odu, G.O., 2019. "Weighting methods for multi-criteria decision making techniquex". *J. Appl. Sci. Environ. Manage.*, Vol. 23, No. 8, pp. 1449–1457.

Saaty, R.W., 1987. "The analytic hierarchy process - what it is and how it is used". *Mathl Modelling*, Vol. 9, No. 3-5, pp. 161–176.

Singhal, J., Bitran, G.R. and Dasu, S., 2013. *Production Management*, Springer US, Boston, MA, pp. 1173–1182.

Williams, T.M., 1984. "Stock control with sporadic and slow-moving demand". *Journal of the Operational Research Society*, Vol. 35, No. 10, pp. 939–948.

7. COPYRIGHT LIABILITY