Dynamické programovanie a problém riadenia výroby a zásob

Dynamic programming and the problem of inventory production management

Monika Ferenčáková[1]

Abstrakt
Príspevok sa zameriava na aplikáciu dynamického programovania v oblasti riadenia výroby a zásob. Predstavuje teoretické východiská a metódy, vrátane Bellmanovho princípu optimality rekurentných vzťahov, ktoré umožňujú efektívne riešenie viacstupňových rozhodovacích procesov. Osobitná pozornosť je venovaná modelu riadenia výroby a zásob, ktorý zohľadňuje výrobné a skladovacie kapacity, ako aj časovo variabilné náklady a dopyt. V praktickej časti je analyzovaný viacperiodický problém plánovania výroby a zásob s cieľom minimalizovať celkové náklady a obmedziť nadbytočné zásoby, čím sa znižuje materiálové a energetické plytvanie. Na riešenie úlohy bol využitý nástroj Python, Výsledkom je optimálny výrobný plán a porovnanie rôznych alternatív riešenia. Výsledky potvrdzujú vysokú efektivitu dynamického programovania pri optimalizácii ekonomických aj environmentálne udržateľných rozhodovacích procesov.

Kľúčové slová
dynamické programovanie, riadenie zásob, rekurentné vzťahy, udržateľné plánovanie

Abstract
The paper focuses on the application of dynamic programming in the field of production and inventory management. It presents theoretical bases and methods, including Bellman's principle of optimality of recurrent relationships, which enable effective solution of multi-stage decision-making processes. Particular attention is paid to the production and inventory management model, which takes into account production and storage capacities as well as time-varying costs and demand. In the practical part, the multi-period problem of production and inventory planning is analyzed in order to minimize the total cost and limit excess inventory, thus reducing material and energy waste. The Python tool was used to solve the problem, resulting in an optimal production plan and a comparison of different solution alternatives. The results confirm the high efficiency of dynamic programming in optimizing economic and environmentally sustainable decision-making processes.

Keywords
dynamic programming, inventory management, recursive relations, sustainable planning

JEL classification
C61, Q01, M11

1 Introduction

Modern operations research and applied mathematics often encounter problems that require a series of decisions made over time, each of which affects both future possibilities and

[1] Bratislava University of Economics and Business, Faculty of Economic Informatics, Department of operational research and econometrics, Dolnozemská cesta 1, 852 35 Bratislava, monika.ferencakova@euba.sk.

overall outcomes. Typical situations arise in inventory management, production scheduling, investment planning or maintenance system management, where the most suitable choice in each step depends on both the current state and the expectation of future events. Effectively addressing such problems requires a framework that considers their own time structure and allows decisions at every step to be based not only on immediate but also on future consequences. This is exactly the framework that dynamic programming provides.

Dynamic programming (*DP*), introduced to the scientific world by Richard Bellman in the 1950s, has become a key method for multi-stage decision-making processes. The essence lies in the division of a complex problem, potentially involving hundreds or thousands of sequential decisions, into a series of simpler subproblems, each of which can be optimally solved and afterwards solutions can subsequently be combined to solve the original problem as a whole (Hillier, Lieberman, 2000). This divide and conquer methodology not only brings significant computational benefits, but also conceptual clarity. Instead of testing every single sequence of decisions, the DP takes advantage of the fact that the state of the system at any given point in time contains all the historical information required for further planning. By focusing on a small number of key variables that describe each state and systematically moving backward from a known outcome at the end of the planning horizon, the optimal strategy can be compiled step by step. This logical structure also makes DP uniquely useful in modelling situations with limited resources, randomness, and complex interconnected decisions.

In the context of current trends, dynamic programming is also applied in finding a balance between economic efficiency and environmental sustainability. Optimizing inventory and production through dynamic programming can directly contribute to reducing energy intensity, reducing overproduction, and using resources more efficiently. In practice, minimizing excess inventory means not only lower costs, but also less environmental burden, less storage, fewer losses, and more economical logistics.

2 Theoretical research

The founder of (DP) is considered to be Richard Bellman, who published the groundbreaking work "***Dynamic Programming"*** in 1957 (Bellman, 1957). Bellman identified the basic principle of optimality, according to which every sub-policy of the optimal solution must be optimal in itself. This principle led to the formulation of the Bellman recurrent equation, which makes it possible to solve problems in the direction from the final state to the initial state.

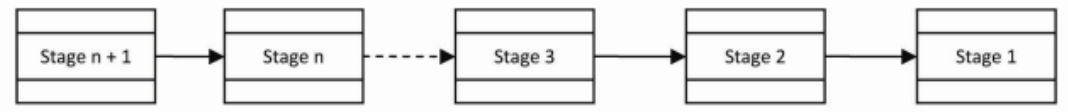
Even before Bellman, Frank Ramsey (1937) dealt with deterministic multiperiodic optimization decision-making in the context of economic modelling of intertemporal consumption, thus implicitly formulating the idea of optimality in time. In the following years, Tjalling Koopmans (1949), who dealt with the allocation of resources in transport systems, also made a significant contribution. Both authors foreshadowed the basic features of dynamic thinking in optimization tasks.

In Slovakia, the theory of dynamical models was significantly developed by Laščiak et al. (1985), authors in the publication Dynamic Models formally described multi-stage decision-making processes as a sequence of stages in which the state of the system is controlled by decisions and is evaluated through a criterion function. They emphasized that the total value of the process is the sum of the elementary contributions of the individual stages and that this process should be mathematically consistent and deterministically describable.

From a formal point of view, the contribution was made by Madaras (2005), who analyzed the structure and solvability of recurrent relations – the basic computational mechanism of the DP. He distinguished between linear and nonlinear recursions, and described their applicability in combinatorial and decision-making problems. These theoretical

frameworks are followed by Goga (2011), who in monograph "***Economic Dynamics***" provides a systematic classification of multi-stage decision-making processes. It identifies two main classes: deterministic and stochastic. In our case, we focus on deterministic, which are suitable for exact modelling of e.g. production and supply processes. A critically important prerequisite for a process to be suitable for solution by the dynamic programming method is its Markov structure. The multi-stage character of dynamic programming can be schematically represented as a sequence of stages in which states and decisions determine the transition to the next phase. This structure is shown in Figure 1.

Figure 1 – Stages of the Dynamic Programming Problem



Source: Selvanathan (2015).

The diagram illustrates the backward solution from the terminal stage to the initial stage, which is the basis of Bellman's recursion used in the next parts of the paper.

In recent years, DP has expanded into new areas, including approximate algorithms and machine learning. Powell (2011) introduced the concept of "*approximate dynamic programming*", which allows the application of DP principles even in cases where the state space is too large or complexly structured. The issue of supply with imperfect information about future demand and inventory returns is analyzed by Gallego and Özer (2001), respectively. Topan et al. (2018) show how DP supports decision-making in real supply chains with uncertainty and imperfect signals.

In addition to economic applications, dynamic programming is also applied in the field of sustainability. Studies show that it can link economic, environmental, and social goals and thus contribute to reducing the ecological footprint (Selvanathan, 2015). In environmental economics, dynamic models are employed in the formulation of climate policies and resource management, helping to balance short-term costs with long-term sustainability (Karp & Traeger, 2024). The development trajectory of dynamic programming highlights its theoretical versatility and practical robustness in a wide range of decision-making areas under uncertainty and constraints.

3 Methodology

In this section, we present a methodological framework based on the principles of deterministic dynamic programming, as formulated by Laščiak et al. (1985) and systematized by Goga (2011) and practically introduced by Anderson et al. (2010).

According to Laščiak et al. (1985), a dynamic process can be broken down into discrete stages, each characterized by a specific state of the system. Within each stage, there is a functional dependence between the state variables, the decision variable, and the instantaneous value of the criterion function. The total value of the process is then defined as the sum of elementary contributions for all stages. This structured approach enables the modeling of multi-stage decision-making processes over time and can be adapted to incorporate environmental or social aspects alongside economic objectives.

As Goga (2011) states, multi-stage processes can be divided into deterministic and stochastic. This paper focuses on a deterministic class where all quantities are known in advance and do not change due to chance. Based on Laščiak et al. (1983), four basic assumptions can

be formulated that must be met in order for the decision-making process to be effectively modelled through deterministic dynamic programming:

1. **Discreetness of time**
   - The process can be divided into a finite number of stages, which represent discrete time periods $n = 1,2,\dots,N$ and are ascendingly arranged.
   - Each period is separately analysable, and decision-making activities are tied to these time units.
2. **State representation of stages**
   - Each stage is clearly determined by a state variable x_n, which expresses the amount of on-hand inventory at the beginning of period n. The state reflects the cumulative results of previous decisions and serves as a starting point for current optimization.
3. **Management through decision variables**
   - At each stage, a decision is made that affects the transition of the system to the next state. These decisions belong to a set of permissible decisions given by technological, capacity, or other limitations.
4. **Additiveness of the criterion function**
   - The overall target indicator (e.g. costs) is expressed as the sum of the sub-contributions from the individual stages, while in each stage the criterion function depends only on the state of play and the decision. This structure serves as the foundation for applying recurrent relationships in solving an optimization problem.

General formulation of the deterministic inventory management model

For the purposes of further analysis, we will start from a deterministic multi-period model of production and inventory management. It aims to minimise overall costs over the planned horizon, considering both production and storage costs. The optimization task takes the following form:

Target Feature:

$$min \sum_{n=1}^{N} (C_n d_n + H_n x_n) \qquad (3.1)$$

The goal is to minimize the total production and storage costs over the entire planning horizon.

Limits:

- Stock balance (transition of stocks):

$$x_n = x_{n-1} + d_n - D_n \qquad n = 1, \dots, N \qquad (3.2)$$

- Production capacity:

$$0 \leq d_n \leq P_n \qquad (3.3)$$

- Storage capacity:

$$0 \leq x_n \leq W_n \qquad (3.4)$$

- Initial stocks that are predetermined at the beginning of the first stage.

Where:

- N - number of periods (stages),
- D_n - demand in the period n,
- x_n - status variable, the amount of stock in stock at the end of the stage n,
- d_n - production quantity (decision variable) at stage n,
- P_n - maximum number of units possible to produce in stage n (production capacity),
- W_n - storage capacity (maximum stocks at the end of the stage n),
- C_n - unit cost of production in the stage n,
- H_n - unit storage cost per unit of inventory at the end of stage n.

The formal basis of dynamic programming lies in the so-called **Bellman's principle of optimality**, which expresses that "regardless of the initial state and the decision made, the subsequent part of the strategy must represent the optimal solution to the remaining subproblem" (Bellman, 1957; Goga, 2011). From a mathematical point of view, the optimality principle can be formalized by means **of a recurrent relation**, which takes the form as follows:

$$f_n(x_n) = \min_{d_n \epsilon [0,P_n]} \{C_n d_n + H_n x_n + f_{n-1}(x_n + d_n - D_n)\} \qquad (3.5)$$

Where:

- x is the decision in the stage n,

Recurrent relationships represent the computational core of dynamic programming, which enables systematic retrospective evaluation of optimal decisions. Their significance lies in the fact that for each possible stock level at the beginning of a given stage, a minimum value of the total costs from this position up to the end of the planning horizon is calculated. This principle ensures that the decisions taken in each period are not isolated but consider the long-term consequences. As a result, dynamic programming not only optimizes local decisions, but also ensures that the entire decision-making sequence is optimally solved globally.

Madaras (2005), in his work on recurrent relationships, lists the different types of deterministic relationships by which the course of a process can be formalized, emphasizing the importance of monotony and finitude of the state space for an effective solution.

Dynamic programming has a wide range of applications across the areas of operations research and industrial optimization. The canonical tasks solved by the DP method include:

- **Backpack problem** (knapsack problem), where the goal is to optimize the selection of elements with a given value and weight,
- **The task of optimal allocation of production capacities**, in which the timing and volume of production are decided,
- **Optimal allocation of a limited resource**, typical for decision-making in public finance or marketing,
- **Multi-stage decision-making in inventory management**, with deterministic and stochastic demand, and with limited storage and production capacity, is explored in detail in the work of Clark and Scarf (1960) and further elaborated for environments with advanced or incomplete demand information by Gallego and Özer (2001) as well as Topan et al. (2018).

Modelling the supply process through DP requires the identification of state variables (inventory in stock), decision variables (quantity of units produced or ordered), transition relationships (inventory update according to demand), and cost structure (inventory costs,

production costs, shortage penalties). Such a framework enables the precise formulation of a criterion function and the application of Bellman recursion.

Modern models of dynamic programming confirm the suitability of this approach also for advanced supply systems, where the condition of the equipment or advanced demand information is considered. Real applications are solved by Gallego and Özer (2001), who analyse the value and impact of advanced demand information (ADI) on order optimization. If the information is incomplete or erroneous, effective combinations of inventory returns and ADI management are also explored in the papers of Topan et al. (2018) and Gayon, Benjaafar and de Véricourt (2008), which show that dynamic programming makes it possible to find the optimal policy even with uncertain or imperfect signals about future demand.

A specific case with a high degree of practical relevance is the deterministic model of production inventory management in discrete periods. It is this type of problem that will be analysed in the following sections, starting from the methodology described in the publication by Anderson et al. (2010). These examples provide a practical illustration of the general Bellman structure and demonstrate the power of DP in optimizing economically relevant systems.

4 Example of a deterministic multi-period production inventory management problem

Consider a deterministic multi-period production and inventory management problem, where the expected demand for the product is known for each period. The goal of the decision-making process is to develop a production plan that ensures smooth coverage of this demand while minimizing the total cost of production and storage.

The basic cost items of the model are variable ***production costs***, which may vary between individual periods depending on technological or capacity conditions, and ***inventory storage costs***, which arise if production in each period exceeds current demand. It is assumed that in each period, exactly one production and adjustment operation takes place, with fixed costs that are constant across all periods. For this reason, they do not affect the optimization structure and can be omitted from the model. The proposed model (Anderson et al., 2010) further allows for the consideration of time-varying constraints on production and storage capacity, which reflects the reality of production plants in practice or warehouses with different availability over time. Such flexibility makes the model robust enough for application in environments with fluctuating resources.

Dynamic programming is suitable for solving this type of task, as it allows to divide complex decision-making into a series of interconnected steps efficiently. At each stage, decisions are made about production and inventory levels, which affect not only current but also future decisions and costs. The principle of reverse induction gradually builds the optimal strategy from the end of the planning horizon towards the beginning, optimizing the sum of immediate and expected future costs in each step.

The input data for the model solution, including demand, production, and storage capacities, as well as unit costs in each period, are summarized in Table 1.

Table n. 1 – Input data

Day	Demand D_n	Production capacity P_n	Storage capacity W_n	Production cost C_n	Storage cost H_n
Monday	2	5	6	175	30
Tuesday	3	2	6	150	30
Wednesday	3	3	5	200	40
Thursday	4	4	5	150	20
Friday	4	1	6	175	10

Source: own processing

Friday – Stage 1:

In the first stage, the problem of decision-making is simplified to the optimization of immediate costs, as other future periods no longer exist. Thus, in this case, the value function consists exclusively of the sum of the production and storage costs that arise on Friday. The model is defined in such a way that the decision variable d_5 expresses the volume of production and the state variable x_5 represents the stock at the end of the period. The recurrent function for the terminal stage takes the form:

$$\min \mathrm{r}_5(x_5, d_5) = 175d_5 + 10(x_4 + d_5 - 4)$$

With limitations:

$$0 \leq d_5 \leq 1$$
$$0 \leq x_5 \leq 5$$
$$0 \leq x_4 + d_5 \leq 6$$
$$x_4 + d_5 \geq 4$$

where the last inequality ensures that Friday's demand is covered $D_5 = 4$. Since this is the last period, other future costs no longer exist and the value function does not contain future recursion, it only summarizes the direct production and storage cost from the output state. Production has its capacity limitations, which makes it possible to go through a finite set of permissible situations where 3 possible situations occur.

Insufficient stock: If the initial state is $x_4 \leq 2$, even with maximum production, it is not possible to meet the demand coverage requirement. Such conditions are therefore marked as inadmissible.

Marginal stock if $x_4 = 3$, it is necessary to replenish the stock by producing one unit to meet the demand. In such a case, only production costs of 175 € are incurred.

$$\mathrm{r}_5(x_5, d_5) = 175 * 1 + 10(3 + 1 - 4) = 175$$

Sufficient supply: with a stock of at least four units ($x_4 \geq 4$) it is possible to cover the demand without additional production. With a stock equal to demand ($x_4 = 4$) it is possible to cover the need of the day exclusively from stocks. In the event that no production takes place ($d_1 = 0$), the situation at the end of the period is zero and the total costs are:

$$r_5(x_5, d_5) = 175 * 0 + 10(4 + 0 - 4) = 0$$

Alternative decision with production involvement ($d_1 = 1$) leads to the total cost of 185 €. In this case, one unit of inventory remains in stock, but the result is disadvantageous from a cost point of view, as there is a combination of additional production and storage costs. With initial stock exceeding demand ($x_5 = 5$) the need can be covered without production activity. If production is not carried out ($d_5 = 0$), one unit remains in stock, and the costs would be 10 €. If the production of a single unit were to take place ($d_5 = 1$), the result is a condition of two surplus pieces, and the total cost would be 195 €.

The analysis of the above cases shows that the optimal strategy in the terminal stage prefers variants without redundant production and with minimization of end inventory, thus eliminating unnecessary increases in production and storage costs. In any case, the production decision that ensures that demand is covered and minimizes the sum of production and output warehouse costs is automatically selected. At the same time, the minimization of excess inventory set up in this way reduces the need for storage and the associated energy consumption and material waste, which is more sustainable in the long run.

Thursday – Stage 2:

In the second stage, the decision-making process is complicated by the fact that the optimization no longer only considers the immediate production and storage costs, but also the expected costs from the next terminal stage. This procedure represents a practical application of Bellman's principle of optimality, according to which each sub-decision must form an optimal solution for the rest of the planning horizon. Therefore, the recurrent function for Thursday is defined as:

$$\min r_4(x_4, d_4) = 150d_4 + 20(x_4 + d_4 - 4) + r_5(x_5, d_5)$$

With limitations:

$$0 \leq d_4 \leq 4$$
$$0 \leq x_4 \leq 5$$
$$0 \leq x_4 + d_4 \leq 5$$
$$x_4 + d_4 \geq 4$$

The last condition ensures that the minimum demand needed on Thursday is covered ($D_4 = 4$). If the resulting stock level does not allow Friday's demand to be covered, the solution is referred to as inadmissible.

On Thursday, the recurrent function explicitly accumulates the immediate production and storage cost, synchronizing each potential future state with the already calculated minimum value of the function for the next stage. This composition guarantees overall optimality, as it considers the deterministically occurring further course of the system, not just the local/current state. The calculation would continue in a similar way for the other days of production, but this calculation would be computationally lengthy.

For example with a relatively low level of production, a manual calculation of the optimal solution is also possible and clear. However, in the environment of large-scale enterprises, the number of variants, statuses and possible decisions is growing rapidly. This makes the manual procedure too lengthy, error-prone, or even unfeasible. In these cases, it is necessary to use

modern calculation procedures and programs. The Python programming language was used to solve the task, which provides tools for efficient processing of combinatorial and optimization problems. The algorithm systematically generates permissible production decisions, evaluates them through recurrent relationships and compiles an optimal production plan. The results of the calculation are presented in Figure 2.

Figure n.2 – Optimal production plan

```
Deň       Začiatočný sklad  Výroba  Dopyt   Konečný sklad   Náklad (€)
-------------------------------------------------------------------------
Pondelok  3                 3       2       4               645
Utorok    4                 2       3       3               390
Streda    3                 3       3       3               720
Štvrtok   3                 4       4       3               660
Piatok    3                 1       4       0               175
-------------------------------------------------------------------------
SPOLU                                               2590 €
```

Source: output from Python

The table shows that production is planned for each day not only according to the current daily demand, but also in regard to the level of inventories and the expected needs of the next days. For example, the final stock never falls below the necessary value, ensuring continuity of production on the next day. The highest circulation is recorded on Wednesday, which is often related to higher production or storage costs. On the contrary, Friday ends with zero inventory, eliminating unnecessary storage. This end-zero state is also advantageous from a sustainability point of view, as it reduces unnecessary inventory holding and the associated operating losses.

The sum of all costs in the table represents an optimal total weekly cost of 2 590 €, which would be significantly difficult to obtain by manual calculations. Thus, Python provides an efficient and accurate tool for building a production plan for complex assignments of this type. Although we have obtained an optimal production plan with the lowest possible cost using Python, the calculation algorithm is also able to generate multiple permissible production plans that meet all the specified conditions of the model, as shown in Table 2. These alternative solutions may have a different distribution of production or supply on individual days, but they will meet both demand coverage and any capacity constraints.

Table n.2 – Permissible production plans

N.	***Production plan***	***Stock***	***Production costs***	***Storage costs***	***Total costs***	***Choice of production plan***
1	(3, 2, 3, 4, 1)	[3, 4, 3, 3, 3, 0]	2 200	390	**2 590**	***Optimal prod. plan***
2	(4, 1, 3, 4, 1)	[3, 5, 3, 3, 3, 0]	2 225	420	2 645	
3	(4, 2, 2, 4, 1)	[3, 5, 4, 3, 3, 0]	2 175	450	2 625	
4	(4, 2, 3, 3, 1)	[3, 5, 4, 4, 3, 0]	2 225	490	2 715	
5	(4, 2, 3, 4, 0)	[3, 5, 4, 4, 4, 0]	2 200	510	2 710	
6	(5, 0, 3, 4, 1)	[3, 6, 3, 3, 3, 0]	2 250	450	2 700	
7	(5, 1, 2, 4, 1)	[3, 6, 4, 3, 3, 0]	2 200	480	2 680	
8	(5, 1, 3, 3, 1)	[3, 6, 4, 4, 3, 0]	2 250	520	2 770	
9	(5, 1, 3, 4, 0)	[3, 6, 4, 4, 4, 0]	2 225	540	2 765	

10	(5, 2, 1, 4, 1)	[3, 6, 5, 3, 3, 0]	2 150	510	2 660	
11	(5, 2, 2, 3, 1)	[3, 6, 5, 4, 3, 0]	2 200	550	2 750	
12	(5, 2, 2, 4, 0)	[3, 6, 5, 4, 4, 0]	2 175	570	2 745	
13	(5, 2, 3, 2, 1)	[3, 6, 5, 5, 3, 0]	2 250	590	2 840	
14	(5, 2, 3, 3, 0)	[3, 6, 5, 5, 4, 0]	2 225	610	2 835	

Source: own processing based on Python output

Such an approach allows management not only to find the optimal solution, but also to analyse potential alternatives and respond flexibly to changes in operations or in the supply chain.

5 Conclusion

Nowadays, dynamic programming is increasingly implemented in complex systems with high levels of uncertainty and variability in inputs. Examples include inventory management in an environment with imperfect demand information and the possibility of returning inventory (Topan et al., 2018), models distinguishing between multiple customer classes and optimizing inventory allocation by so-called rationing rules (Gayon et al., 2008), as well as complex systems where production, maintenance, and supply are integrated and the optimum is sought using dynamic programming.

In this paper, we applied DP to a production and inventory problem with time varying costs and capacity constraints. By formulating the state, decision, and transition equations of the model and solving it with backward induction, we derived an optimal production and inventory plan that minimises total variable costs while satisfying all demand and respecting production and storage limits. The resulting policy, together with the set of admissible alternative plans, demonstrates how DP can provide transparent, quantitatively grounded guidance for tactical production and inventory decisions.

With the advancement of information technology and better data, the level of detail and quality of dynamic programming models in the field of production inventory management will continue to increase, enabling more detailed optimization, robust operation even in uncertain environments, and truly data driven decision making at all levels of the modern supply chain.

This work was supported by research project A-25-103/3020-08 „Preskúmanie interdisciplinárnych stratégií pre efektívne manažovanie životného prostredia a udržateľný rozvoj. “

This work was supported by VEGA grant no. 1/0120/23, "Environmental models as a tool for ecological and economic decision-making."

Literature

1. Anderson, D. R., Sweeney, D. J., Williams, T. A. et al. (2010). *An Introduction to Management Science*, 13th Edition. Cengage Learning.
2. Bellman, R. B. (1972). *DYNAMIC PROGRAMMING (6th printing)*. Princeton, United States of America: Princeton University Press. Retrieved from https://gwern.net/doc/statistics/decision/1957-bellman-dynamicprogramming.pdf
3. Clark, A. J., & Scarf, H. (2004). *Optimal Policies for a Multi-Echelon Inventory Problem*. Management Science, 50(12), 1782–1790. http://www.jstor.org/stable/30046146
4. Gayon, J., Benjaafar, S., & De Véricourt, F. (2008). *Using Imperfect Advance Demand Information in Production-Inventory Systems with Multiple Customer Classes*.

Manufacturing & Service Operations Management, 11(1), 128–143. https://doi.org/10.1287/msom.1070.0201
5. Gallego, G., & Özer, Ö. (2001). *Integrating Replenishment Decisions with Advance Demand Information*. Management Science, 47(10), 1344–1360. https://doi.org/10.1287/mnsc.47.10.1344.10261
6. Goga, M. (2011). *Ekonomická dynamika*. Bratislava: Wolters Kluwer. ISBN 978-80-8078-394-5.
7. Hillier, F. S., & Lieberman, G. J. (2010). *Introduction to Operations research (ninth*). McGraw-Hill.
8. Karp, L., & Traeger, C. P. (2024). *Dynamic Environmental and Resource Economics*. Draft. University of California, Berkeley & University of Oslo. Dostupné na: https://traeger.eu/pdf/KarpTraegerDraft.pdf
9. Koopmans, T. C. (1949). *Optimum Utilization of the Transportation System*. Dostupné na: http://www.jstor.org/stable/1907301
10. Laščiak, A. et al. (1985). *Dynamické modely*. Bratislava: Alfa.
11. Laščiak, A. (1983). *Optimálne programovanie*. Česko: Alfa. ISBN 80-05-00663-2.
12. Madaras, T. (2005). *Rekurentné vzťahy*. Kombinatorika pre informatikov. Dostupné na: https://umv.science.upjs.sk/madaras/slide7i.pdf
13. Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Spojené kráľovstvo: Wiley.
14. Ramsey, F. (1937). *A Mathematical Theory of Saving*.
15. Selvanathan, R. (2015). A Dynamic Programming Approach to Sustainability. *Journal of Management and Sustainability*, 5(1), 1-10. http://dx.doi.org/10.5539/jms.v5n1p1
16. Topan, E., Tan, T., Van Houtum, G., & Dekker, R. (2018). Using imperfect advance demand information in lost-sales inventory systems with the option of returning inventory. IISE Transactions, 50(3), 246–264. https://doi.org/10.1080/24725854.2017.1403060