Project-Based Learning for Teaching the Knapsack Problem in the Course “Telecommunication Security”

Adriana Naydenova Borodzhieva
Department of Telecommunications
University of Ruse “Angel Kanchev”
Ruse, Bulgaria
aborodzhieva@uni-ruse.bg

2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH) | 978-1-6654-3778-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOTEH53737.2022.9751273

Abstract—**The paper presents the implementation of a project-based approach used in the course “Telecommunication Security” for the seventh-semester students-bachelors from the specialty “Internet and Mobile Communications” at the University of Ruse “Angel Kanchev” when teaching the knapsack problem. Working on projects, one of the trends of future education, Education 4.0, allows the development of basic 21st-century skills, such as literacy, numeracy, ICT literacy, creativity, initiative, critical thinking, problem-solving skills, etc.**

Keywords-Problem-based learning; educational process; 21st-century skills; knapsack problem; cryptosystems; MS Excel.

I. INTRODUCTION

In recent years there has been increasing talk of Industry 4.0 and Education 4.0. Education 4.0 is the answer to the global needs for advanced integration of people and technology. The use of computer and communication technologies by leading universities for acquiring educational resources to improve student engagement and learning can be the way to support the implementation of Education 4.0 [1]. The integration of ICT and technological tools in universities is a major challenge for the new era of the Education 4.0 systems.

Opportunities to encourage different thinking and creativity are limited in traditional lecture-oriented classrooms, which could discourage students from applying and developing their creativity and different thinking in an academic context [2]. Project-based learning and STEM-based learning are an alternative, including an innovative teaching method, to traditional learning that can be used to challenge students to identify creative solutions to problems. The main idea of project-based learning (PBL), one of the main trends of future education, known as Education 4.0 [1], is to focus on real-world problems for attracting students’ interest and developing their critical thinking and problem-solving skills [3], the basic 21st-century skills defined within the 2020 Strategic Framework for Education and Training, the recommendation of 21st-century skills guide [4, 5], that schools/universities in the member states of the European Union must develop in children and young people/students. Integrating STEM PBL into classrooms can help students develop a positive attitude toward engaging with different thinking and positive perceptions of their creative problem-solving skills, thus increasing their perseverance in engaging in complex problems that require the generation of original ideas [2].

Nowadays, educational institutions must face the challenge of 21st-century competencies, understanding them as a set of knowledge, skills, attitudes, and values enabling students to raise their place in society in constant change and how to organize their educational activities to develop these skills [5].

The paper presents the implementation of project-based learning used in the course “Telecommunication Security”, intended for the seventh-semester students from the specialty “Internet and Mobile Communications” at the University of Ruse “Angel Kanchev”, when teaching the knapsack problem, as well as the 21st-century skills developed in students during the training process.

II. PROJECT-BASED LEARNING FOR TEACHING THE KNAPSACK PROBLEM

A. *Project-Based Learning in the Course “Telecommunication Security”*

The course “Telecommunication Security” covers the main issues in the fields of cryptography and security. Cryptography is a technique for converting data from plaintext to encrypted text and vice versa. Security protects any information from theft or damage from hardware or software [3]. The course deals with issues of cryptographic protection of information, ranging from classical ciphers to asymmetric encryption systems, based on complex mathematical problems to solve.

Since the 2010-2011 academic year active teaching methods have been used in the course. For each topic studied students received an individual assignment, which had to be solved manually on a pre-prepared form for the exercise (published in the e-learning platform of the University of Ruse) and submitted until the end of class to the lecturer. During the classes, the students received guidelines for the implementation of each task and, if necessary, received additional help from the teacher and/or colleagues. The use of active teaching methods

The work presented in this paper is completed as partial fulfilment of Project 2022 – FEEA – 03, financed under the Scientific and Research Fund of the University of Ruse “Angel Kanchev”.

definitely motivated students to work and increased the level of mastering the material.

In the last academic year, 2020-2021, due to the Covid-19 pandemic, the teaching staff of the course "Telecommunication Security" decided to apply the approach of project-based learning. The lecturers give the key project concept to the students and the students find an appropriate solution to a problem by using different computer-based tools. After completion of the projects, students submit the implementation as documentation and the teachers evaluate them. The result is that students have built up enough confidence to apply not only what they have learned in class, but can also put these concepts into practice.

The students were acquainted with various works of the author, which are designed mainly to facilitate the lecturer in generating different options of the individual assignments for the independent work of students.

Some of these applications were developed using MATLAB ("a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models" [6]) and GUIDE (an option for creating applications with graphical user interfaces in MATLAB [7]) – for example, implementations of bifid ciphers in MATLAB [8] and affine ciphers with GUI in MATLAB [9], pictures of which are presented in Fig. 1.

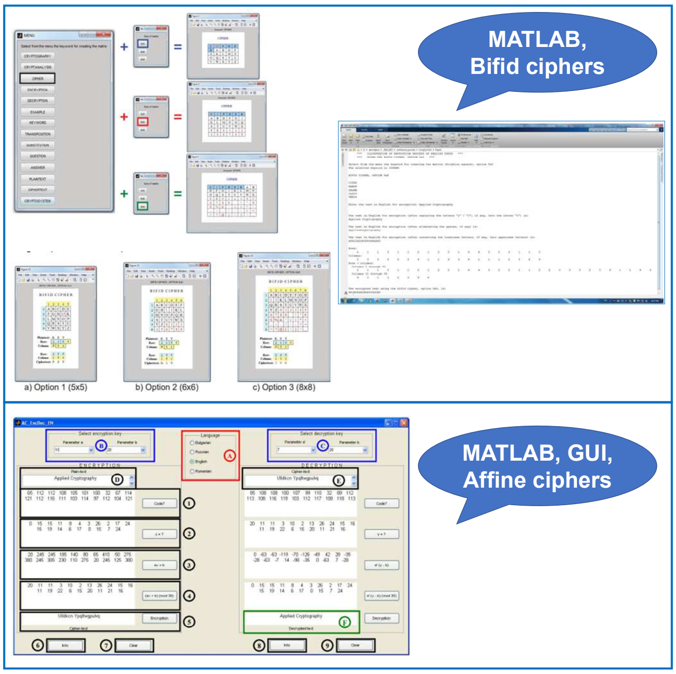

Figure 1. Applications developed in MATLAB for implementing some classical ciphers such as bifid ciphers, affine ciphers

Most of these applications were developed using MS Excel [10], which is extremely convenient due to the tabular presentation of the calculations that need to be performed when solving a specific task, for example on the topics: Hill ciphers (a polygraphic substitution cipher based on linear algebra [11]), RSA algorithm [12], Shamir's Secret Sharing algorithm [13], cryptosystems based on linear feedback shift registers [14] or non-linear feedback shift register, Fibonacci, Galois [15]. Pictures of these applications are presented in Fig. 2.

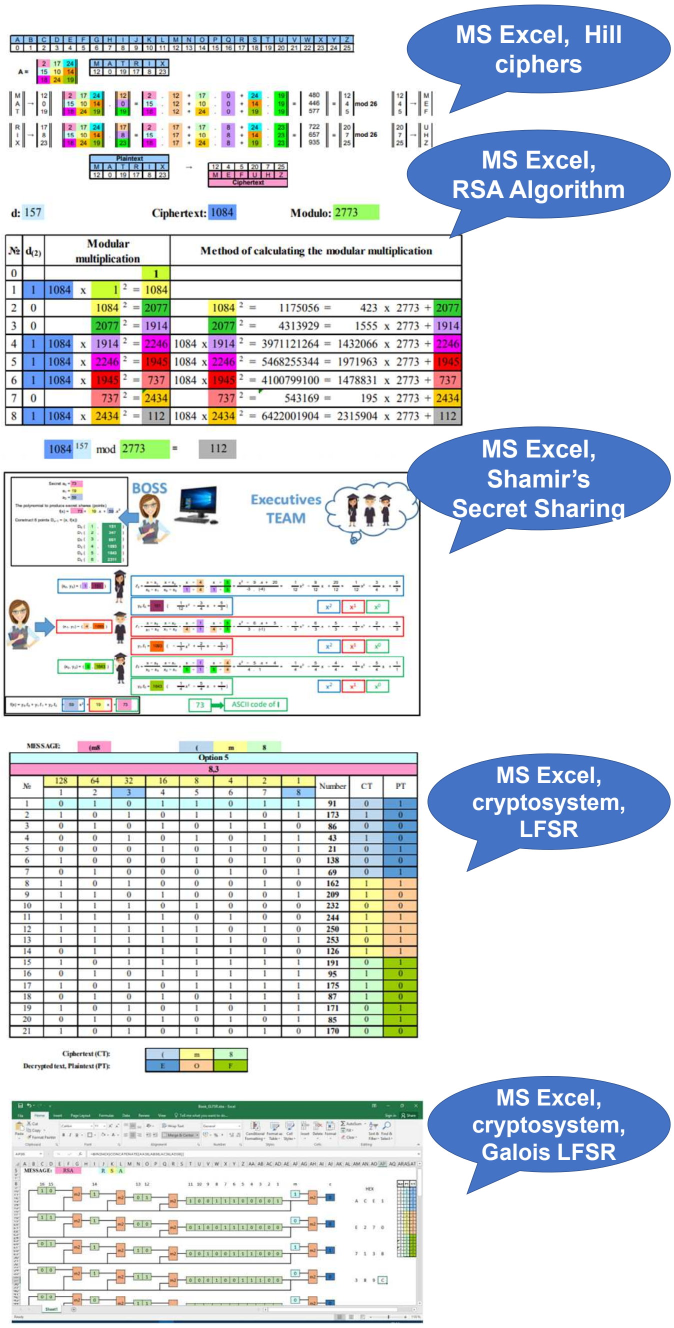

Figure 2. Applications developed in MS Excel for implementing some classical ciphers and cryptographic algorithms such as Hill ciphers, RSA, Shamir's Secret Sharing, stream cryptosystems

Many of these applications were developed using various computer-based tools for synthesis and analysis of cryptosystems, where in addition to MS Excel, software such as Logisim [16] (for analysis of various digital circuits, such as cryptosystems based on linear or non-linear feedback shift registers) and ISE Project Navigator [17]. These systems were also implemented on FPGA-based laboratory boards developed at our university and actively used in the teaching process, by drawing the logic diagrams in the ISE Project Navigator and subsequent programming of the board. Pictures of these applications are presented in Fig. 3 (for Logisim [15, 18]) and Fig. 4 (for programming the board [19, 20]).

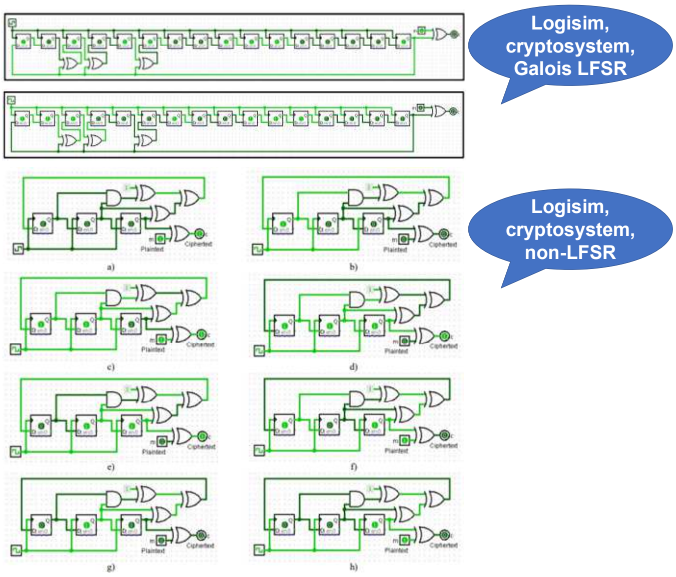

Figure 3. Applications developed in Logisim for implementing some cyptosystems with linear or non-linear feedback shift register

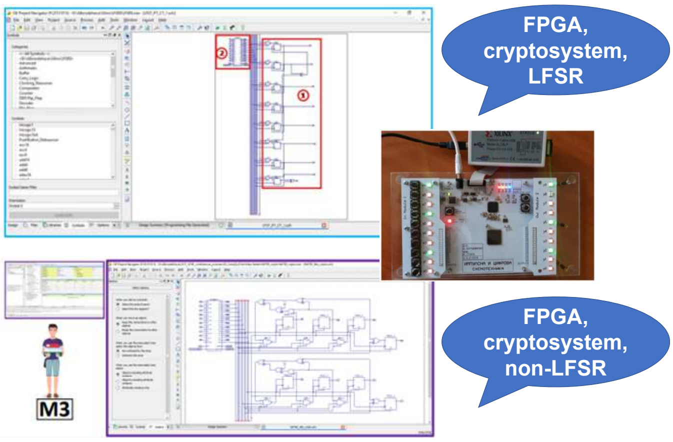

Figure 4. Applications developed in ISE Project Navigator for implementing some cyptosystems with linear or non-linear feedback shift register and for programming the FPGA-based laboratory board

The acquaintance of the students with previously developed by the teaching team works (Fig. 1…Fig. 4) strengthened their interest in the material studied and motivated them for the implementation of their assignment on project-based learning.

As a start to the project-based learning initiative introduced the last 2020-2021 academic year, it was decided by the teaching staff to entrust students with the implementation of the knapsack problem algorithm, as relatively easier, and due to its great application in various fields and cryptosystems. The students had to get acquainted in advance with the topic "Knapsack Problem" [21], presented briefly later (section *B*) as well as knapsack examples (section *C*).

B. The Knapsack Problem

The knapsack problem, a problem in combinatorial optimization, has been studied for more than 100 years, with early works dating back to 1897 (the paper "On the partition of numbers" of G. B. Mathews). The name "knapsack problem" dates back to the early works of the American mathematician Tobias Dantzig (1884–1956) (the paper "The Language of Science" in 1930) and refers to "the commonplace problem of packing the most valuable or useful items without overloading the luggage" [21]. It is formulated as follows: "Given a set of items, each with a weight and a value, determine the number of each item included in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible." [22] The knapsack problem "often arises in resource allocation where the decision makers have to choose from a set of non-divisible projects or tasks under a fixed budget or time constraint, respectively" [23].

Fig. 5 illustrates the classic knapsack problem. Here, the knapsack is filled with a subset of items shown with weights in grams. The problem is to determine which items are inserted in the knapsack given the weight of the filled knapsack. For this simple example (only 6 items in Fig. 5), the solution can easily be found by "trial and error" method. However, if there are 100 possible items in the set instead of 6, the problem may become computationally infeasible.

Figure 5. Knapsack problem

In the knapsack problem a knapsack vector and a data vector are defined. The knapsack vector (1) is an n-tuple of distinct integers (analogous to the set of possible knapsack items) and the data vector (2) is an n-tuple of binary symbols

$$\boldsymbol{a} = a_1, a_2, ..., a_n \tag{1}$$

$$\boldsymbol{x} = x_1, x_2, ..., x_n \tag{2}$$

The knapsack, S, is the sum (3) of a subset of the components of the knapsack vector and the problem can be stated as follows: Given S and knowing $\boldsymbol{a}$, determine $\boldsymbol{x}$.

$$S = \sum_{i=1}^{n} a_i x_i = \boldsymbol{ax}\,, \text{ where } x_i = 0, 1\,. \tag{3}$$

C. Knapsack Examples

1. Given $\boldsymbol{a} = 1,2,4,8,16,32$ and $S = \boldsymbol{a}.\boldsymbol{x} = 26$, find $\boldsymbol{x}$. In this example it might be seen that $\boldsymbol{x}$ is the binary representation of S considering that $\boldsymbol{a} = 2^0, 2^1, 2^2, 2^3, 2^4, 2^5$ and the problem is reduced to the decimal-to-binary conversion. The data vector $\boldsymbol{x}$ is easily found since $\boldsymbol{a}$ is super-increasing, which means that each component of the n-tuple $\boldsymbol{a}$ is larger than the sum of the preceding components, i.e.

$$a_i > \sum_{j=1}^{i-1} a_j \tag{4}$$

When $\boldsymbol{a}$ is super-increasing, the solution of $\boldsymbol{x}$ is found by starting with $x_n = 1$, if $S \geq a_n$ (otherwise $x_n = 0$) and continuing according to the relationship

$$x_i = \begin{cases} 1, \, if \;\; S - \sum_{j=i+1}^{n} x_j a_j \geq a_i \\ 0, \, otherwise \end{cases}, \text{ where } i = n-1, n-2, \ldots, 1. \tag{5}$$

Using (5) it is easy to compute $\boldsymbol{x} = 010110$ [21].

2. Given $\boldsymbol{a} = 171, 197, 459, 1191, 2410, 4517$ and $S = \boldsymbol{a}.\boldsymbol{x} = 3798$, find $\boldsymbol{x}$. Here, $\boldsymbol{a}$ is also super-increasing; therefore, $\boldsymbol{x}$ can be computed also using (5), which again yields $\boldsymbol{x} = 010110$ [21].

D. Knapsack Problem in the Educational Process

Before the lesson, students should consider the two examples (section *C*), detailing the solution of the problems based on (4) and (5). During the lesson, they solve two similar tasks manually (formulated below). After understanding the mathematical foundations, students must develop MS Excel-based applications (Fig. 6 and Fig. 7) to solve the two tasks.

Task 1: A knapsack vector $\boldsymbol{a} = [1, 2, 4, \ldots, 256, 512, 1024]$ is given. **1.** Prove that the knapsack vector is a super-increasing (fast-growing) vector. **2.** Determine the data vector $\boldsymbol{x}$, if the weight of the knapsack is known. Present the solution of the problem in tabular form.

Task 2: Peter, Ivan and Andrew go on an excursion on the occasion of the Student Holiday. Everyone takes things with them in their knapsack, and the weight of each item is known in advance: 171 g – sheets of paper and pencils; 197 g – compass; 459 g – first aid kit; 1191 g – clothes; 2410 g – drinks (beer, Coca Cola, natural juice, etc.); 4517 g – food products (meat, sandwiches, etc.). **1.** Prove that the knapsack vector is a super-increasing vector. **2.** Determine the contents of the knapsack of each of the three students, if its weight is known. Present the solution of the problem in tabular form. If the problem **does not have** an exact solution in some cases, determine the error in measuring the weight of the knapsack, for example "given the weight of the knapsack 5881 g, there is an error of 2 grams, the exact weight of the knapsack is 5879 g, and its contents are: ....".

Knapsack vector length: n = 11

Knapsack vector: $\underline{a}$

a_{11} =	1024
a_{10} =	512
a_9 =	256
a_8 =	128
a_7 =	64
a_6 =	32
a_5 =	16
a_4 =	8
a_3 =	4
a_2 =	2
a_1 =	1

Sum of the elements of the knapsack vector: 2047

Knapsack weight: S = 1955

Determining the data vector: $\underline{x}$

			S_{i+1}	$\geq a_i$	x_i	$x_i{*}a_i$	$S_{i+2} - x_{i+1}{*}a_{i+1}$
Iteration 1:	i = 11	$S_{12} \geq a_{11}$	1955	≥ 1024	1	1024	–
Iteration 2:	i = 10	$S_{11} \geq a_{10}$	931	≥ 512	1	512	931
Iteration 3:	i = 9	$S_{10} \geq a_9$	419	≥ 256	1	256	419
Iteration 4:	i = 8	$S_9 \geq a_8$	163	≥ 128	1	128	163
Iteration 5:	i = 7	$S_8 \geq a_7$	35	≥ 64	0	0	35
Iteration 6:	i = 6	$S_7 \geq a_6$	35	≥ 32	1	32	35
Iteration 7:	i = 5	$S_6 \geq a_5$	3	≥ 16	0	0	3
Iteration 8:	i = 4	$S_5 \geq a_4$	3	≥ 8	0	0	3
Iteration 9:	i = 3	$S_4 \geq a_3$	3	≥ 4	0	0	3
Iteration 10:	i = 2	$S_3 \geq a_2$	3	≥ 2	1	2	3
Iteration 11:	i = 1	$S_2 \geq a_1$	1	≥ 1	1	1	1

Determining the data vector: $\underline{x}$

$\underline{x}$ = [1;1;0;0;0;1;0;1;1;1;1]

At the given weight of the knapsack the problem:
has an exact solution

a)

Knapsack vector length: n = 11

Knapsack vector: $\underline{a}$

a_{11} =	1024
a_{10} =	512
a_9 =	256
a_8 =	128
a_7 =	64
a_6 =	32
a_5 =	16
a_4 =	8
a_3 =	4
a_2 =	2
a_1 =	1

Sum of the elements of the knapsack vector: 2047

Knapsack weight: S = 2105

Determining the data vector: $\underline{x}$

			S_{i+1}	$\geq a_i$	x_i	$x_i{*}a_i$	$S_{i+2} - x_{i+1}{*}a_{i+1}$
Iteration 1:	i = 11	$S_{12} \geq a_{11}$	2105	≥ 1024	1	1024	–
Iteration 2:	i = 10	$S_{11} \geq a_{10}$	1081	≥ 512	1	512	1081
Iteration 3:	i = 9	$S_{10} \geq a_9$	569	≥ 256	1	256	569
Iteration 4:	i = 8	$S_9 \geq a_8$	313	≥ 128	1	128	313
Iteration 5:	i = 7	$S_8 \geq a_7$	185	≥ 64	1	64	185
Iteration 6:	i = 6	$S_7 \geq a_6$	121	≥ 32	1	32	121
Iteration 7:	i = 5	$S_6 \geq a_5$	89	≥ 16	1	16	89
Iteration 8:	i = 4	$S_5 \geq a_4$	73	≥ 8	1	8	73
Iteration 9:	i = 3	$S_4 \geq a_3$	65	≥ 4	1	4	65
Iteration 10:	i = 2	$S_3 \geq a_2$	61	≥ 2	1	2	61
Iteration 11:	i = 1	$S_2 \geq a_1$	59	≥ 1	1	1	59

Determining the data vector: $\underline{x}$

$\underline{x}$ = [1;1;1;1;1;1;1;1;1;1;1]

At the given weight of the knapsack the problem:
has no exact solution

b)

Figure 6. MS Excel-based application for solving the knapsack problem, Task 1

Knapsack vector length: n = 6

Knapsack vector: $\underline{a}$

a_6 =	4517
a_5 =	2410
a_4 =	1191
a_3 =	459
a_2 =	197
a_1 =	171

Sum of the elements of the knapsack vector: 8945

Knapsack weight: S = 3066

Determining the data vector: $\underline{x}$

			S_{i+1}	$\geq a_i$	x_i	$x_i * a_i$	$S_{i+2} - x_{i+1} * a_{i+1}$
Iteration 1:	i = 6	$S_7 \geq a_6$	3066	≥ 4517	0	0	–
Iteration 2:	i = 5	$S_6 \geq a_5$	3066	≥ 2410	1	2410	3066
Iteration 3:	i = 4	$S_5 \geq a_4$	656	≥ 1191	0	0	656
Iteration 4:	i = 3	$S_4 \geq a_3$	656	≥ 459	1	459	656
Iteration 5:	i = 2	$S_3 \geq a_2$	197	≥ 197	1	197	197
Iteration 6:	i = 1	$S_2 \geq a_1$	0	≥ 171	0	0	0

Determining the data vector: $\underline{x}$

$\underline{x}$ = [0;1;1;0;1;0]

At the given weight of the knapsack the problem:
has an exact solution

a)

Knapsack vector length: n = 6

Knapsack vector: $\underline{a}$

a_6 =	4517
a_5 =	2410
a_4 =	1191
a_3 =	459
a_2 =	197
a_1 =	171

Sum of the elements of the knapsack vector: 8945

Knapsack weight: S = 3057

Determining the data vector: $\underline{x}$

			S_{i+1}	$\geq a_i$	x_i	$x_i * a_i$	$S_{i+2} - x_{i+1} * a_{i+1}$
Iteration 1:	i = 6	$S_7 \geq a_6$	3057	≥ 4517	0	0	–
Iteration 2:	i = 5	$S_6 \geq a_5$	3057	≥ 2410	1	2410	3057
Iteration 3:	i = 4	$S_5 \geq a_4$	647	≥ 1191	0	0	647
Iteration 4:	i = 3	$S_4 \geq a_3$	647	≥ 459	1	459	647
Iteration 5:	i = 2	$S_3 \geq a_2$	188	≥ 197	0	0	188
Iteration 6:	i = 1	$S_2 \geq a_1$	188	≥ 171	1	171	188

Determining the data vector: $\underline{x}$

$\underline{x}$ = [1;0;1;0;1;0]

At the given weight of the knapsack the problem:
has no exact solution

b)

Figure 7. MS Excel-based application for solving the knapsack problem, Task 2

Fig. 6 a and Fig. 7 a present the cases in which **the problem has an exact solution**, and Fig. 6 b and Fig. 7 b – the cases in which **the problem has no exact solution** (a message is displayed in the bottom of the worksheet in MS Excel in the corresponding color using Conditional Formatting).

It should be noted that when using a super-increasing knapsack vector, which contains powers of 2 (Fig. 6), the problem will always have a solution if the weight of the knapsack is less than or equal to the sum of the elements of the knapsack vector (Fig. 6 a) and will not have a solution if the weight of the knapsack is greater than the sum of the elements of the knapsack vector (Fig. 6 b). This problem corresponds to the conversion from decimal to binary number system which explains the explanation above.

When using a super-increasing knapsack vector, which does not contain powers of 2 (Fig. 7), the problem may not have an exact solution, even when the weight of the knapsack is less than the sum of the elements of the knapsack vector (visualized in Fig. 7 b).

The detailed writing of the solution of the problem by the students using (4) and (5) is illustrated in Fig. 8 (Example 2).

$$\boldsymbol{a} = \underbrace{171}_{a_1}, \underbrace{197}_{a_2}, \underbrace{459}_{a_3}, \underbrace{1191}_{a_4}, \underbrace{2410}_{a_5}, \underbrace{4517}_{a_6}$$

$$a_i > \sum_{j=1}^{i-1} a_j, \; i = 2,3,\ldots,n$$

a – super-increasing vector

$$i = 2 \Rightarrow \; a_2 > \sum_{j=1}^{2-1} a_j = a_1; 197 > 171 - OK$$

$$i = 3 \Rightarrow \; a_3 > \sum_{j=1}^{3-1} a_j = a_1 + a_2 = 171 + 197 = 368; 459 > 368 - OK$$

$$i = 4 \Rightarrow \; a_4 > \sum_{j=1}^{4-1} a_j = a_1 + a_2 + a_3 = 368 + 459 = 827; 1191 > 827 - OK$$

$$i = 5 \Rightarrow \; a_5 > \sum_{j=1}^{5-1} a_j = a_1 + a_2 + a_3 + a_4 = 827 + 1191 = 2018; 2410 > 2018 - OK$$

$$i = 6 \Rightarrow \; a_6 > \sum_{j=1}^{6-1} a_j = a_1 + a_2 + a_3 + a_4 + a_5 = 2018 + 2410 = 4428; 4517 > 4428 - OK$$

$$n = 6 \qquad x_6 = 1, \text{ if } S \geq a_6, i.e. 3798 \geq 4517 - NO \Rightarrow x_6 = 0 \quad S_7 = S$$

$$i = 6 - 1 = 5 \Rightarrow x_5 = \begin{cases} 1, \; if \; S - \sum_{j=5+1}^{6} x_j a_j \geq a_5 \\ 0, \; otherwise \end{cases}$$

$$S_6 = S - \sum_{j=5+1}^{6} x_j a_j = \underbrace{S - x_6 a_6}_{S_6} = 3798 - 0.4517 = 3798 \geq a_5 = 2410, i.e\; 3798 \geq 2410 - OK \Rightarrow x_5 = 1$$

$$i = 6 - 2 = 4 \Rightarrow x_4 = \begin{cases} 1, \; if \; S - \sum_{j=4+1}^{6} x_j a_j \geq a_4 \\ 0, \; otherwise \end{cases}$$

$$S_5 = S - \sum_{j=4+1}^{6} x_j a_j = \underbrace{S - (x_5 a_5 + x_6 a_6)}_{S_5} = S_6 - x_5 a_5 = 3798 - 1.2410 = 1388 \geq a_4 = 1191, i.e. 1388 \geq 1191 - OK \Rightarrow x_4 = 1$$

$$i = 6 - 3 = 3 \Rightarrow x_3 = \begin{cases} 1, \; if \; S - \sum_{j=3+1}^{6} x_j a_j \geq a_3 \\ 0, \; otherwise \end{cases}$$

$$S_4 = S - \sum_{j=3+1}^{6} x_j a_j = \underbrace{S - (x_4 a_4 + x_5 a_5 + x_6 a_6)}_{S_4} = S_5 - x_4 a_4 = 1388 - 1.1191 = 197 \geq a_3 = 459, i.e. 197 \geq 459 - NO \Rightarrow x_3 = 0$$

$$i = 6 - 4 = 2 \Rightarrow x_2 = \begin{cases} 1, \; if \; S - \sum_{j=2+1}^{6} x_j a_j \geq a_2 \\ 0, \; otherwise \end{cases}$$

$$S_3 = S - \sum_{j=2+1}^{6} x_j a_j = \underbrace{S - (x_3 a_3 + x_4 a_4 + x_5 a_5 + x_6 a_6)}_{S_3} = S_4 - x_3 a_3 = 197 - 0.459 = 197 \geq a_2 = 197, i.e. 197 \geq 197 - OK \Rightarrow x_2 = 1$$

$$i = 6 - 5 = 1 \Rightarrow x_1 = \begin{cases} 1, \; if \; S - \sum_{j=1+1}^{6} x_j a_j \geq a_1 \\ 0, \; otherwise \end{cases}$$

$$S_2 = S - \sum_{j=1+1}^{6} x_j a_j = \underbrace{S - (x_2 a_2 + x_3 a_3 + x_4 a_4 + x_5 a_5 + x_6 a_6)}_{S_2} = S_3 - x_2 a_2 = 197 - 1.197 = 0 \geq a_1 = 171, i.e. 0 \geq 171 - NO \Rightarrow x_1 = 0$$

Finding x

$$\boldsymbol{x} = \underbrace{0}_{x_1}, \underbrace{1}_{x_2}, \underbrace{0}_{x_3}, \underbrace{1}_{x_4}, \underbrace{1}_{x_5}, \underbrace{0}_{x_6}$$

Figure 8. Detailed solution of the knapsack problem done by the students for formulation of the algorithm, implemented in MS Excel-based application

This detailed solution (Fig. 8) helps students to understand the mathematical apparatus needed to solve the problem and to formulate the algorithm (6) for building the application.

$$S_{n+1} = S;\; x_i = \begin{cases} 1, \; if \;\; S_{i+2} = S_{i+1} - x_{i+1}.a_{i+1} \geq a_i \\ 0, \; otherwise \end{cases}, \qquad i = (n-1) \div 1. \quad (6)$$

It can be seen (Fig. 6, Fig. 7) that during the development of the application corresponding columns for S_{i+1}, $x_{i+1}.a_{i+1}$, $S_{i+2} - x_{i+1}.a_{i+1}$ are provided for the calculation of the elements in (6), which easily traces the algorithm for the "step-by-step" solution of the knapsack problem and the determination of the data vector **x**. Fig. 9 illustrates the formulas embedded when implementing the application.

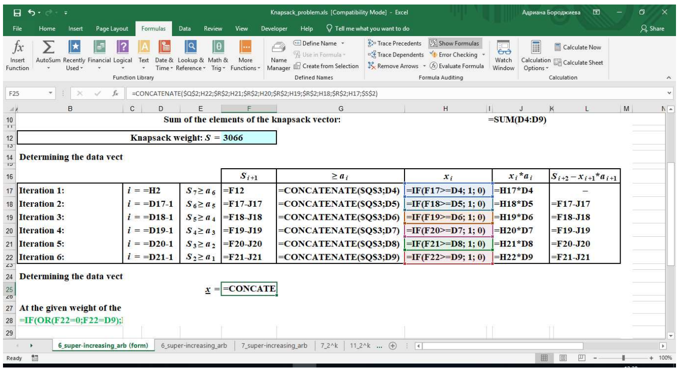

Figure 9. Formulas embedded when implementing the application in Excel

III. CONCLUSIONS

The paper presents the application of project-based learning in the course "Telecommunication Security" at the University of Ruse "Angel Kanchev" when teaching the knapsack problem. Knapsack problems are used in real-world decision-making processes in many fields, such as generating keys for the Merkle-Hellman and other cryptosystems, finding the least wasteful way to cut raw materials, selection of investments and portfolios, or assets for asset-backed securitization.

REFERENCES

[1] S. Ghavifekr, S.Y. Wong, "Technology leadership in malaysian schools: the way forward to education 4.0 – ICT utilization and digital transformation", International Journal of Asian Business and Information Management, Open Access, Volume 13, Issue 2, 2022, doi 10.4018/IJABIM.20220701.oa3.

[2] A. Bicer, Y. Lee, R.M. Capraro, M.M. Capraro, L.R. Barroso, M. Rugh, "Examining the effects of STEM PBL on students' divergent thinking attitudes related to creative problem solving", Proceedings – Frontiers in Education Conference, FIE, 2019, doi 10.1109/FIE43999.2019.9028431.

[3] B. Panapt, C. Pandit, "Project-based learning approach in undergraduate engineering course of cryptography and security in computer science", Journal of Engineering Education Transformations, Volume 33, Special Issue 1, pp. 153 – 158, 2019, doi 10.16920/jeet/2019/v33i1/149006.

[4] A. Milheiro Silva, L. García-Docampo, S. Marques Silva, M. del Mar Lorenzo-Moledo. "Challenges of the border educational centers of Portugal and Spain towards the «21st century skills»", Teoria de la Educacion, Vol. 34 (1), pp. 167 – 187, 2022, doi 10.14201/TERI.25682.

[5] N. Chaiyama, N. Kaewpila, "The development of life and career skills in 21st century test for undergraduate students", European Journal of Education Research, Vol. 11 (1), pp. 51 – 68, 2022, doi 10.12973/eu-jer.11.1.51.

[6] MATLAB, oficial site, https://www.mathworks.com (January 2022)

[7] Create apps with graphical user interfaces in MATLAB, https://www.mathworks.com/discovery/matlab-gui.html (January 2022).

[8] A. Borodzhieva, "MATLAB-based software tool for implementation of bifid ciphers". 18-th International Conference on Computer Systems and Technologies – CompSysTech'17, Ruse, 2017, Proceedings, pp. 326 – 333, ISBN: 978-1-4503-5234-5.

[9] A. Borodzhieva, P. Manoilov, "Training module with graphical user interface for encryption and decryption using affine ciphers applied in cryptosystems". 2014 IEEE 20th International Symposium for Design and Technology in Electronic Packaging, SIITME 2014, Bucharest, Romania, Proceedings, pp. 281 – 286, ISBN: 978-1-4799-6961-6.

[10] MS Excel, oficial site, https://www.microsoft.com/bg-bg/microsoft-365/excel (January 2022)

[11] A. Borodzhieva, "MS Excel-based application for encryption and decryption of english texts with the Hill cipher on the basis of 3×3-matrix". XXV International Scientific Conference Electronics – ET2016, Sozopol, Bulgaria, pp. 67-71, ISBN: 978-1-5090-2881-8.

[12] A. Borodzhieva, "Software implementation of a module for encryption and decryption using the RSA algorithm". XXV International Scientific Conference Electronics – ET2016, 2016, Sozopol, Bulgaria, pp. 63-66, ISBN: 978-1-5090-2881-8.

[13] A. Borodzhieva, "MS Excel-based application for implementing the cryptographic algorithm Shamir's secret sharing". 43rd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2020, Opatija, Croatia, Engineering Education, Proceedings, pp. 1611-1616, ISBN: 978-953233099-1, doi: 10.23919/MIPRO48935.2020.9245422.

[14] A. Borodzhieva, "Modeling of cryptosystems based on linear feedback shift registers using spreadsheets". 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019, Opatija, Croatia, Engineering Education, pp. 1713 – 1718, ISBN: 1847-3946.

[15] A. Borodzhieva, "Computer-based education for teaching the topic "Galois linear feedback shift registers". 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging, SIITME 2020, Pitești, Romania, Proceedings, pp. 291-294, ISBN: 978-1-5386-5577-1, doi: 10.1109/SIITME50350.2020.9292268.

[16] Logisim, www.cburch.com/logisim/ (January 2022).

[17] Xilinx documentation, https://www.xilinx.com/content/xilinx/en/downloadNav/design-tools/v2012_4---14_7.htm (January 2022).

[18] A. Borodzhieva, "Computer-based tools applied in the course "Telecommunication Security"". 16th International Scientific Conference "eLearning and Software for Education" eLSE 2020, Bucharest, Romania, Vol. 2, pp. 42 – 52, doi: 10.12753/2066-026X-20-091.

[19] Borodzhieva, A., I. Stoev, V. Mutkov. FPGA implementation of cryptosystems based on linear feedback shift registers for educational purposes. 29th Annual Conference of the European Association for Education in Electrical and Information Engineering, 2019, Ruse, Bulgaria, Proceedings, pp. 306 – 309, ISBN: 978-1-7281-3221-1.

[20] A. Borodzhieva, I. Tsvetkova, S. Zaharieva, D. Dimitrov and V. Mutkov. "Active and interactive methods used in teaching and learning the topic "Cryptosystems Based on Nonlinear Feedback Shift Registers"". 2021 IEEE 27th International Symposium for Design and Technology in Electronic Packaging, SIITME 2021, Timisoara, Romania, pp. 137-142, doi: 10.1109/SIITME53254.2021.9663418.

[21] B. Sklar, "Digital Communications. Fundamentals and Applications." Second Edition. Communications Engineering Services, Tarzana, California, 2001.

[22] Y. Fairstein, A. Kulik, J.S. Naor, D. Raz, H. Shachnai, "An almost optimal approximation algorithm for monotone submodular multiple knapsack", Journal of Computer and System Sciences, Volume 125, pp, 149 – 165, 2022, DOI 10.1016/j.jcss.2021.11.005.

[23] K. Fleszar, "A branch-and-bound algorithm for the quadratic multiple knapsack problem", European Journal of Operational Research, Volume 298, Issue 1, pp. 89 – 98, 2022, DOI 10.1016/j.ejor.2021.06.018.