Government of Russian Federation

Federal State Autonomous Educational Institution of Higher Education

“National Research University Higher School of Economics”

Syllabus for the course
“Discrete optimization”

for postgraduate program in 09.06.01 Computer Science and Computer Engineering / 05.13.01 “Systems Analysis, Control Theory, and Information Processing”, 05.13.11 “Mathematical Theory and Software for Computing Machinery, Systems, and Networks”, 05.13.17 “Theoretical Foundations of Computer Science”, 05.13.18 “Mathematical Modeling, Numerical Methods, and Software Systems”

Author:
Anton Osokin, PhD, associate professor, aosokin@hse.ru

Moscow - 2017

1. Scope of Use

This program establishes the minimal requirements to postgraduate students’ knowledge and skills for 09.06.01 Computer Science and Computer Engineering / 05.13.01 “Systems Analysis, Control Theory, and Information Processing”, 05.13.11 “Mathematical Theory and Software for Computing Machinery, Systems, and Networks”, 05.13.17 “Theoretical Foundations of Computer Science”, “05.13.18 Mathematical Modeling, Numerical Methods, and Software Systems” and determines the content of the course and educational techniques used in teaching the course.

The present syllabus is aimed at faculty teaching the course and postgraduate students studying 09.06.01 Computer Science and Computer Engineering / 05.13.01 “Systems Analysis, Control Theory, and Information Processing”, 05.13.11 “Mathematical Theory and Software for Computing Machinery, Systems, and Networks”, 05.13.17 “Theoretical Foundations of Computer Science”, 05.13.18 “Mathematical Modeling, Numerical Methods, and Software Systems”.

This syllabus meets the standards required by:

- Educational standards of National Research University Higher School of Economics;
- Postgraduate educational program for 09.06.01 Computer Science and Computer Engineering.
- University curriculum of the postgraduate program for 09.06.01 Computer Science and Computer Engineering / 05.13.01 “Systems Analysis, Control Theory, and Information Processing”, 05.13.11 “Mathematical Theory and Software for Computing Machinery, Systems, and Networks”, 05.13.17 “Theoretical Foundations of Computer Science”, 05.13.18 Mathematical Modeling, Numerical Methods, and Software Systems”, approved in 2014.

2. Learning Objectives

The learning objective of the course “Discrete optimization” is to give students basic and advanced tools for solving discrete (combinatorial) optimization problems. The course focuses on practical techniques for approaching provably hard tasks (such as graph coloring, traveling salesman problem, etc.).

3. Main Competencies Developed after Completing the Study of This Discipline

After completing the study of the discipline, the PhD student should have:

- Knowledge about different approaches for tackling discrete optimization problems and their successful use cases.
- Hands-on experience with easy and hard discrete optimization problems.
- Knowledge about how to design and develop discrete optimization methods for new problems and how to use existing tools.

After completing the study of the discipline, the student should have developed the following competencies:

Competence	Code	Descriptors (indicators of achievement of the result)	Educative forms and methods aimed at generation and development of the competence
the ability to carry out theoretical and experimental research in the field of professional activity	ОПК-1	PhD students obtain necessary knowledge in the field of discrete optimization	Lectures, Assignments, additional reading provided
the ability to develop new re-	ОПК-2	The PhD student is able to	Examples covered during

search methods and apply them in research in one’s professional field		choose an appropriate method for real-life problems and to calibrate the hyperparameters.	the lectures and tutorials. Assignments.
the ability to objectively evaluate the outcomes of research and development carried out by other specialists in other scientific institutions	ОПК-4	The PhD student is able to carry out comparative testing of competing models or methods.	Examples covered during the lectures and tutorials. Assignments.
the ability to do research in transformation of information into data and knowledge, models of data and knowledge representation, methods for knowledge processing, machine learning and knowledge discovery methods, principles of building and operating software for automation of these processes	ПК-4	The PhD student is able to recognize and analyze discrete optimization problems, choose appropriate methods, implement them in a programming language, and select the method that suits best for the tasks at hand.	Lectures, tutorials, and assignments.

4. Place of the Discipline in the Postgraduate Program Structure

This is an elective course for 05.13.01 “Systems Analysis, Control Theory, and Information Processing”, 05.13.11 “Mathematical Theory and Software for Computing Machinery, Systems, and Networks”, 05.13.17 “Theoretical Foundations of Computer Science”, 05.13.18 “Mathematical Modeling, Numerical Methods, and Software Systems”.

Postgraduate students are expected to be already familiar with basic data structured and algorithms, and have skills in analysis, linear algebra, optimization, computational complexity, and programming.

The following knowledge and competences are needed to study the discipline:

- A good command of the English language, both oral and written.
- A sound knowledge of algorithms and data structures, computational complexity, programming and system design principles.

5. Schedule for one 1 module

№	Topic	Total hours	Contact hours			Self-study
			Lectures	Seminars	Practice lessons	
1.	Introduction to discrete optimization	2	2	0		0
2.	Basic methods: dynamic programming, exhaustive search, branch-and-bounds	50	6	6		38
3.	Approximate methods: local search, mixed-integer programming, constraint programming	50	6	6		38

6.	Advanced topics: submodular optimization, Discrete optimization meets deep learning	50	6	6		38
	Total	152	20	18		114

6. Requirements and Grading

Homework assignments	2	Programming assignments consists in solving some discrete optimization problems
Exam	1	Combined written-oral exam. Consists in preparing answers to theoretical or practical questions and defence of the answers.

7. Assessment

Final assessments are based on the results of the programming assignments and the final exam. Students have to demonstrate knowledge of the material covered during the entire course.

8. The grade formula

The exam is worth 30% of the final mark.

Final course mark *is obtained from the following formula*: Final=0.7*(homework assignments)+ 0.3*(exam).

All grades having a fractional part greater or equal than 0.5 are rounded up.

Table of Grade Accordance

Ten-point grading Scale	**Five-point grading Scale**	
1 - very bad 2 – bad 3 – no pass	Unsatisfactory - 2	**FAIL**
4 – pass 5 – highly pass	Satisfactory – 3	**PASS**
6 – good 7 – very good	Good – 4	
8 – almost excellent 9 – excellent 10 – perfect	Excellent – 5	

9. Course description.

Topic 1. Introduction to discrete optimization

Examples of problems, computational complexity, some basic algorithms.

Topic 2. Dynamic programming

Types of dynamic programming, examples: Fibonacci numbers, shortest paths, text justification, problems of subsequences, etc.

Topic 3. Branch-and-bound

Combinatorial search, strategies to organize computations, Knapsack problem, examples of effective bounds.

Topic 4. Local search
Local search: advantages and disadvantages, examples of simple moves, local search for the traveling salesman problem (2-OPT, K-OPT), local search for scheduling problems.

Topic 5. Mixed-integer programming
Notion of linear relaxation, algorithm to solve linear programs, notion of mixed-integer programming, existing solvers; examples: graph coloring, traveling salesman problem.

Topic 6. Constraint programming
Notion of constraint programming: when useful, examples: 8 queens, Sudoku, etc.

Topic 7. Submodular optimization
Notion of submodularity, exact algorithms, exact methods: connection to min-cut/max-flow.

Topic 8. Discrete optimization meets deep learning.
Discrete optimization for deep learning (deep structured prediction), deep learning for discrete optimization (using deep learning to learn algorithms).

10. Educational technologies

The following educational technologies are used in the study process:

- discussion and analysis of the results during the tutorials;
- regular assignments to test the progress of the PhD student;

11. Final exam questions

The final exam will consist of a selection of problems. No electronic devices are allowed for the exam. Each question will focus on a particular topic presented during the lectures and seminars.

The questions consist in exercises on any topic seen during the lectures. To be prepared for the final exam, PhD students must be able to answer questions from the topics covered during the lecture and seminars.

12. Reading and Materials

Extra reading literature

Algorithms
Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
Algorithms by Robert Sedgewick and Kevin Wayne (free online)
Algorithms by Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani

Local Search
Local Search in Combinatorial Optimization edited by Emile Aarts, Jan Karel Lenstra
Handbook of Metaheuristics (International Series in Operations Research & Management Science) edited by Michel Gendreau, Jean-Yves Potvin
Constraint-Based Local Search by Pascal van Van Hentenryck, Laurent Michel

Linear and Integer Programming
Integer Programming by Laurence A. Wolsey
Integer and Combinatorial Optimization by Laurence A. Wolsey, George L. Nemhauser
Large Scale Linear and Integer Optimization: A Unified Approach by Richard Kipp Martin
Introduction to Linear Optimization by Dimitris Bertsimas, John N. Tsitsiklis
Understanding and Using Linear Programming by Jiri Matousek, Bernd Gärtner
Theory of Linear and Integer Programming by Alexander Schrijver

Constraint Programming
Programming with Constraints: An Introduction by Kimbal Marriott, Peter Stuckey
Handbook of Constraint Programming edited by Francesca Rossi, Peter van Beek, Toby Walsh
The OPL Optimization Programming Language by Pascal Van Hentenryck

13. Equipment.

The course requires a computer room, laptop and a projector.