Project- Applications in Optimization Prof. T. Lahmer 11. Juli 2025 The project is for students who are registered in the Module ”Optimization in Applications”. This Module is counted as 3 credits and its evaluation is separate from the Module ”Introduction to Optimization”. The project contains 2 main parts: • Tasks (1-3): applications of what you have learned from the classes of optimization and • Project which can be one of the following:– Experience Optimization Vividly with Self-built Robots Project. Only two groups of three students each will work on this project (due to the limited number of robots and the limited time). The project will be supervised by instructors and the time will be scheduled in advance. Please drop us an email that includes the students' names and emails (in cc). However, the first two groups who requested this project will be selected.– Topology Optimization Design. The project can be done individually or in small groups of 2-3 students. Organization of the submission: should be made as follows: • For the students, who will work on ”The Experience Optimization Vividly with Self-built Robots Project”, Tasks 2 and 3 should be done for each matriculation number. Task 1 and the project must be performed by the group. • The students, who will work on ”The topology optimization project”, can choose to do the work individually or in small groups of 2-3 students.– Students who choose to work individually: perfom Tasks 1-3 and the topology optimization project.– Students who choose to work in groups: Tasks 2 and 3 are completed for each matriculation number. Task 1 and the project must be performed by the group. Remark for the students who choose to work in groups: In Tasks 2-3, each student will be graded from his/her own codes, plots, results, and interpretations. 1 Content of the submission: • ONE PDF: The PDF should include in the first page the name of the student(s) and their matriculation numbers. The report contains all the results and interpretations of the results in your own words. The Matlab codes, which were used with the initial guess for each student, should also be included in the report. Make sure it includes well-written explanatory texts and properly formatted graphics/figures, and make sure you follow the rules of good scientific working. • ONE Archive fine (zip, rar) which contains:– solely Python-files for the students who choose to work on ”the Experience Optimization Vividly with Self-built Robots Project”.– solely the.stl file of the final optimized object for the students who choose to work on ”the topology optimization project”. Deadline: 30 September on Moodle Optimization in Applications via submission function. 1 Task 1: Linear Optimization Problem (5 points) The following minimization problem is given: min x1,x2 Z = 100x1 +125x2, s.t x1 ≤ 5, (c1) x2 ≤ 4, (c2) 3x1 +6x2 ≤ 30, (c3) 8x1 +4x2 ≤ 44, (c4) x1 +x2 ≥4, (c5) x1,x2 ≥ 0 (c6) 1. Graph the set of admissible solutions and solve the problem graphically. 2. We add another constraint (c7) as follows: x1 + x2 ≥ 7. Does the optimal solution change? Is this constraint an active constraint? Justify your response. 2 Task 2: Non-linear Optimization Problem (15 points) The following optimization problem is given: min x,y f(x,y) = x2(4−2.1x2 + 1 3x4)+xy +y2(−4+4y2) Use the initial guess [x0,y0]T. For x0 you take the second last digit of your matriculation number, for y0 you take the last digit of it. 2 1. Plot in 3D the function using the meshgrid function in the intervals −2 ≤ x ≤ 2 and −1 ≤y ≤1. Use the 3D plot to obtain graphically the local and global minimum points (please write separately the approximate coordinates of the local and global points). Here, please use [X,Y]=meshgrid(linspace(-2,2),linspace(-1,1)) for visualization. 2. Use the built-in Matlab function ”fminsearch”to get an overview of the minimum points. Print the results you get from Matlab. 3. Now, we want to calculate the optimal solutions using different methods. In the following questi ons, print the Matlab output of the final optimal solutions and the number of iterations. Visualize then the iteration points in the contour plot and the convergence function for each case. Please use [X,Y]=meshgrid(linspace(-30,30),linspace(-30,30)) for the visualization. (a) Evaluate the gradient of the cost function and use the steepest descent with the simplified Armijo-Rule method to find the optimal points. (b) Use now the conjugate gradient method with any approach you choose, e.g. the Polak Ribi`ere approach, to find the optimal points. (c) Use the Nelder-Mead method, which we implemented in class (not the built-in Matlab function ”fminsearch”) to find the optimal points. (d) Use the Powell method to find the optimal points. 4. Compare the results you obtained previously (from questions (1), (2) and (3)) and write your own observations to deduct the final minimum points. 3 Task 3: Constrained optimization problem (10 points) Given the following constrained optimization problem: min x,y f(x,y) = κ·(x2 −y)2 +(x−1)2, s.t (x−1)3 −y+1 ≤0, (c1) x+y ≤2. (c2) Please use for κ the place of your first letter of the family name in the alphabet times four, e.g., Lahmer = L, so κ = 12×4=48. The optimization shall begin with the initial guess (x0,y0)T (for each student in groups). For x0 take the second last digit of your matriculation number, for y0 you take the last digit. 1. Solve the problem using the penalty method. Plot the contour lines, the constraints, and the optimal solution. (Hint: make a table of comparison. For the contour lines, please use in Matlab the following function: [X1,X2]=meshgrid(linspace(-5,5),linspace(-1,25))) 2. While varying the penalty wighting factor, does the results change? Justify your response. 3. Solve the problem by hand using the KKT method (you can provide a handwritten solution for this question). 3 4 Topology optimization Design (30 points) Design an object virtually using topology optimization. The project can be conducted individually or in groups or 2-3 students. The more students in the group, the better and more sophisticated the results should be! In addition to optimizing the structure, you may discuss its reliability. Calculate structural responses in terms of stresses, strains, and displacements. Vary your assumptions about the material to be used. Vary also the assumptions on the load cases that you assumed. You may use any of the following packages, • Matlab (Top3D)... • Ansys Workbench (fair to use. Includes SpaceClaim for re-modelling), • Ansys APDL • Rhino3D (a tool used by architects), • Grasshopper • Large scale topology optimization code using PETSc: (https://www.topopt.mek.dtu.dk/Apps-and-software/Large-scale-topology-optimization-code-using-PETSc), needs building own executable. Makefile is provided. For sophisticated programmers, recommended. Optimize any daily life product, e.g. a chair, a shelf, a bicycle, a ladder, a wrench, ... Plot your results accordingly! Recommended: After the topology optimization, apply a smoothing of the geometry (you can use RHINO or Ansys spaceclaim). Export the.stl file of your object. The nicest solutions will be ma nufactured additively (3D-Plot) after the project. 5 Experience Optimization Vividly with Self-built Robots Project (30 points): Within the framework of this project, you will analyze a suite of optimization algorithms designed to guide an autonomous rover to the minimum value of a given objective function. The core task is to conduct a data-driven, comparative performance analysis to determine the suitability of each algorithm for navigating different types of terrain (test functions). Your analysis must be based on quantitative metrics and should address the following aspects for each algorithm when applied to the specified benchmark functions: • Accuracy: The algorithm's ability to locate the known global minimum. • Efficiency (Function Calls): The total number of objective function evaluations required to reach a solution. This is the primary metric for efficiency, as each call represents an expensive rover operation. • Computational Cost: The total execution time (wall-clock time) of the optimization run. 4 • Robustness: An assessment of the algorithm's ability to avoid getting trapped in local minima on complex, non-convex problems. Algorithms for Analysis You will test the following four optimization algorithms, for which starter Python scripts are provided: • Steepest Descent (Advanced, Gradient-Based) • Simplex Nelder-Mead (Derivative-Free) • Simulated Annealing (Probabilistic, Global Search) • Adaptive Metamodeling (Surrogate-Based) Benchmark Test Environments The performance of the algorithms must be evaluated on the following benchmark functions. For a fair comparison, all algorithms must be run from the same starting point for a given test function, which represents the rover's initial deployment position. Required Test Cases: • Booth Function: A simple, convex-like test case. Use the starting point: [8.0, 9.0]. • Teughels Function: A complex, non-convex test case with multiple local minima. Use the starting point: [2.0, 2.0]. • In addition to the required functions, you must select, implement and test on two other well known 2D optimization benchmark functions of your choice. (You must briefly justify your choices in your report) Deliverables You are required to submit the following items for evaluation: 1. An archive file containing all Python source code files, including a main script (main analysis.py) that executes all required optimization runs and prints a clear, summa ry table of the performance metrics for each algorithm on all four test functions. 2. The findings are in the PDF report containing your comparative analysis, and conclude with a justified recommendation on which algorithm you would deploy on the rover for a mission in an unknown environment. Recommendation: After implementing the tests, visualize the optimization paths on contour plots for your analysis. Including one or two illustrative plots in your final report is highly encouraged to support your conclusions. Good luck!!